



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

La Universidad Católica de Loja

ESCUELA DE CIENCIAS DE LA COMPUTACIÓN

MODALIDAD ABIERTA

**Gestor de recursos bibliográficos para los centros de la
UTPL, utilizando herramientas sociales y mapas de
conocimiento: Mapas del Conocimiento
GERBI - MAP**

Trabajo de fin de carrera previa la
obtención del título de ingeniero en
informática

Autor: Vivanco Granda Marco Vinicio

Directora: Rodríguez Morales Germania del Roció, Ing.

Directora Puzzle: Cueva Carrión Samanta Patricia, Ing.

Centro Universitario Cariamanga

2012

CERTIFICACIÓN

Ingeniera.

Samanta Cueva C.

DIRECTORA PROYECTO PUZZLE

CERTIFICA:

Que el presente trabajo, denominado “Gestor de recursos bibliográficos para los centros de la UTPL, utilizando herramientas sociales y mapas de conocimiento: Mapas del Conocimiento” realizado por el profesional en formación: Marco Vinicio Vivanco Granda; cumple los requerimientos establecidos en las normas generales para la Graduación en la Universidad Técnica Particular de Loja, tanto en el aspecto de forma como de contenido, por lo cual me permito autorizar su presentación para los fines pertinentes.

Loja, mayo del 2012

f.-----

Ing. Samanta Cueva C.

DIRECTORA PROYECTO PUZZLE

CERTIFICACIÓN

Ingeniera.

Germania Rodríguez.

DIRECTORA DEL TRABAJO DE FIN CARRERA

CERTIFICA:

Que el presente trabajo, denominado “Gestor de recursos bibliográficos para los centros de la UTPL, utilizando herramientas sociales y mapas de conocimiento: Mapas del Conocimiento” realizado por el profesional en formación: Marco Vinicio Vivanco Granda; cumple los requerimientos establecidos en las normas generales para la Graduación en la Universidad Técnica Particular de Loja, tanto en el aspecto de forma como de contenido, por lo cual me permito autorizar su presentación para los fines pertinentes.

Loja, mayo del 2012

f.-----

Ing. Germania Rodríguez
DIRECTORA DE TESIS

CESIÓN DE DERECHOS

“Yo Marco Vinicio Vivanco Granda 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”.

f.-----

Vivanco Granda, Marco Vinicio
110455332-4

AUTORÍA

El presente proyecto de tesis con cada una de las observaciones, análisis, evaluaciones, conclusiones y recomendaciones emitidas, son de absoluta responsabilidad del autor.

De igual manera, es necesario indicar que la información de otros autores incluida en el presente trabajo se encuentra debidamente especificada en las fuentes de referencia y apartados bibliográficos.

f.-----
Vivanco Granda, Marco Vinicio
110455332-4

DEDICATORIA

El presente trabajo está dedicado a una persona muy especial a quien llevo en mi corazón, es el ser más importante de mi vida, Dios.

A mis padres Manuel Vivanco y María Granda por el apoyo incondicional ya que sin sus esfuerzos, cariños, sacrificios y dedicación a sus seis hijos nos han podido sacar a delante, con buen trabajo, honradez y esfuerzo, ellos son la piedra fundamental en mi vida, mi alegría se la debo a ellos ya que por su ejemplo y desvelo he sido la persona que ahora soy. A mis hermanas Paola, María Cristina, Betty y Mayra, y hermano Richard por la comprensión, compañía y ayuda brindada.

Marco Vinicio Vivanco Granda

AGRADECIMIENTO

En primer lugar agradezco a Dios por concederme la vida y por darme la oportunidad de estudiar una carrera y poder desarrollarme como profesional.

A mis padres por ser partícipes directos tanto moral como económicamente en mi formación profesional, brindando método el apoyo, ánimo y fuerzas durante todo el momento del desarrollo del presente proyecto, ya que sin ellos nada de esto hubiese sido posible.

A mis directoras del proyecto puzzle, Ing. Samanta Cueva C., directora de tesis, Ing. Germania Rodríguez, por la dirección brindada durante todo el desarrollo del proyecto,

A todos mis compañeros, amigos, sin su ayuda y consejos no hubiese sido posible la elaboración de la tesis.

Finalmente a la Universidad Técnica Particular de Loja por forjarme como profesional en un ambiente dotado de tecnología, por la formación recibida de parte de los docentes investigadores quienes han impartido sus conocimientos y han ayudado en mi crecimiento cognoscitivo capaz de aceptar cualquier reto profesional.

Marco Vinicio Vivanco Granda

ÍNDICE

ÍNDICE DE CONTENIDO

CERTIFICACIÓN	ii
CESIÓN DE DERECHOS	iv
AUTORÍA	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
RESUMEN EJECUTIVO	xx
INTRODUCCIÓN	xxi
OBJETIVOS	xxiii
CAPÍTULO I: REPRESENTACIÓN DEL CONOCIMIENTO	1
INTRODUCCIÓN	2
1.1. TÉCNICAS DE REPRESENTACIÓN	3
1.2. LENGUAJES DE REPRESENTACIÓN	6
1.3. HERRAMIENTAS PARA REPRESENTAR EL CONOCIMIENTO	10
1.4. COMPARATIVA RESUMEN DE LAS HERRAMIENTAS	19
CAPÍTULO II: MAPAS DE CONOCIMIENTO	23
2.1. DEFINICIONES	24
2.2. MAPAS DE CONOCIMIENTO COMO TÉCNICAS DE REPRESENTACIÓN DEL CONOCIMIENTO	24
2.2.1. MAPAS CONCEPTUALES	24
2.2.2. NUBES DE ETIQUETAS	26
2.3. CLASIFICACIÓN	29

2.4. UTILIDAD	29
2.5. HERRAMIENTAS	30
2.6. COMPARATIVA DE LAS HERRAMIENTAS	45
CAPÍTULO III: REPRESENTACIONES DEL CONOCIMIENTO DESDE ESTRUCTURAS SEMÁNTICAS	49
3.1. SEMÁNTICA	50
3.1.1. WEB SEMÁNTICA	50
3.2. RDF	51
3.2.1. COMPONENTES DE RDF	52
3.2.2. EL MODELO DE DATOS RDF	52
3.2.3. FORMAS DE REPRESENTACIÓN DE RDF	54
3.2.4. LENGUAJES DE CONSULTA PARA RDF	57
3.2.5. HERRAMIENTAS	59
3.3. COMPARATIVA DE LAS HERRAMIENTAS	67
CAPÍTULO IV: DISEÑO DE LA SOLUCIÓN	69
4.1. REQUERIMIENTOS	70
4.1.1. CASOS DE USOS	71
4.2. ARQUITECTURA DEL PROYECTO	78
4.2.1. COMPONENTES	80
4.3. TECNOLOGÍAS	81
4.3.1. APLICACIÓN WEB	81
4.3.2. ALMACÉN DE RDF	83
4.3.3. BASES DE DATOS	84

4.3.3.1. DISEÑO DE DATOS:	84
4.4. LENGUAJES DE CONSULTAS DE RDF	85
4.4.1. COMPARATIVA	89
4.4.2. PRUEBAS	91
4.4.2.1. DATOS Y ESCENARIOS DE CONSULTAS	92
4.4.2.2. COMPARATIVA CONSULTAS SERQL Y SPARQL	98
4.4.3. SELECCIÓN	105
4.5. DISEÑO DE INTERFACES	106
CAPÍTULO V: DESARROLLO DE APLICACIÓN	111
5.1. RECOLECCIÓN DE INFORMACIÓN	112
5.1.1. CONVERSIÓN DE COORDENADAS	113
5.2. EDICIÓN DE INFORMACIÓN	115
5.3. PROCESAMIENTO DE INFORMACIÓN	118
5.4. ALMACENAMIENTO	120
5.4.1. VALIDACIÓN DE GRAFO - SPARQL	121
5.4.2. ALMACENAMIENTO DE ESTADÍSTICAS	124
5.5. VISUALIZACIÓN DE INFORMACIÓN	125
5.5.1. RECUPERACIÓN DE INFORMACIÓN	126
5.5.2. PROCESAMIENTO DE INFORMACIÓN	129
5.5.3. VISUALIZACIÓN	131
5.5.3.1. IMPLEMENTACIÓN MAPA	131
5.5.3.2. INDIVIDUOS	134
5.5.3.3. ESTADÍSTICAS	140

DISCUSIÓN	143
CONCLUSIONES	148
RECOMENDACIONES	150
TRABAJO FUTURO	154
GLOSARIO	156
REFERENCIAS BIBLIOGRÁFICAS	159
ANEXOS	162
Anexo A: Instalación de Sesame	163
Anexo B: Instalación de Virtuoso OpenLink	167
Anexo C: Instalación de Apache - PHP- MySQL	171
Anexo D: Migración de Datos	173
Anexo E: Listado de Centros Seleccionados	182
Anexo G: Interfaz de Aplicativo	185

ÍNDICE DE FIGURAS

Figura 1.1. Evolución del Conocimiento	2
Figura 1.2. Ejemplo de Tripletas O-A-V	3
Figura 1.3. Ejemplo de Redes Semánticas	5
Figura 1.4. Salida de Jess	12
Figura 1.5. Salida de SWI-Prolog's	15
Figura 1.6. Salida de PowerLoom	18
Figura.2.1. Salida de CmapTools	32
Figura 2.2. Salida de Inspiration	34
Figura 2.3. Salida de SmartDraw	35
Figura 2.4. Salida de SmartDraw	37
Figura 2.5. Salida de TagCrowd	39
Figura 2.6. Salida de TagCanvas	40
Figura 2.7. Salida de InfoRapid KnowledgeMap	42
Figura 2.8. Salida de Knowledge Master	44
Figura 3.1. El modelo RDF	54
Figura 3.2. Grafo RDF Representación	55
Figura 3.3. Salida de Sesame	60
Figura 3.4. Salida de SemanticWorks	62
Figura 3.5. Salida de RDF Gravity	64
Figura 3.6. Salida de Protege	66
Figura 4.1: Vista de Caso de uso: Obtención de Información.	71
Figura 4.2. Vista de Caso de Uso: Generación de Individuos	72
Figura 4.3. Vista de Caso de Uso: Carga de Datos	74
Figura 4.4. Vista de Caso de Uso: Visualización Centros universitarios - Mapa	75
Figura 4.5. Vista de Caso de Uso: Registro - Visualización de Estadísticas	76

Figura 4.6. Arquitectura cliente-servidor	79
Figura 4.7. Arquitectura de la Solución	79
Figura 4.8. Diagrama Relacional de Base de Datos	85
Figura 4.9. Estructura básica de una sentencia RQL.	86
Figura 4.10. Estructura básica de una sentencia RDQL.	87
Figura 4.11. Estructura básica de una sentencia SeRQL	88
Figura 4.12. Sintaxis permitidas en Jena	91
Figura 4.13. Clases de archivo RDF.	93
Figura 4.14. Clases y Propiedades del archivo RDF.	96
Figura 4.15. Contenido del archivo RDF.	96
Figura 4.16. Interfaz Principal.	107
Figura 4.17. Interfaz seleccionando Centro Universitario.	108
Figura 4.18. Categorías del Centro Universitario	109
Figura 4.19. PopUp - Centro Universitario.	109
Figura 4.20. PopUp (mas+) - Centro Universitario.	110
Figura 5.1. Proceso previo de la información	112
Figura 5.2. Conversor de Coordenadas	114
Figura 5.3. Diagrama RDF - GERBI_MAP.	116
Figura 5.4. Archivos CSV	117
Figura 5.5. Estructura de Individuo (general)	118
Figura 5.6. Métodos de extracción.	119
Figura 5.7. Métodos de extracción.	119
Figura 5.8. Creación de Store - Virtuoso	121
Figura 5.9. Grafo en la Lista de Virtuoso	121
Figura 5.10. Especificación de Grafo - EndPoint	122
Figura 5.11. Registro de Estadísticas	124

Figura 5.12. Esquema de Funcionamiento - GERBI-MAP	126
Figura 5.13. Extracto de respuesta JSON	127
Figura 5.14. Función para petición HTTP al Store	128
Figura 5.15. Petición Ajax - JQuery	129
Figura 5.16. Recuperación de Variable POST	130
Figura 5.17. Extracción Atributos JSON	130
Figura 5.18. Inicialización de Mapa	132
Figura 5.19. Instancia de Mapa inicial.	133
Figura 5.20. Coordenadas para limitar a territorio nacional.	134
Figura 5.21. Estilo centros universitarios - OpenLayers	134
Figura 5.22. Creación de Centros	135
Figura 5.23. Centros universitarios sobre el GoogleMaps	136
Figura 5.24. Activar PopUp - Centro Asociado	136
Figura 5.25. Vista del PopUp.	137
Figura 5.26. Consulta SPARQL con las opciones offset y limit.	138
Figura 5.27. PopUp de Listado e Información del individuo.	139
Figura 5.28. Manejo de HTML con JQuery	139
Figura 5.29. Estadísticas de búsqueda	140
Figura 5.30. Estadísticas de búsqueda	141
Figura 1. Subir un archivo WAR. Tomcat	164
Figura 2. Aplicaciones Desplegadas	164
Figura 3. Interfaz Principal SeSame	165
Figura 4. Crear nuevo repositorio	165
Figura 5. Crear nuevo repositorio (Add RDF)	166
Figura 6. Espacio de Consulta - Sesame	166
Figura 7. Interfaz Principal Virtuoso	168

Figura 8. Interfaz de Administración - CONDUCTOR	169
Figura 9. Carga de Archivo OWL	170
Figura 10. Grafo Creado	170
Figura 11. Simbología	184
Figura 12. Interfaz Aplicativo: Vista General	185
Figura 13. Interfaz Aplicativo: Simbología	186
Figura 14. Interfaz Aplicativo: Estadísticas	186
Figura 15. Interfaz Aplicativo: PopUp	187
Figura 16. Interfaz Aplicativo: Mapa de Etiquetas	187
Figura 17. Interfaz Aplicativo: Listado de Items	188
Figura 18. Interfaz Aplicativo: Información del Individuo	188
Figura 19. Acerca de	189

ÍNDICE DE TABLAS

Tabla 1.1. Características Funcionales de Jess.	11
Tabla 1.2. Características Técnicas de Jess.	11
Tabla 1.3. Características Funcionales de Knowledge Machine 2.0	13
Tabla 1.4. Características Técnicas de Knowledge Machine 2.0	13
Tabla 1.5. Características Funcionales de SWI-Prolog's	14
Tabla 1.6. Características Técnicas de SWI-Prolog's	15
Tabla 1.7. Características Funcionales de KAON2	16
Tabla 1.8. Características Técnicas de KAON2	16
Tabla 1.9. Características Funcionales de PowerLoom	17
Tabla 1.10. Características Técnicas de PowerLoom	18
Tabla 1.11. Comparativa Resumen de las Herramientas para Representar el Conocimiento	19
Tabla 2.1. Características de los Mapas Conceptuales	25
Tabla 2.2. Características de Nubes de Etiquetas	27
Tabla 2.3. Características Funcionales de CmapTools	31
Tabla 2.4. Características Técnicas de CmapTools	31
Tabla 2.5. Características Funcionales de Inspiration	33
Tabla 2.6. Características Técnicas de Inspiration	33
Tabla 2.7. Características Funcionales de SmartDraw	35
Tabla 2.8. Características Técnicas de SmartDraw	35
Tabla 2.9. Características Funcionales de SemanticScuttle	36
Tabla 2.10. Características Técnicas de SemanticScuttle	37
Tabla 2.11. Características Funcionales de TagCrowd	38
Tabla 2.12. Características Técnicas de TagCrowd	39
Tabla 2.13. Características Funcionales de TagCanvas	40

Tabla 2.14. Características Técnicas de TagCanvas	40
Tabla 2.15. Características Funcionales de InfoRapid KnowledgeMap	41
Tabla 2.16. Características Técnicas de InfoRapid KnowledgeMap	42
Tabla 2.17. Características Funcionales de Knowledge Master	43
Tabla 2.18. Características Técnicas de Knowledge Master	43
Tabla 2.19. Comparativa Resumen de las Herramientas para Representar el Conocimiento.	45
Tabla 3.1. Características de RDF	52
Tabla 3.2. Características Funcionales de Sesame.	59
Tabla 3.3. Características Técnicas de Sesame	60
Tabla 3.4. Características Funcionales de SemanticWorks.	61
Tabla 3.5. Características Técnicas de SemanticWorks.	62
Tabla 3.6. Características Funcionales de RDF Gravity.	63
Tabla 3.7. Características Técnicas de RDF Gravity	64
Tabla 3.8. Características Funcionales de Protege.	65
Tabla 3.9. Características Técnicas de Protege.	65
Tabla 3.10. Comparativa Resumen de las Herramientas para RDF.	67
Tabla 4.1. Especificación Caso de Uso: Obtención de la información.	71
Tabla 4.2. Especificación Caso de Uso: Generación de Individuos.	73
Tabla 4.3. Especificación Caso de Uso: Carga de Datos	74
Tabla 4.4. Especificación Caso de Uso: Visualización Centros universitarios – Mapa	75
Tabla 4.5. Especificación Caso de Uso: Registro - Visualización de Estadísticas	77
Tabla 4.6. Sintaxis básica de una consulta SPARQL.	89
Tabla 4.7. Comparativa Lenguajes de Consulta para RDF.	90
Tabla 4.8. Atributos de Clase LIBRO.	93

Tabla 4.9. Atributos de Clase TESIS.	94
Tabla 4.10. Atributos de Clase REVISTA.	94
Tabla 4.11. Atributos de Clase GUÍA.	95
Tabla 4.12. Comparativa consultas SeRQL y SPARQL	98
Tabla 4.13. Resultados de SeRQL & SPARQL	105
Tabla 5.1. Individuos por Categoría.	115
Tabla 5.2. Propiedades RDF.	116
Tabla 5.3. Propiedades tipo Objeto - RDF.	117

RESUMEN EJECUTIVO

El proyecto GERBI-MAP realiza la construcción de un aplicativo para la representación del conocimiento sobre un mapa geolocalizado de la información bibliográfica (Libros, Tesis, Multimedia y Guías.) de los centros universitarios de la Universidad Técnica Particular de Loja utilizando tecnologías semánticas, migrando de estructuras convencionales a representaciones en tripletas (RDF) y almacenados sobre un TripleStore que es accesible a través de una interfaz web.

De esta forma el aplicativo GERBI-MAP está implementado en el servidor se puede decir que el aporte que el presente trabajo deja, se basa en la demostración de la posibilidad de migrar y visualizar datos en tripletas RDF de manera accesible y transparente para el usuario; además, la forma de comunicación entre un almacén de tripletas (Virtuoso) y una aplicación web.

Considerando de esta manera al aplicativo GERBI-MAP como una demostración más de la usabilidad de datos enlazados en la red mediante el uso de los principios de Linked Data.

INTRODUCCIÓN

Con el transcurso del tiempo, la información se ha convertido en parte fundamental de la sociedad, de igual manera la forma en que se representa, almacena y visualiza ha ido evolucionando de forma constante. Dejar pasando de los medios físicos a estructuras digitales.

A pesar del afán por adquirir y compartir información, la necesidad de generar conocimiento a partir de ella sigue siendo el fin único. De tal forma que se ha conseguido crear estructuras que almacenen información y permitan la deducción de conocimiento de una forma relativamente sencilla, conocido también como representación semántica.

Pero el problema no solo es la representación de información, sino el hacerla pública para que cualquier persona o servicio pueda acceder a la misma, en la actualidad se intenta conseguir este objetivo mediante lo que se conoce como la Publicación de Datos, siguiendo los principios propuestos por Open LinkedData¹, buscando así llegar a una web de datos capaz de generar conocimiento y no solo presentar información.

De esta forma, el presente proyecto de tesis, muestra una posibilidad de almacenar y representar información en una estructura semántica, mediante el uso de mapas de conocimiento. Para lo cual se profundiza en la temática de representación de conocimiento, estructuras semánticas y se analiza mediante comparativas, los lenguajes de consulta disponibles actualmente para la extracción de información de stores semánticos, siendo esta la base para el montaje de un servicio de almacenaje semántico totalmente funcional.

¹ <http://linkeddata.org/>

Las formas de representación de conocimiento existentes son tratadas en el apartado correspondiente al capítulo I. En el capítulo II, se comenta sobre el uso de mapas de conocimiento y sus diferentes tipos, así como aplicaciones. La representación del conocimiento en estructuras semánticas, así como la comparativa de los lenguajes de consulta sobre los almacenes de tripletas es tratada en el capítulo III. Y finalmente en el capítulo IV y V se describe la estructura y construcción del aplicativo que demuestra el uso de estructuras semánticas.

El propósito del Sistema Gestión de Biblioteca es de contar con un Gestor de Recursos Bibliográficos para todos los centros universitarios de la UTPL, utilizando la representación de mapas del conocimiento para visualizar las búsquedas de recursos de los Repositorios UTPL.

OBJETIVOS

OBJETIVO GENERAL

Realizar la representación de información bibliográfica, mediante el uso de estructuras semánticas, para ser visualizadas sobre un mapa de conocimiento.

OBJETIVOS ESPECÍFICOS

- Analizar las distintas formas de representación del conocimiento.
- Construir un repositorio semántico con información bibliográfica relacionada a los centros universitarios de la Universidad Técnica Particular de Loja.
- Determinar mediante una comparativa, el lenguaje de consulta semántico, que mejor se adapte al entorno a desarrollar.
- Desarrollar un aplicativo que permita consumir la información del Store Semántico y presentarlo al usuario como un mapa de conocimiento.
- Recolectar un conjunto de información relacionada a la visita de recursos que se encuentran almacenados en el Store.

**CAPÍTULO I:
REPRESENTACIÓN DEL
CONOCIMIENTO**

INTRODUCCIÓN

El **conocimiento** es una progresión que comienza con los **datos** que son de utilidad limitada [1].

- Mediante la organización o el análisis de los datos, entendemos lo que significan los datos, y esto se convierte en **información**.
- La interpretación o la evaluación del **conocimiento** de la información de rendimiento.
- La comprensión de los principios consagrados en el conocimiento es **sabiduría**.

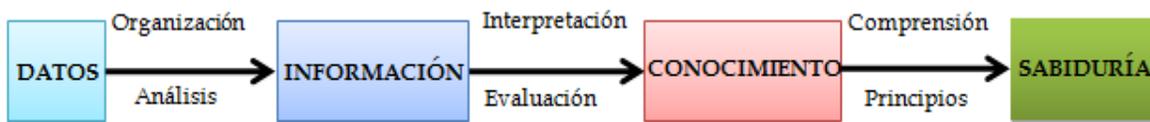


Figura 1.1. Evolución del Conocimiento

En palabras de *Brachman* y *Levesque* [2], la *Representación del Conocimiento* es “escribir, en algún lenguaje o medio de comunicación, descripciones o esquemas (imágenes) que se correspondan de alguna manera con aspectos importantes con el mundo”.

Otra definición de Representación del Conocimiento “es el proceso de transformación del conocimiento de un dominio a un lenguaje simbólico para ser comprendido de forma más eficiente por los seres humanos y por la computadora”.

1.1. TÉCNICAS DE REPRESENTACIÓN

Existen varias técnicas de representación de conocimiento pero las más usadas según [3] son las siguientes:

- **Tripletas Objeto-Atributo-Valor(O-A-V):** se utilizan para derivar una representación gráfica del conocimiento organizada en tres niveles: objetos compuesto de atributos que contienen valores. Una tripleta OAV, puede utilizarse para caracterizar todo el conocimiento de una red semántica; se representan en forma de grafos², representando los objetos mediante una elipse, un cuadrado para el valor, y para representar los atributos se utiliza una flecha entre ambos elementos.

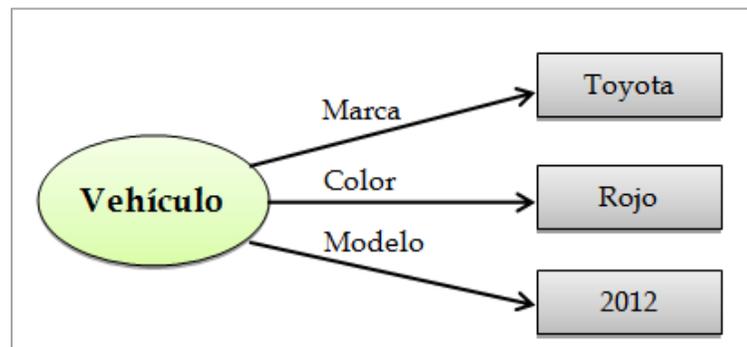


Figura 1.2. Ejemplo de Tripletas O-A-V

Los objetos suelen tener más de un atributo, por lo que en una representación gráfica puede haber varias flechas y cuadrados de cada elipse [3].

- **Rules o Reglas:** Las reglas son bastante fáciles de entender por los seres humanos y son también una potente máquina basada en esquema de representación del conocimiento, esta técnica utiliza unas premisas y conclusiones para representar el conocimiento.

² **Grafos:** se representa gráficamente como un conjunto de puntos (nodos), de los cuales algún par de ellos está conectado por unas líneas (aristas).

Se representan de la forma IF-THEN. Donde las premisas se colocan después del IF en forma normalmente de tripletas O-A-V y utilizando operadores booleanos (AND, OR, NOT), mientras que las conclusiones se encuentran después del THEN.

Una regla describe la lógica de pasar de un estado inicial (un conjunto de premisas válidas) a un estado objetivo (descrito por las conclusiones de la regla) [3].

Su forma general es [4]:

```
<rule> ::= If(<clausula_compuesta>
    Then(<clausula_compuesta>)
<clause> ::= <fact> | <pattern>
<clausula_compuesta> ::= <clause> | NOT<clausula_compuesta>
    | <clausula_compuesta>AND<clausula_compuesta>
    | <clausula_compuesta>OR<clausula_compuesta>
<fact> ::= <string>
<pattern> ::= "Una secuencia de las variables y cadenas "
<variable> ::= ? <string>
```

- **Redes Semánticas:** son redes de nodos y flechas. Los nodos representan conceptos o elementos particulares, las instancias y las relaciones se representan con las flechas. Estas relaciones pueden ser de cualquier tipo, además permiten representar estructuras jerárquicas del conocimiento.

Es fácil insertar nuevos nodos, las relaciones y los valores en una red semántica.

Esta técnica es intuitivamente clara y fácil de aprender y es muy utilizada por los especialistas en Inteligencia Artificial. Sin embargo, las redes semánticas se sabe que son débiles en la representación y el manejo de excepciones a los conocimientos que ellos representan. En redes grandes, todos los nodos deben ser cuidadosamente examinados para las excepciones [3].

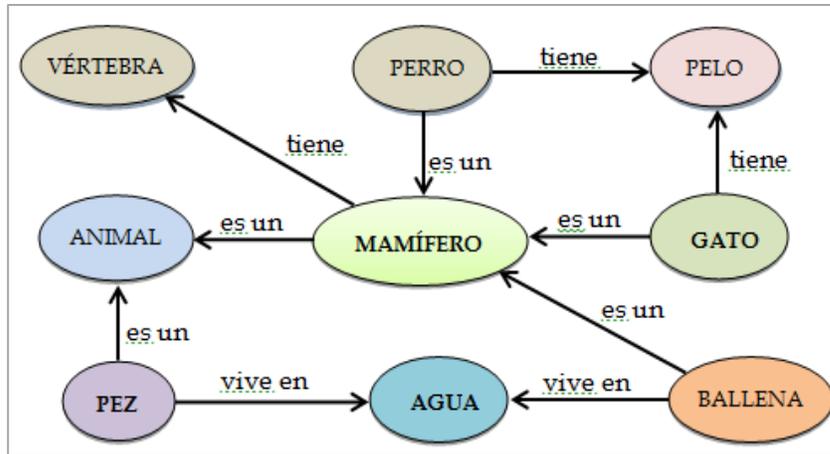


Figura 1.3. Ejemplo de Redes Semánticas

- **Frames o Marcos** [3]: es una técnica de representación muy similar a la utilizada en la programación orientada a objetos. Consta de **class frames**³, similares a las clases, que representan conjuntos de objetos con características similares. A partir de ellas se crean las **instance frames**⁴ que representan elementos concretos de esa clase. Por ejemplo, podríamos tener el marco de clase “Persona” y la instancia “Juan”. Cada frame dispone por otra parte de una serie de **slots**⁵ equivalentes a los atributos y propiedades en orientación a objetos.

Los marcos son apropiados para la representación de conocimientos de estereotipados. Las ranuras de los marcos se pueden insertar y borrar lo que proporciona una gran flexibilidad.

Su forma general es [4]:

```

<frame> ::= < frame_name > (<is_a>< slot >*)
<is_a> ::= (is_aframe_name >*)
<slot> ::= (<slot_name><slot_value>)
<slot_value > ::= < frame_name > | < string > | < number >

```

³ Término utilizado para crear una clase del marco

⁴ Término utilizado para crear una instancia de una clase del marco

⁵ **Def. slots.** Constituye el cuerpo y cada uno representa una propiedad estándar

Ejemplo:

(#Automóvil (es_un #Vehículo)

(Número_de_llantas 4)

(Color "Rojo")

(Propiedad # Toyota))

- **Representación Lógica:** Es el método más común de representar conocimientos. La lógica es el lenguaje como representación del conocimiento. Generalmente sus declaraciones se componen de dos partes: [5].
 - **Premisas:** la información y las observaciones
 - **Inferencias:** las conclusiones extraídas de las premisas

Hay varios tipos de representación de Lógica:

- Lógica de Proposicional (representación simbólica).
- Lógica de Predicados (lenguaje como representación).
- Lógica de Primer Orden (una mezcla de Proposicional y Predicado).
- Lógica Temporal.
- Lógica Difusa.

1.2. LENGUAJES DE REPRESENTACIÓN

Existen diversos tipos de lenguajes para la representación del conocimiento, a continuación se explican algunos de los más destacados lenguajes que se han utilizado y que se siguen utilizando hoy en día en diferentes ámbitos [3].

- **Lenguajes basados en Lógica Formal**

La lógica formal [3] es una ventaja para la representación del conocimiento, partiendo de una sintaxis y semántica bien definidas que detallan perfectamente la forma de construir sentencias y razonamientos sobre ellas.

Además, cada lógica formal tiene una sintaxis claramente definida que determina cómo:

- Las oraciones se construyen en el lenguaje,
- La semántica que determina el significado de las oraciones, y
- Un procedimiento de inferencia que determina las frases que se pueden derivar de otras sentencias.

Además, la mayoría de los lenguajes lógicos están diseñados para ser precisos y evitar las expresiones incompletas y ambiguas con frecuencia en los lenguajes naturales.

Lógica Proposicional: Lógica más sencilla. Representa únicamente hechos discretos del mundo real que pueden ser verdaderos o falsos. La lógica Proposicional asume que el conocimiento solo contiene hechos.

Una vez que la verdad de un enunciado se conoce, se convierte en una premisa que puede ser utilizada para obtener las nuevas proposiciones o las inferencias [5].

Símbolos de la lógica o conectores utilizados en la lógica de proposicional se muestran como:

- \wedge (And) Conjunciones
- \vee (Or) Disyunciones
- \neg (Not) Negaciones
- \Rightarrow (Implicación) Si- Entonces
- \Leftrightarrow (Equivalencia)
- \forall (Para todos)
- \exists (Existe)

Partiendo de los símbolos y utilizando los diferentes operadores se construyen proposiciones complejas, las cuales es posible obtener si son ciertas o falsas operando a partir de los valores de verdad de cada uno de los símbolos

iniciales, utilizando el cálculo proposicional [3].

Por ejemplo la sentencia si “La princesa está en el palacio (A) y el Rey está en el jardín (B), El Rey no puede ver a la princesa (C)”, podría representarse como: $A \wedge B \Rightarrow C$.

- **Lenguajes basados en Frames o Marcos**

Estos lenguajes son similares a los lenguajes de programación orientados a objetos, en el sentido de que modelan el conocimiento utilizando clases (frames), atributos, objetos y relaciones, y utilizan relaciones de generalización y especialización para representar la organización jerárquica de los conceptos.

Los lenguajes basados en marcos también permiten expresar la generalización/especialización, es decir, los conceptos de organización en jerarquías.

Uno lenguaje basado en marcos es KnowledgeMachine [6]. Mediante este lenguaje se define a continuación que cada “venta” consta de un objeto que se vende y un vendedor y comprador:

```
(every Venta has
  (vendedor((a Agent)))
  (objeto((a Thing)))
  (comprador((a Agent)))
)
```

En KM normalmente las clases necesitan dos marcos para describirlas, una para las propiedades de la clase, y otra para las propiedades de los miembros de la clase.

- **Lenguajes basados en Reglas**

Estos lenguajes son fáciles de entender debido a su sencillez conceptual y a su paralelismo con las estructuras de control más simples utilizadas en programación.

Todos los lenguajes basados en reglas tienen una sintaxis adecuada para representar la estructura de reglas If-Then. Por otro lado, existen considerables diferencias sintácticas entre los diferentes lenguajes en cuanto a la forma en que en realidad representan las reglas If-Then [3].

Sin embargo, los sistemas construidos con los lenguajes de reglas tradicionales tienen dos limitaciones importantes [7]:

- Que son muy restrictivas cuando se aplican a problemas de gran tamaño (es decir, cuando la base de reglas es grande), ya que no hay orden de las reglas.
- Las inferencias no puede limitarse a los que se ocupan sólo de los objetos de interés.

Esto tipo de lenguajes surgieron a flote a partir de la aparición de la web semántica, ya que sirven como herramientas para definir servicios web.

Para ello se ha creado una iniciativa Rule Markup Initiative que ha diseñado el lenguaje **RuleML**⁶, basado en XML y que por lo tanto está pensado desde su concepción para garantizar la interoperabilidad necesaria en el web.

Podemos representar en RuleML la sentencia "Si a alguien juega futbol, entonces a Luis le gusta jugar" de la siguiente forma:

```
<rulebase label="jugarFutbol">
  <imp>
    <_head><atom>
      <rel>jugar</rel>
      <ind>Luis</ind>
      <var>x</var>
    </atom></_head>
    <_body><atom>
```

⁶ <http://www.ruleml.org/>

```
        <rel>jugar</rel>
        <var>x</var>
        <ind>futbol</ind>
    </atom></_body>
</imp>
</rulebase>
```

Una regla no aporta mucho, pero un conjunto de reglas pueden formar una cadena capaz de alcanzar una conclusión significativa

Una de las tendencias más recientes pasa por mezclar los conceptos de frames y reglas, como en el caso de Jess, para disponer de lenguajes que permitan reunir la información de cada concepto y asociar a alguno de sus slots conjuntos de reglas.

1.3. HERRAMIENTAS PARA REPRESENTAR EL CONOCIMIENTO

En este apartado se menciona las distintas herramientas que permiten representar el conocimiento según las categorías (Reglas, Marcos, Semánticas y Representación Lógica).

- **Herramienta basada en Reglas**

Jess⁷ es un motor de reglas y entorno de programación escrito en su totalidad en el lenguaje Java de Sun por Ernest Friedman-Hill en los Laboratorios Nacionales Sandia, en Livermore, California.

Jess es una herramienta compatible con un lenguaje de reglas basado en XML, es pequeño, ligero, y uno de los motores más rápidos de reglas disponibles. Esta herramienta como pose un potente lenguaje de script da acceso a todos los APIs de Java, además incluye un entorno de desarrollo

⁷ <http://www.jessrules.com/>

basado en la plataforma Eclipse. Posee un lenguaje llamado JessML que fue diseñado para estar cerca de la estructura del lenguaje propio de Jess.

Las siguientes tablas muestran las características funcionales y técnicas de Jess.

Tabla 1.1. Características Funcionales de Jess.

XML nativo regla de apoyo al lenguaje	Jess tiene su propio lenguaje declarativo regla XML llamado "JessML ⁸ ", el cual está diseñado para ser fácil de transformar en otros lenguajes de reglas XML, así como en el lenguaje de la regla Jess.
Regla de la prevención de bucle	Problemas con las reglas bucles son una cosa del pasado. Jess da el poder para tratar con ellos de una manera clara y sensible.
Nueva API simplificada para consultas	La memoria de trabajo de búsqueda de Jess una interfaz familiar JDBC-like.
La manipulación de objetos de Java más rápido	Trabajando con objetos de Java de Jess es más fácil que nunca, y más rápido hasta un 50%.

Tabla 1.2. Características Técnicas de Jess.

Lenguajes	Java, XML
Sistemas Operativos	Windows, Linux, Mac OS X
Licencia	Código Cerrado (Not Open Source)
APIs	APIs de Java (javax.rules:JSR 94 API)

⁸ <http://www.jessrules.com/doc/70/XML.HTML>

Salida

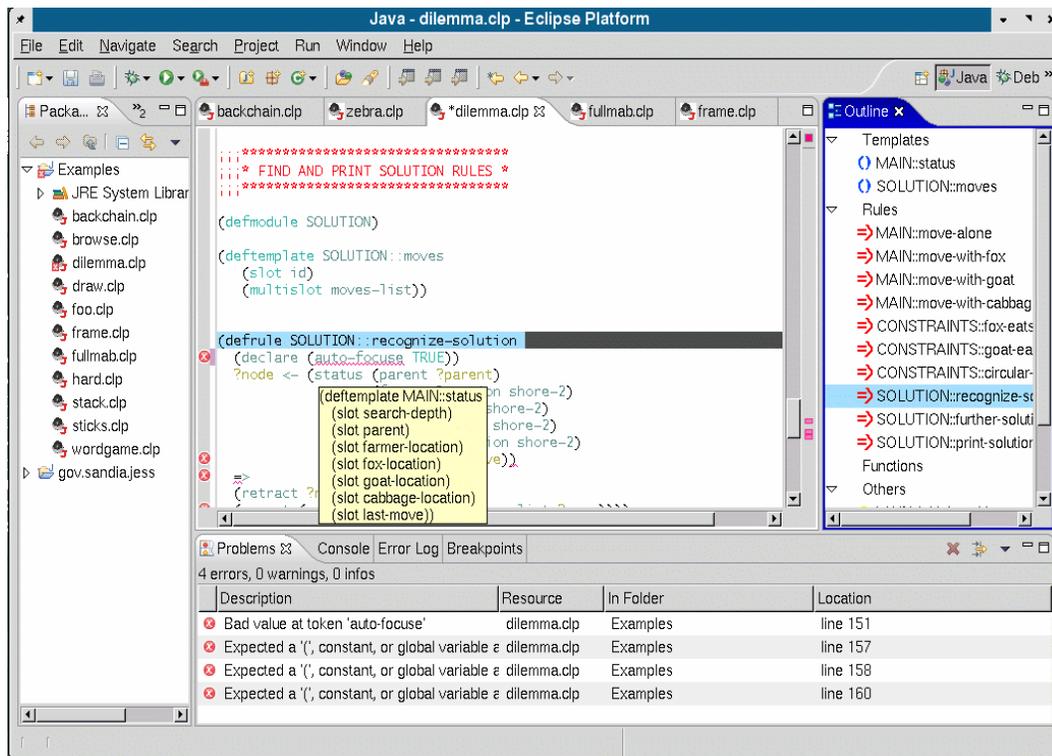


Figura 1.4. Salida de Jess⁹

- **Herramienta basada en Marcos**

La herramienta **KM (Knowledge Machine 2.0)** [6] es un potente y ligero lenguaje de representación del conocimiento basado en marcos con una semántica clara de la lógica de primer orden. Contiene una maquinaria compleja para el razonamiento, incluyendo la selección de la descripción, la unificación, clasificación y razonamiento acerca de las acciones mediante un mecanismo de situaciones.

Sus orígenes fueron el lenguaje THEO y el lenguaje KRL (ahora obsoleto). En la actualidad esta implementado en Lisp.

Las siguientes tablas muestran las características funcionales y técnicas de Knowledge Machine 2.0.

⁹ <http://www.jessrules.com/jess/editor.gif>

Tabla 1.3. Características Funcionales de Knowledge Machine 2.0

Capacidad de Inferencia	KM ha sido diseñado para ser capacidad de inferencia, es decir, no sólo para representar la razón, sino también como una amplia variedad de formas de representación.
Expresivo	KM es muy expresivo e incluye soporte para muchos de los fenómenos fundamentales de representación incluyendo cuantificación existencial y patrones universales, contextos, clases definidas, restricciones, representaciones intencionales, situaciones y metaclasses.
Semántica Formal	Las declaraciones de KM tienen una semántica clara y bien definida (en su mayoría) la lógica de primer orden.
Conjunto orientado a la evaluación	KM se trata de conjuntos de valores, en lugar de valores individuales, al encontrar "el valor de lo mucho" (es decir, los valores para el segundo argumento un predicado binario, dado su primero).
Maduro	KM ha evolucionado y se ha utilizado durante un largo período, y por lo tanto es un lenguaje relativamente maduro, incluido el apoyo para el seguimiento y la depuración, y la documentación del usuario.

Tabla 1.4. Características Técnicas de Knowledge Machine 2.0

Lenguajes	Lisp
Sistemas Operativos	Windows, Linux
Licencia	Free bajo la GNU Lesser General Public License (pero con fuera de garantía).
APIs	KM's API y KM's Lisp API

Salida

La salida de Knowledge Machine 2.0 es en un entorno DOS formato solo texto, no permite la salida gráficamente.

- **Herramienta basada en Redes Semánticas**

SWI-Prolog's¹⁰. Es un software libre, licenciado bajo la Licencia Pública GNU. Posee un kit de herramientas gráficas XPCE cuyo impulsado se dio por las necesidades de aplicaciones del mundo real. En la actualidad SWI-Prolog es utilizado para la investigación, la educación y para aplicaciones comerciales.

La web semántica es una de las áreas de aplicación más competente para SWI-Prolog. Prolog se encarga del modelo RDF natural de la web semántica, donde RDF proporciona un modelo estable para la representación del conocimiento con una semántica común.

Las siguientes tablas muestran las características funcionales y técnicas de SWI-Prolog's.

Tabla 1.5. Características Funcionales de SWI-Prolog's

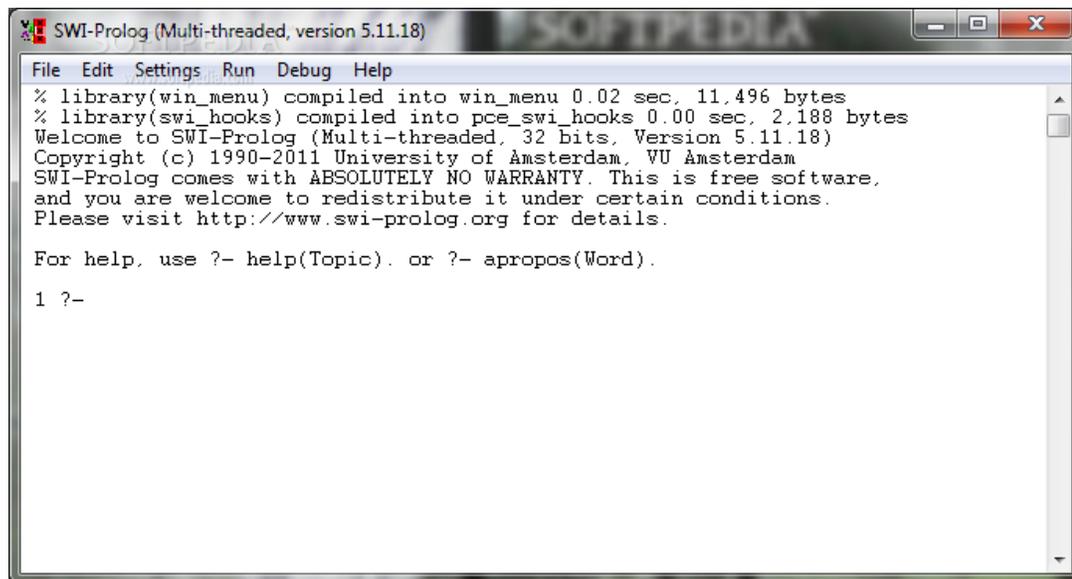
Robusta y libre de fugas de memoria	En el uso de varios servidores que se ejecutan 24x7 (incluyendo esta web-service).
Escalas para aplicaciones de gran tamaño.	No hay límite en el tamaño del programa, el átomo de duración, valores enteros. No hay degradación del rendimiento en los predicados con varios hechos.
Portátil a muchas plataformas	Incluyendo casi todos los Unix/Linux, plataformas de Windows, Mac OS X y muchos más.
Distribuciones binarias	Para Windows (32/64 bits) y Mac OS X (64 bits)).
Bibliotecas rápida y flexible	Para el análisis de SGML/XML (HTML), el análisis de RDF, almacenar y consultar el RDF <i>modelo de triple</i> . Un editor gráfico para RDF/RDFS y OWL.

¹⁰ <http://www.swi-prolog.org/>

Tabla 1.6. Características Técnicas de SWI-Prolog's

Lenguajes	Prolog
Sistemas Operativos	Windows, Unix/Linux, Macintosh
Licencia	Código Abierto (Open Source)
APIs	API ODBC (bajo nivel)

Salida



```
SWI-Prolog (Multi-threaded, version 5.11.18)
File Edit Settings Run Debug Help
% library(win_menu) compiled into win_menu 0.02 sec, 11,496 bytes
% library(swi_hooks) compiled into pce_swi_hooks 0.00 sec, 2,188 bytes
Welcome to SWI-Prolog (Multi-threaded, 32 bits, Version 5.11.18)
Copyright (c) 1990-2011 University of Amsterdam, VU Amsterdam
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to redistribute it under certain conditions.
Please visit http://www.swi-prolog.org for details.

For help, use ?- help(Topic). or ?- apropos(Word).

1 ?-
```

Figura 1.5. Salida de SWI-Prolog's¹¹

KAON2¹² sucesor de KAON1 es una infraestructura para la gestión de OWL-DL (*Web de Lenguaje de Ontologías*), SWRL (*Web semántica lenguaje de reglas*), y la F-lógica ontologías.

La diferencia está en el lenguaje de ontologías compatibles: KAON1 utiliza una extensión de propiedad de RDFS, mientras que KAON2 se basa en OWL-DL y F-Logic.

Las siguientes tablas muestran las características funcionales y técnicas de KAON2

¹¹ <http://www.softpedia.es/imagenes-SWI-Prolog-184315.HTML>

¹² <http://kaon2.semanticweb.org/>

Tabla 1.7. Características Funcionales de KAON2

API	Para la gestión programática de OWL-DL, SWRL, y la lógica F- ontologías.
Servidor independiente	Proporciona acceso a ontologías de forma distribuida con RMI.
Motor de Inferencia	Para responder a consultas conjuntivas (expresado con sintaxis SPARQL).
DIG Interfaz	Permitiendo el acceso de las herramientas como Protégé.
Módulo	Para extraer instancias de ontologías de bases de datos relacionales.

Tabla 1.8. Características Técnicas de KAON2

Lenguajes	Java
Base de Datos	SPARQL,SQL2, PostgreSQL, IBM DB2, Oracle 9i
Sistemas Operativos	Windows, Unix/Linux, Macintosh
Licencia	GNU Lesser General Public Licence (LGPL)
APIs	Si tiene API

- **Herramienta basada en Representación Lógica**

PowerLoom¹³ es el sucesor del sistema de representación del conocimiento Telar. Es totalmente expresivo, basado en la lógica del lenguaje de representación. Utiliza un motor de inferencia de la deducción natural que combina encadenamiento hacia adelante y atrás para obtener lo que se deduce lógicamente de los hechos y las normas afirmó en el conocimiento.

Para llevar a cabo PowerLoom se desarrolló un nuevo lenguaje de programación llamado **STELLA**, que es un establecimiento inflexible del tipo de lenguaje Lisp-Likes que puede ser traducido en Lisp, C++ y Java.

¹³ <http://www.isi.edu/isd/LOOM/PowerLoom/>

PowerLoom utiliza una API de java para presentar la interfaz gráfica de usuario. PowerLoom está escrito en STELLA y por lo tanto, disponibles en Common-Lisp, C++ y versiones de Java.

Las siguientes tablas muestran las características funcionales y técnicas de PowerLoom.

Tabla 1.9. Características Funcionales de PowerLoom

Representación del lenguaje	Utiliza el lenguaje de la lógica de predicados para representar el conocimiento. La sintaxis es KIF (formato de intercambio de conocimiento), que es una de las sintaxis de apoyo de la próxima lógica común estándar.
Razonamiento	PowerLoom utiliza un sistema de deducción natural para responder a consultas, pero también tiene un gran número de procedimientos de razonamiento especializados para manejar eficientemente el concepto y la relación de las jerarquías, los sistemas, los predicados marco, el control de búsqueda, etc.
Explicación	PowerLoom puede explicar su razonamiento mediante el registro de los árboles de inferencia y la representación en los humanos comprensibles las explicaciones.
Escalabilidad	A pesar de su énfasis en la representación expresiva que por lo general tiene que ser pagado con los procedimientos de razonamiento difíciles y costosas. PowerLoom es muy escalable y viene con una variedad de mecanismos para el control de búsqueda y manejar grandes ontologías y bases de conocimiento.

Tabla 1.10. Características Técnicas de PowerLoom

Lenguajes	STELLA, Lisp, C++, Java
Sistemas Operativos	Windows, Unix/Linux, Mac OS
Licencia	Código Abierto (Open Source)
APIs	KB's API

Salida

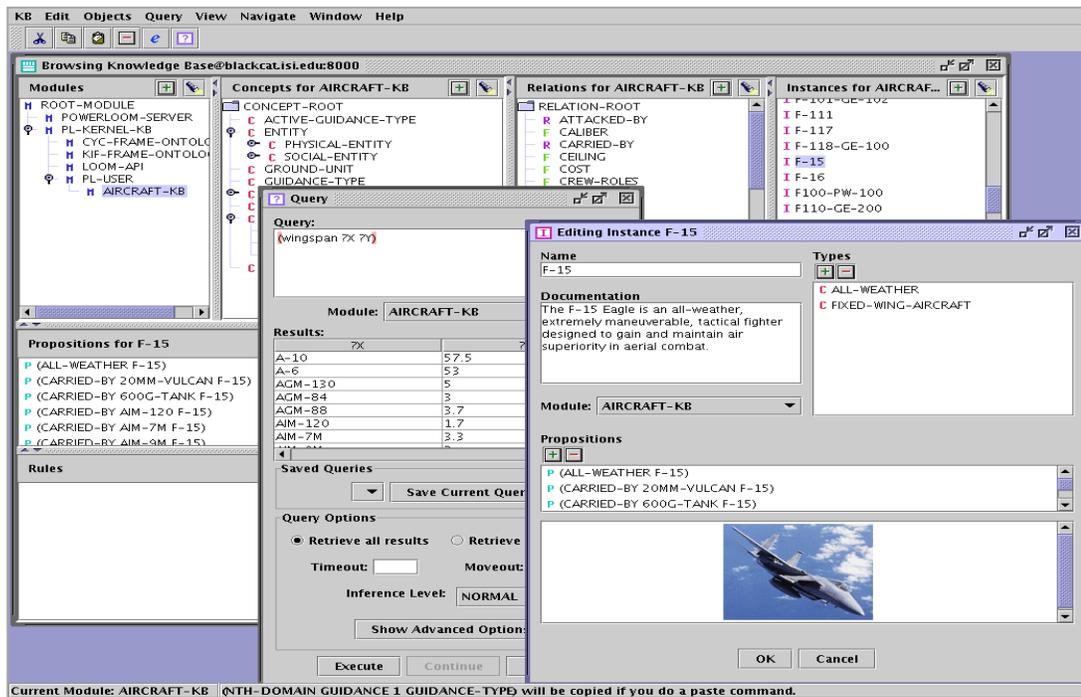


Figura 1.6. Salida de PowerLoom¹⁴

¹⁴ <http://www.isi.edu/isd/LOOM/PowerLoom/documentation/gui-screenshot.gif>

1.4. COMPARATIVA RESUMEN DE LAS HERRAMIENTAS

La siguiente tabla comparativa muestra un resumen de las diferentes herramientas que se mencionó anteriormente, al finalizar se escogerá la mejor herramienta que se adapte a la solución.

Tabla 1.11. Comparativa Resumen de las Herramientas para Representar el Conocimiento

Técnicas de Conocimiento	Ventajas	Desventajas	Lenguajes	Herramientas
Tripletas Objeto - Atributo- Valor	<ul style="list-style-type: none"> Sirve para ordenar el conocimiento en forma de grafos Permite traducir el conocimiento al código de la computadora mediante la inducción de reglas. 	<ul style="list-style-type: none"> No permite asignar graduaciones de certeza a las asignaciones. No se utiliza herencia, sólo relaciones casuales o enlaces, partes o subpartes. Existe poca información por lo que no es muy usada. 		
Reglas	<ul style="list-style-type: none"> Cada regla es independiente de las demás. Las inferencias y explicaciones son obtenidas fácilmente. Las modificaciones son relativamente fáciles. 	<ul style="list-style-type: none"> El conocimiento complejo requiere muchas reglas. Dependencia de reglas en conocimiento complejo. El mecanismo de inferencia suele ser pre-definidos para una base de reglas. 	<ul style="list-style-type: none"> RuleML Jess JessML 	<ul style="list-style-type: none"> Jess 7

	<ul style="list-style-type: none"> • Es fácil combinar la incertidumbre con las reglas. 	<ul style="list-style-type: none"> • No representan el conocimiento declarativo, naturalmente que lo utilizan. • Simples implementaciones son muy ineficientes. • Algunos tipos de conocimientos no son expresados fácilmente en dichas reglas. • Si las bases de conocimientos crecen lo suficientemente grandes, son difíciles de mantener y depurar. 		
Marcos	<ul style="list-style-type: none"> • Se pueden almacenar intervalos o valores dinámicos en las ranuras. Esto es útil para hacer simulación. • Conocimiento significativamente estructurado. • Se les pueden asignar valores y tipos default¹⁵. • Usan los conceptos conocidos de 	<ul style="list-style-type: none"> • Ninguna descripción del procedimiento. • La jerarquía “es_un” se utiliza tanto para la especialización e instancias. • Difícil de representar las influencias y las dependencias • Dificultad para representar objetos que se alejan de los estereotipos. • Problemas para representar 	<ul style="list-style-type: none"> • CLASSIC • KL-ONE • KM (Knowledge Machine) • KRL (Knowledge Representation Language) 	<ul style="list-style-type: none"> • Knowledge Machine 2.0

¹⁵ Valores por defecto: “se utilizan para representar el sentido común”.

	<p>la POO¹⁶ como: clases, herencias, encapsulamiento, etc.</p> <ul style="list-style-type: none"> • Es útil para la representación del conocimiento de estereotipados. • Permiten trabajar con información incompleta. 	<p>conocimiento no taxonómico¹⁷.</p> <ul style="list-style-type: none"> • Dificultad para describir conocimiento heurístico que mucho más fácil de representar mediante reglas. 		
Redes Semánticas	<ul style="list-style-type: none"> • Se introduce el concepto de distancia semántica¹⁸ • Representan relaciones jerárquicas de forma modular. • No depende del orden. • Se pueden relacionar varias redes semánticas entre sí. 	<ul style="list-style-type: none"> • Semántica poco clara (no “normalizada”). • Problemas para representar disyunciones, implicaciones, negaciones. • No hay representación de la semántica de los enlaces. • Las relaciones entre múltiples nodos no es posible. • No existe una clasificación de los nodos. • No hay una descripción del 	<ul style="list-style-type: none"> • KL-ONE • PROLOG • Telos 	<ul style="list-style-type: none"> • SWI-Prolog's • KAON2

¹⁶ POO: Programación Orientado a Objetos

¹⁷ **Conocimiento taxonómico:** *Benjamín Blooms*: Tiene un orden jerárquico, es decir, va desde el conocimiento más simple al más elaborado o complejo.

¹⁸ **Distancia semántica:** “cantidad de enlaces que separan un nodo de otro”.

		<p>procedimiento.</p> <ul style="list-style-type: none"> • La falta de normas para las relaciones entre las instancias. • La representación gráfica puede ser muy grande para los problemas complicados. 		
Representación Lógica	<ul style="list-style-type: none"> • Notación gráfica facilita comprensión. • Fácil especificar y manejar excepciones. • Modelo de ejecución más sencillo y eficiente. 	<ul style="list-style-type: none"> • Los aspectos de procedimiento solo pueden ser representados por mecanismos de inferencia • Una vez activado, un mecanismo de inferencia generará todo • Es difícil de representar generalizaciones 	<ul style="list-style-type: none"> • KIF • STELLA 	<ul style="list-style-type: none"> • PowerLoom

Al finalizar el capítulo I, luego de analizar las formas de representar el conocimiento, la técnica que mejor se adapta a la solución son las redes semánticas, esto debido a que:

- Una red semántica posee una estructura capaz de permitir el acople de diferente información a la estructura ya existente, de tal forma que esta pueda crecer.
- Permite la derivación y la relación de información mediante enlaces (propiedades) que dan sentido a los datos.

CAPÍTULO II:

MAPAS DE CONOCIMIENTO

MAPAS DE CONOCIMIENTO

Los mapas de conocimiento han atraído la atención de los investigadores por la necesidad de incrementar la transparencia y reducir la complejidad del conocimiento en su gestión [8].

2.1. DEFINICIONES

Según la definición propuesta por *Davenport y Prusak* [9], quienes definen un mapa de conocimiento “como guías que ayudan a trazar o encontrar el conocimiento de una organización, por ejemplo, por medio de apuntadores a las fuentes de ese conocimiento”.

Así mismo a los mapas de conocimiento “se los podría considerar como representaciones gráficas que permiten buscar qué conocimientos están disponibles y algunos atributos como están localizados, con que otros están relacionados y quiénes son los poseedores de los mismos”.

Los mapas de conocimiento son una herramienta eficaz para visualizar y optimizar el uso de la información, válida para la toma de decisiones.

2.2. MAPAS DE CONOCIMIENTO COMO TÉCNICAS DE REPRESENTACIÓN DEL CONOCIMIENTO

En base a las definiciones de las técnicas de representación del conocimiento analizadas en el Capítulo I la más relacionada con los mapas de conocimiento serían las Redes Semánticas.

Sin embargo en la bibliografía disponible se identifican otras formas de representación del conocimiento como las que se mencionan a continuación:

2.2.1. MAPAS CONCEPTUALES

Los mapas conceptuales según [10] son instrumentos de representación de conocimientos sencillos y prácticos, que permiten transmitir con claridad

mensajes conceptuales complejos y facilitar tanto el aprendizaje como la enseñanza.

Según la definición propuesta por Joseph D. Novak, define a un mapa conceptual como “un recurso esquemático para presentar un conjunto de significados conceptuales incluidos en una estructura de proposiciones”.

El objetivo es **representar relaciones entre conceptos en forma de proposiciones**. Los conceptos están incluidos en círculos, mientras que las relaciones entre ellos se explicitan mediante líneas que unen sus círculos respectivos. Las líneas, a su vez, tienen palabras asociadas que describen cuál es la naturaleza de la relación que liga los conceptos.

La siguiente tabla muestra las características de los mapas conceptuales como son:

Tabla 2.1. Características de los Mapas Conceptuales

Jerarquización	<p><i>En la parte superior</i> de la estructura del mapa, van los conceptos más importantes y amplios y que incluyen a los otros.</p> <p><i>En la parte inferior</i>, van los demás conceptos, de acuerdo a cuán importantes y específicos sean.</p> <p>No hay repetición de conceptos.</p>
Selección	<p>El mapa contiene lo más significativo de un tema; puede ser ampliado en subtemas y puede variar de acuerdo con el proceso de la investigación.</p>
Impacto Visual	<p>Sencillo y ágil para captar la observación visual.</p>

Tipos

Existe diversos tipos de mapas conceptuales pero los más utilizados son:

- **Mapa Jerárquico:** es un modelo de datos en el cual los datos son organizados en una estructura parecida a un árbol. Es decir a partir del concepto principal va descendiendo verticalmente según el orden de importancia.
- **Mapa en Araña:** este tipo de mapas se organizan situando el tema principal en el centro y hacia afuera se van dibujando los subtemas que lo rodean, similares a las patas de una araña.
- **Mapa de Organigrama:** este tipo de mapas organiza la información de manera lineal o bidimensional, pero con la adición de entradas y salidas.

Herramientas

Existen algunas herramientas disponibles para la creación de mapas conceptuales entre ellas:

- CmapTools
- Inspiration
- SmartDraw

Todas estas herramientas se mencionan con más detalle en el apartado de la sección 2.5.

2.2.2. NUBES DE ETIQUETAS

La palabra “Etiqueta” (tag, en inglés) se suelen utilizar para clasificar información. Como metadatos, las etiquetas se utilizan para clasificar la información.

Según la definición propuesta por Iván Lasso Clemente [11] define a una etiqueta como “una palabra o una expresión corta que se asigna a una información, como una dirección web, la entrada de un blog, un vídeo, una imagen, etc.”. Las etiquetas permiten una clasificación más flexible y sencilla, pues asociamos la información con palabras relevantes para nosotros.

Las etiquetas son utilizadas en la actualidad a diario en blogs, páginas web, redes sociales, etc.

Una nube de etiquetases una simple representación gráfica de los términos utilizados en las taxonomías de una página web, blog, etc., donde el tamaño de la fuente de cada palabra que forman parte de la nube varia, es decir mientras más veces se repite la palabra mayor sea el tamaño.

La siguiente tabla muestra las características funcionales de Nubes de Etiquetas como son:

Tabla 2.2. Características de Nubes de Etiquetas

<p>Ordenación</p>	<ul style="list-style-type: none"> • Alfabética: Los términos se ordenan alfabéticamente. • Por frecuencia: Los términos se muestran por frecuencia de uso.
<p>Agrupación</p>	<ul style="list-style-type: none"> • Alfabética: Los términos se separan por letra inicial. • Semántica: Los términos se agrupan por co-ocurrencia.
<p>Espacio donde se muestra la nube</p>	<ul style="list-style-type: none"> • El espaciado e interlineado entre tags son gestionados por el navegador. • Se aplican técnicas de CSS y HTML para aprovechar mejor el espacio.

Tipos

Existen varias maneras de implementar las nubes de etiquetas como:

- **Las etiquetas se ordenan alfabéticamente.** Los términos más importantes y frecuentes se destacan a través de un tamaño de fuente apropiado. Términos más importantes se resaltan con un color de fuente o un color de fondo.
- **Las etiquetas son ordenados según su importancia o frecuencia.** Ambos tamaño de fuente y colores se pueden utilizar para destacar la importancia de los términos.
- **Las etiquetas no son ordenados en absoluto.** En tamaño y peso de la fuente y los colores en uso.
- **Las etiquetas son ordenados de acuerdo a su similitud.** Términos similares aparecen como vecinos junto a la otra. Una variedad de formato visual se puede aplicar.
- **Las etiquetas se utilizan como un método de categorización de los elementos de contenido.** Las etiquetas son representados en una nube donde las grandes marcas representan la cantidad de elementos de contenido en esa categoría.

Herramientas

Existen algunas herramientas disponibles para la creación de nubes de etiquetas entre ellas:

- SemanticScuttle
- TagCrowd
- TagCanvas

Todas estas herramientas se mencionan con más detalle en el apartado de la sección 2.5.

2.3. CLASIFICACIÓN

Existen diferentes enfoques acerca de mapas de conocimiento. Los más simples según [12] se enfocan en “bases de datos, con campos que se refieren a los conocimientos disponibles en la organización, la actividad en la que se aplican y las personas de la organización que poseen dichos conocimientos”.

De acuerdo con la UNIVERSIDAD EAFIT [13], se identifican tres modelos de mapas de conocimiento, que son:

1. **Modelo de puntos:** señalan la fuente del conocimiento (persona), suministran información limitada y como ejemplo se tienen las “páginas amarillas corporativas”.
2. **Modelo de links:** contiene lo del modelo de puntos y adiciona relaciones del estilo cómo-cuándo, apoyando la creación de una visión del negocio.
3. **Modelo de soluciones:** contiene lo del modelo de puntos y de links. Relaciona el conocimiento con la solución de problemas, es tanto descriptivo como prescriptivo y requiere de más inversión y planificación.

2.4. UTILIDAD

La principal utilidad de los mapas de conocimiento radica en la facilidad que brindan como herramienta de visualización y localización de la información, permitiendo obtener información rápidamente y mostrarla al usuario de manera eficaz y fácil de accederla.

De acuerdo con DAVENPORT & PRUSAK [14], “el más claro beneficio de un mapa de conocimiento es mostrar a las personas en la organización a dónde ir cuando ellos requieren de experticia”. Así mismo, complementan que también puede servir como un inventario al representar gráficamente qué existe en la organización y dónde puede ser localizado.

La utilización de los mapas de conocimiento radican en:

- Facilitar la concentración de recursos.
- Evitar que las personas se dediquen a crear conocimientos que ya existen.
- Permitir localizar la mejor fuente para conseguir un conocimiento.

2.5. HERRAMIENTAS

En esta sección se mencionan las diferentes herramientas para la creación de mapas conceptuales, mapas de etiquetas y mapas de conocimiento como son:

Herramientas para Mapas Conceptuales

Las siguientes herramientas que se mencionan permiten la creación de mapas conceptuales, donde la más utilizada es la herramienta CmapTools:

- **IHMC CmapTools¹⁹**

CmapTools es el software para la edición de mapas conceptuales, desarrollado en el Institute For Human And Machine Cognition o IHMC. El IHMC es un instituto de investigación sin ánimo de lucro del sistema de Universidades de la Florida.

Con CmapTools, los usuarios pueden construir, navegar, y compartir modelos de conocimiento en muy pocos pasos. Esta herramienta es intuitiva y fácil de usar ya que CmapTools ofrece una serie de herramientas para que cada usuario elabore sus mapas conceptuales en su computador, y después publique en Internet a través de un servidor. Esta herramienta permite exportar los gráficos elaborados en forma de: imagen, página Web, texto o formato XML.

¹⁹ <http://cmap.ihmc.us/download/>

IHMC CmapTools COE (Concept Ontology Environment) es una herramienta semántica construida encima de la versión IHMC CmapTools Suite de aplicación básica para mapas conceptuales [15].

COE importa y visualiza ontologías (OWL, RDF, URI) en formato .cmap. Esta aplicación pública y exporta los mapas conceptuales (.cmap) a varios formatos de ontología OWL/RDF, RDF/XML, Turtle y N3.

Las siguientes tablas muestran las características funcionales y técnicas de la herramienta CmapTools como son:

Tabla 2.3. Características Funcionales de CmapTools

Mecanismos de Elaboración	Permite varios tipos de mecanismos de elaboración del mapa conceptual (inserción y eliminación de conceptos y enlaces) son fáciles y cómodos.
Interfaz	Interfaz comunicativa y fácil de usar en cuanto a las funcionalidades disponibles.
Conceptos	Permite acompañar a cada concepto de una imagen, permitiendo hacer mapas más fáciles de interpretarlos.
Enlaces	Al enlace es posible incorporarle flechas indicativas de la dirección del enlace.
Recursos	Posibilidad de vincular recursos a los conceptos.

Tabla 2.4. Características Técnicas de CmapTools

Sistemas Operativos	Windows, Mac OS X, Linux y Solaris
Licencia	Open Source, Freeware y Comercial
APIs	No existe

Salida

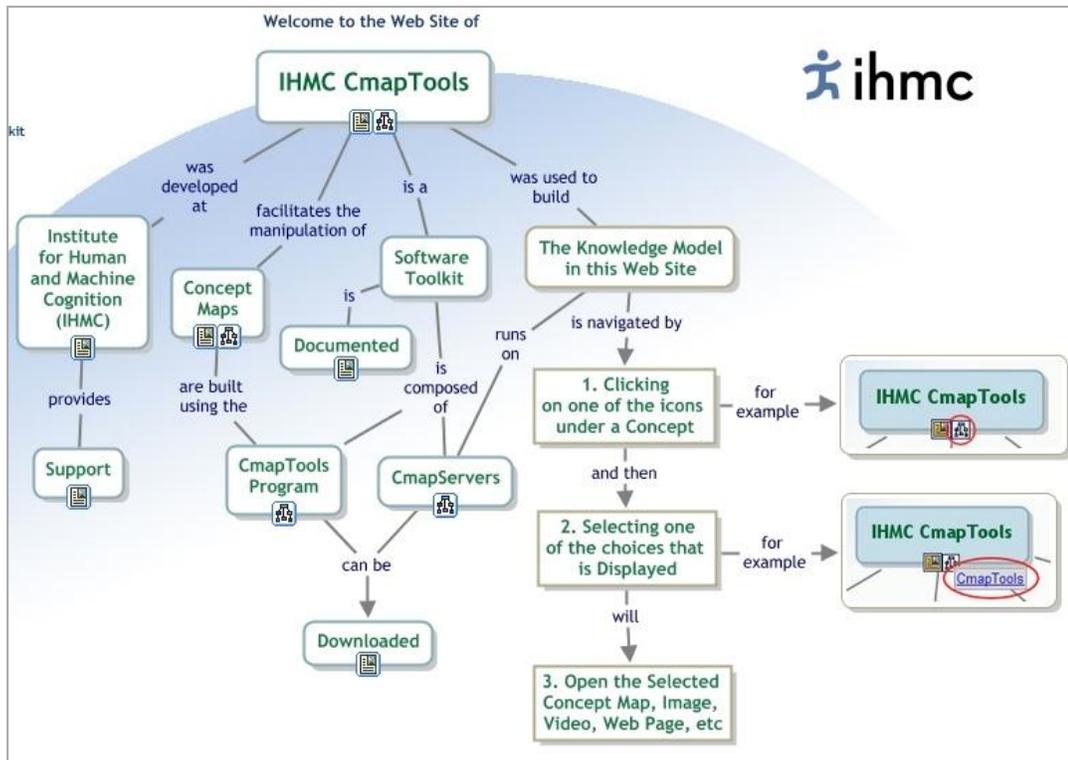


Figura.2.1. Salida de CmapTools²⁰

- **Inspiration²¹**

Inspiration 9 es la última herramienta de pensamiento y aprendizaje. Permite crear una lluvia de ideas, estructurarlos pensamientos y la comunicación visualmente de conceptos para fortalecer el entendimiento de los diagrama.

Esta herramienta cuenta con una interfaz suficientemente sencilla y fácil de usar para el usuario, permitiendo crear mapas conceptuales para expresar gráficamente un concepto o poder asimilar de un vistazo un conjunto de ideas. Permite exportar los mapas creados a formatos de gráficos.

Es una herramienta de aprendizaje visual, diseñada para la creación de diagramas en forma de telaraña, mapas de ideas y mapas conceptuales.

²⁰ <http://cmap.ihmc.us/>

²¹ <http://www.inspiration.com/Inspiration>

Las siguientes tablas muestran las características funcionales y técnicas de la herramienta Inspiration como son:

Tabla 2.5. Características Funcionales de Inspiration

Maneras de participar y aprender visualmente.	Permite crear webs, mapas de ideas, mapas mentales, mapas conceptuales, organizadores gráficos, y otros diagramas para pensar, organizar y escribir.
Planificar y organizar el trabajo por escrito	Organiza las ideas a través de proyectos y tareas con mayor eficacia utilizando una de vista Esquema. Con la vista Esquema, permite desarrollar rápidamente las ideas principales y secundarias y añadir detalles para ayudar a escribir su propósito.
Mejorar la comunicación	Inspiration proporciona varias herramientas para mejorar la comunicación y compartir lo que se ha aprendido, mediante la creación de presentaciones organizadas que transmitan ideas con claridad y demostrar su comprensión y conocimiento.

Tabla 2.6. Características Técnicas de Inspiration

Sistemas Operativos	Windows y Macintosh.
Licencia	Código Cerrado (Not Open Source)
APIs	No existe

Salida

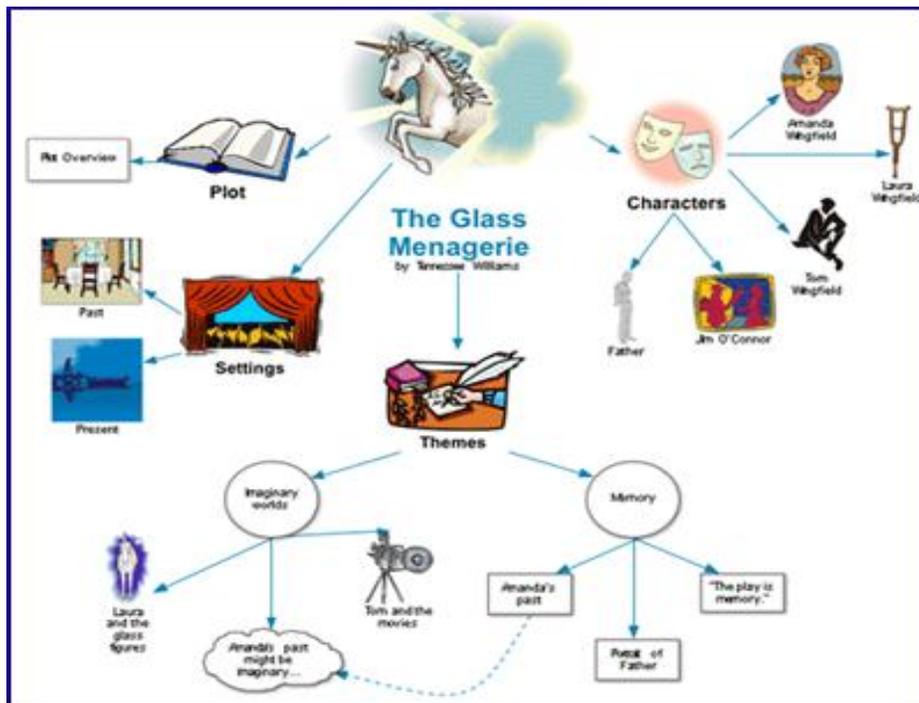


Figura 2.2. Salida de Inspiration²²

- **SmartDraw²³**

SmartDraw es una herramienta de modelado interactiva que se destaca por su gran variedad de modos de modelado, adaptándose a cualquier tipo de objetivo que se puede definir o diseñar. Esta herramienta facilita la elaboración de diagramas de flujo, diagramas causa-efecto, mapas de ideas, mapas conceptuales, redes, etc.

SmartDraw no es gratis, es de uso comercial, esta es la desventaja de esta herramienta, se puede probar sin límite durante siete días, a partir de entonces su uso requiere una licencia individual.

Las siguientes tablas muestran las características funcionales y técnicas de la herramienta SmartDraw como son:

²² <http://www.inspiration.com/Inspiration/examples>

²³ <http://www.smartdraw.com/>

Tabla 2.7. Características Funcionales de SmartDraw

Entorno de trabajo	Se configura de acuerdo con el tipo de diagrama que se esté elaborando.
drag-and-drop (Arrastrar y soltar)	SmartDraw tiene la característica de arrastrar y soltar lo permite al usuario copiar fácilmente los objetos y símbolos.
Multiherramienta	Proporciona la opción de crear más de 70 entornos visuales, incluyendo esquemas, diagramas de flujo, esquemas temporales, mapas, etc.
Integrado	Trabaja con Microsoft Office, permite transferir sus presentaciones a Word o PowerPoint, también permite importar datos de Excel fácilmente.

Tabla 2.8. Características Técnicas de SmartDraw

Sistemas Operativos	Windows y Macintosh.
Lenguaje	XML
Licencia	Código Cerrado (Not Open Source)
APIs	No utiliza API

Salida



Figura 2.3. Salida de SmartDraw²⁴

²⁴ <http://www.smartdraw.com/>

Herramientas para Mapas de Etiquetas

Las siguientes herramientas que se mencionan permiten la creación de mapas de etiquetas, donde la más utilizada es la herramienta TagCanvas:

- **SemanticScuttle²⁵**

SemanticScuttle es una herramienta de código abierto de aplicaciones web de marcadores sociales a experimentar con características similares a las etiquetas (tags). Esta herramienta se basa en Scuttle (un clon de código abierto de del.icio.us). SemanticScuttle se caracteriza por poseer estabilidad y facilidad de uso.

Relativamente sencillo, permite el sistema de marcado y un análisis sofisticado de etiquetas. El esquema consta de once tablas de SQL, libres de modificar. Posee una agradable nube de etiquetas que representa la clasificación de sus contenidos, este sistema puede ser administrado por uno o más usuarios.

Las siguientes tablas muestran las características funcionales y técnicas de la herramienta SemanticScuttle como son:

Tabla 2.9. Características Funcionales de SemanticScuttle

Autenticación de usuario personalizable	Una de las tareas básicas de la integración es la autenticación de usuarios contra una base de datos central - ya sea una base de datos central de usuarios, un LDAP o un servidor de Active Directory.
Certificados SSL de cliente	Mediante el uso de certificados SSL de cliente, se registra automáticamente en SemanticScuttle sin necesidad de utilizar una contraseña o enviar un formulario de inicio de sesión.
Temas	SemanticScuttle puede cambiar visualmente mediante el

²⁵ <http://sourceforge.net/projects/semanticscuttle/>

	suministro de costumbre "temas" (skins) que modifican el aspecto visual.
Archivos de Configuración Personalizables	SemanticScuttle utiliza al menos dos archivos de configuración: <ol style="list-style-type: none"> 1. Configuración por defecto de archivos <i>config.default.php</i>. 2. Configuración personalizada de archivos <i>config.php</i>.

Tabla 2.10. Características Técnicas de SemanticScuttle

Lenguaje	PHP
Base de Datos	ADODB, MySQL
Sistemas Operativos	Windows, Linux y Macintosh.
Licencia	Código Abierto.GNU General Public License
APIs	API de aplicación Del.icio.us

Salida

The screenshot shows the SemanticScuttle website interface. At the top, there is a navigation bar with a yellow star logo and the text "semanticscuttle". To the right of the logo are links for "bookmarks", "tags", "watchlist", "profile", and "add a bookmark". Below the logo is the tagline "More than tags...". On the far right of the navigation bar is a user profile link "mensonge (log out)".

Below the navigation bar is a section titled "popular tags". It features a "Sort by:" dropdown menu with "Alphabet" and "Popularity" options. The main content area displays a grid of tags, with "acier", "animation", "avion", "cao", "cnf", "format", and "logiciel" being the most prominent. Other visible tags include "3d", "777", "a340", "ac-limoges", "académie de caen", "académie_ah-marseille", "académie_amiens", "académie_besançon", "airbus", "asservissements", "automatique", "automobile", "bac", "bac pro", "bep", "bo", "boeing", "boeing777", "bois", "bts-cpi", "c_est_pas_sorcier", "cachan", "calcul", "centre-intérêt", "cerpet", "cni", "cni-cmao", "conique", "construction", "construction mécanique", "construction mécanique", "cours", "couteau", "cycle-acier", "desco", "éditeur", "education", "électronique", "en", "ens", "cachan", "entreprise", "essai", "fabrication", "flash", "flexion", "formage", "format", "formation continue", "forum", "france", "freinage", "gaël balduini", "génie mécanique", "généiemécanique", "gestion", "gps", "historique", "igen-sti", "images", "informatique", "industrielle", "inventor", "jean-baptiste vidal", "lang", "logiciel", "logistique", "loi matériel", "mailinglist", "maintenance industrielle", "matériel-didactique", "mecamedia", "mécanique", "men", "mesr", "microtechniques", "ministère", "moteur", "option-bois", "otua", "outil", "papier", "pédagogie", "perspectives", "pgm", "photos", "pignon", "portail", "portail académique", "production", "productique", "productique mécanique", "professeur", "qualité", "recyclage", "référentiels", "ressources", "robot", "simulation", "solidworks", "sti", "structures métalliques", "support".

On the right side of the page, there is a section titled "linked tags (plus)" which lists tags such as "avion", "a380", "airbus", "a340", "a380", "boeing", "777", "boeing777", "bac", "sti", "option-bois", "techno", "sti", "option-bois", and "bacsti".

Figura 2.4. Salida de SmartDraw²⁶

²⁶ <http://sourceforge.net/projects/semanticscuttle/>

- **TagCrowd**²⁷

TagCrowd es una aplicación web que permite la visualización de tags mediante la creación de nube de etiquetas. Fue creado en julio de 2006 por Daniel Steinbock.

TagCrowd se especializa en hacer nubes de etiquetas fáciles de leer y analizar, funciona en línea (online), pero la desventaja es que no permite descargar. Las nubes de etiquetas creadas con esta herramienta son excelentes, presenta una imagen informativa que comunica bastante con solo una mirada.

Las siguientes tablas muestran las características funcionales y técnicas de la herramienta TagCrowd como son:

Tabla 2.11. Características Funcionales de TagCrowd

Personalizable nubes HTML	Se puede personalizar algunos de los estilos CSS para adaptarse al estilo de su página web. Estilos personalizados incluyen tamaño de la fuente y la fuente, el tamaño de la nube en general, los márgenes, relleno, bordes y color de fondo.
Límites de tamaño de archivo	Se puede ingresar archivos de texto muy grande en TagCrowd utilizando el botón "Upload a File". El límite es de 6 megabytes.
Guardar nubes como PDF	Se puede guardar nubes de texto en formato PDF. Eso significa que son escalables para imprimir en cualquier tamaño.

²⁷ <http://www.tagcrowd.com/>

Tabla 2.12. Características Técnicas de TagCrowd

Sistemas Operativos	Windows
Licencia	Es gratis para uso no-comercial
APIs	Si posee API

Salida



Figura 2.5. Salida de TagCrowd²⁸

- **TagCanvas²⁹**

TagCanvas es una clase de JavaScript que permite dibujar y animar una nube de etiquetas basado en HTML5. Canvas es un elemento HTML que puede ser utilizado para representar algo visualmente en tu navegador como por ejemplo: interfaz de usuarios elegantes, animaciones, gráficos y tablas, etc., esto se hace usando secuencias de comandos (normalmente JavaScript).

El uso del elemento canvas no es muy difícil pero sí es necesario un conocimiento básico de HTML y JavaScript. Canvas fue iniciado por Apple para Mac OS X y en la actualidad esta implementado en Safari y Google Chrome.

²⁸ <http://www.tagcrowd.com/>

²⁹ <http://www.goat1000.com/tagcanvas.php>

Las siguientes tablas muestran las características funcionales y técnicas de la herramienta TagCanvas como son:

Tabla 2.13. Características Funcionales de TagCanvas

Carga	La carga es más rápida, a diferencia si se cargara una etiqueta de imagen.
Conocimiento	El conocimiento de HTML y JavaScript se incrementa.
Compatibilidad	Funciona en navegadores con versiones actuales, es limitante con versiones muy antiguas.

Tabla 2.14. Características Técnicas de TagCanvas

Sistemas Operativos	Windows, Linux, Mac OS X
Licencia	Código Abierto (LGPL v3)
APIs	API Canvas

Salida



Figura 2.6. Salida de TagCanvas³⁰

³⁰ <http://www.goat1000.com/tagcanvas.php>

Herramientas para Mapas de Conocimiento

Las siguientes herramientas que se mencionan permiten la creación de mapas de conocimiento:

- **InfoRapid KnowledgeMap³¹**

InfoRapid KnowledgeMap es un completo sistema de gestión de conocimiento consta de dos componentes: el primero de edición de imagen (dibujo), programa para crear gráficamente sofisticados mapas y el segundo un servidor inteligente de KnowledgeMap, que analiza el contenido de texto y lo integra con un mapa de conocimiento. InfoRapid KnowledgeMap también es una gran herramienta para la recopilación de información.

El Mapa de Conocimiento contiene una descripción gráfica de cada documento con los elementos más importantes y extractos de textos. Los elementos se ordenan jerárquicamente y se agrupan por tipo de contenido. El propósito de la descripción de contenido es para darle una buena idea en el menor tiempo posible de la naturaleza.

Las siguientes tablas muestran las características funcionales y técnicas de la herramienta InfoRapid KnowledgeMap como son:

Tabla 2.15. Características Funcionales de InfoRapid KnowledgeMap

Componentes	Consta de dos componentes: uno de edición de imágenes (dibujo) y el inteligente servidor KnowledgeMap.
Importar / Exportar	Puede importar documentos de texto en HTML, RTF, PDF, analizar y resumir en un mapa de conocimiento.
Versiones	Edición Privada. Está disponible sin costo alguno, sólo se puede utilizar para uso privado, sin fines comerciales. Edición Profesional. Para fines comerciales o de gobierno.

³¹ <http://www.inforapid.de/HTML/knowledgemapeng.htm>

Tabla 2.16. Características Técnicas de InfoRapid KnowledgeMap

Lenguajes	XML
Sistemas Operativos	Windows
Licencia	CódigoAbierto (Open Source)
APIs	No existe

Salida

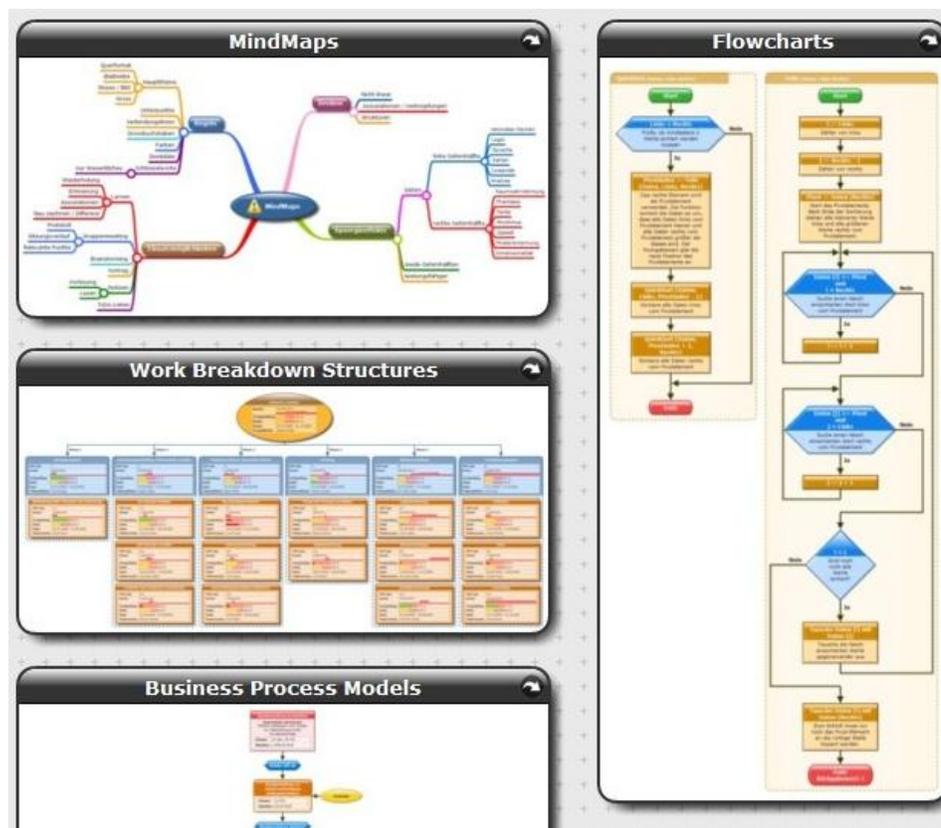


Figura 2.7. Salida de InfoRapid KnowledgeMap³²

- **Knowledge Master(KM)**³³

KM es una excelente herramienta para construir mapas mentales, mapas conceptuales, mapas semánticos, redes semánticas, bases de conocimiento conceptual y diagramas con una extensión funcional que facilita la gestión activa e interactiva del conocimiento.

³² <http://www.knowledgemaps.info/indexeng.htm>

³³ <http://www.knowledgemaster.us/default-esp.htm>

KM, la herramienta para el aprendizaje visual, activo y colaborativo, también permite asociar documentos a los conceptos, completando la capacidad de transferencia del conocimiento conceptual a los estudiantes.

Las siguientes tablas muestran las características funcionales y técnicas de la herramienta Knowledge Master como son:

Tabla 2.17. Características Funcionales de Knowledge Master

Recursos de productividad	La utilización de la interfaz gráfica de KM, es intuitiva y especialmente orientada a la interacción con los mapas, aumenta la productividad y permite aún a los usuarios ocasionales apropiarse rápidamente del instrumento.
Búsquedas	Con KM es posible realizar la búsqueda en Internet, asociando los resultados a los conceptos del mapa. Realizar búsquedas semánticas y búsquedas full text.
Exportación de contenidos	Exportar el mapa a un archivo XML. Exportar el mapa a un file textual en modo outline, incluyendo las anotaciones, en texto simple o formato RTF o Word.
Gestión del conocimiento	Organizar los conceptos en tipos (categorías) de conceptos, y definir aspectos gráficos específicos para los conceptos pertenecientes a los tipos distintos. Crear instancias de conceptos. Crear y consultar bases de conocimiento conceptual.

Tabla 2.18. Características Técnicas de Knowledge Master

Sistemas Operativos	Windows
Licencia	Código Cerrado (Not Open Source)
APIs	No existe

Salida

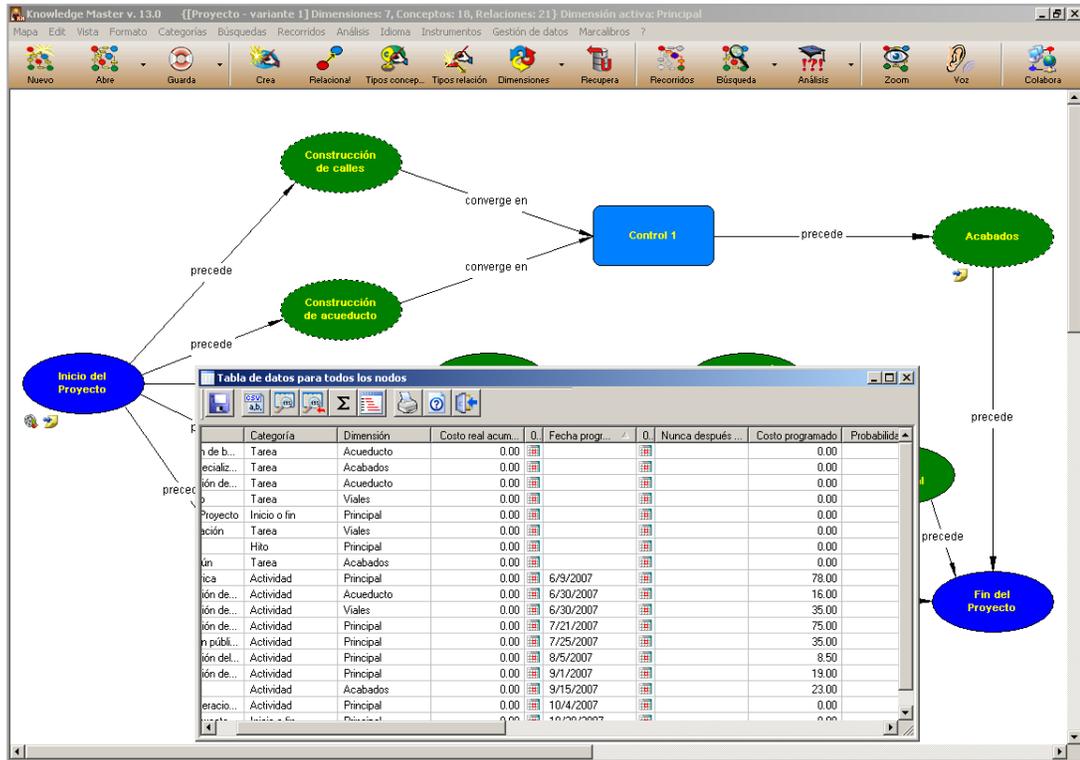


Figura 2.8. Salida de Knowledge Master³⁴

³⁴ <http://www.conceptmaps.it/KM-Project-Screen-esp.htm>

2.6. COMPARATIVA DE LAS HERRAMIENTAS

La siguiente tabla comparativa muestra un resumen de todas las herramientas que se menciona anteriormente, al finalizar se escogerá la mejor herramienta que se adapte a la solución.

Tabla 2.19. Comparativa Resumen de las Herramientas para Representar el Conocimiento.

Categorías	Herramientas	Ventajas	Desventajas
Mapas Conceptuales	<ul style="list-style-type: none"> • CmapTools 	<ul style="list-style-type: none"> • Es software libre y gratuito. • Facilidad de uso. • Puede ser administrada por cada usuario. • Permite usar el mapa y hacer búsquedas en google. • Permite la exportación a OWL, RDF, XML, N3, etc. 	<ul style="list-style-type: none"> • Tarda mucho tiempo en construir los mapas conceptuales. • No es integral, le faltan algunas herramientas didácticas. • No asocia las notas a los conceptos. • Al exportar los mapas como páginas web las anotaciones, los nodos anidados y los comentarios, desaparecen.
	<ul style="list-style-type: none"> • Inspiration 	<ul style="list-style-type: none"> • Permite crear un documento en modo de texto y posteriormente se puede convertir en diagrama o viceversa. • Proporciona muchas herramientas distintas para facilitar la captación y organización de ideas en diagramas. • Permite la exportación a gráficos, PDF. 	<ul style="list-style-type: none"> • Software no gratuito solo está en modo de prueba. • La exportación no permite a formatos de redes semánticas (OWL, RDF, etc.).

	<ul style="list-style-type: none"> • SmartDraw 	<ul style="list-style-type: none"> • Posee bastantes librerías de imágenes. • Creación de diagramas potentes y seguros. • Trabaja con Microsoft Office. 	<ul style="list-style-type: none"> • La compatibilidad: no es compatible con las aplicaciones más comunes. • Consume bastantes recursos. • La instalación es demasiada lenta. • No es gratuito, solo modo de prueba por siete días. • La exportación no permite a formatos de redes semánticas (OWL, RDF, etc.).
Mapas de Etiquetas	<ul style="list-style-type: none"> • SemanticScuttle 	<ul style="list-style-type: none"> • Es software libre y gratuito. • Facilidad de uso. • Compatibilidad con más de un sistema de Bases de Datos. • Crea metadatos sobre las páginas Web. • Integración auto clasificación de Contenidos. • Código modificable. • Es rápida al cargarse en el navegador. 	<ul style="list-style-type: none"> • No permite la generación de script de nubes de etiquetas para implementarse en otras plataformas. • La exportación no permite a formatos de redes semánticas (OWL, RDF, etc.).
	<ul style="list-style-type: none"> • TagCrowd 	<ul style="list-style-type: none"> • Es gratuito y fácil de usar. • Permite personalizar los estilos CSS de las nubes de etiquetas. 	<ul style="list-style-type: none"> • Solo funciona en línea. • Compatibilidad con los lenguajes de programación y los motores de bases de datos. • Demora en la carga de datos. • No se adapta a aplicaciones de google.

	<ul style="list-style-type: none"> • TagCanvas 	<ul style="list-style-type: none"> • Es gratuito y bastante funcional. • Crea atractivas las nubes de etiquetas. • Es personalizable, mediante código HTML y JavaScript permite dibujar y animar una nube de tags. • Carga rápidamente las tags. • Compatibilidad con más de un sistema de Bases de Datos y lenguajes de programación. • Compatibilidad con los navegadores web. • Permite descargar las librerías y utilizar en cualesquier aplicación. 	<ul style="list-style-type: none"> • Conocimiento básico de HTML y JavaScript. • Tiene limitantes en navegadores con versiones antiguas.
Mapas de Conocimiento	<ul style="list-style-type: none"> • InfoRapid KnowledgeMap 	<ul style="list-style-type: none"> • Es gratuito. • Cuenta con un servidor inteligente de conocimiento. • Permite importar documentos de textos a HTML, PDF y resumir en un mapa de conocimiento. 	<ul style="list-style-type: none"> • No permite utilizar conocimiento semántico para crear mapas de conocimiento. • La exportación no permite a formatos de redes semánticas (OWL, RDF, etc.).
	<ul style="list-style-type: none"> • Knowledge Master 	<ul style="list-style-type: none"> • Interfaz gráfica intuitiva y orientada a la interacción con los mapas. • Permite realizar búsquedas semánticas. • Exportar el mapa a un archivo XML. • Crear y consultar bases de conocimiento conceptual. 	<ul style="list-style-type: none"> • Software comercial, se debe comprar una licencia para su uso. • No permite la salida de OWL, RDF, etc.

Al finalizar el capítulo II luego de analizar las diferentes herramientas para crear mapas conceptuales, nubes de etiquetas y mapas de conocimiento, en donde la herramienta TagCanvas seleccionada para utilizarla en el desarrollo de la aplicación, posee las siguientes características notables:

- Es un componente creado utilizando etiquetas HTML y JavaScript, lo que lo hace compatible con entornos existentes.
- No requiere plugins para la visualización.
- Puede ser generado dinámicamente y a su vez el contenido puede ser modificado.
- Brinda al usuario una mayor experiencia visual, así como mayor capacidad para interacción.

**CAPÍTULO III:
REPRESENTACIONES DEL
CONOCIMIENTO DESDE
ESTRUCTURAS SEMÁNTICAS**

REPRESENTACIONES SEMÁNTICAS

La representación del conocimiento utilizando estructuras semánticas es utilizada hoy en día para el intercambio de datos en la Web, por tal razón se analizan los conceptos de RDF, su sintaxis y los diferentes lenguajes de consulta y herramientas que se utilizan para representar el conocimiento.

3.1. SEMÁNTICA

La Semántica es la parte de la Gramática que se ocupa del significado de los signos lingüísticos o palabras, expresiones o representaciones formales, a los que denomina lexías.

La semántica, no es el estudio lexical de las palabras, ya que considera que la unidad fundamental del discurso y del significado es la frase.

3.1.1. WEB SEMÁNTICA³⁵

La Web Semántica es una web con más significado y semántica que la semántica gramatical y, por lo tanto, se pueden obtener soluciones a problemas habituales en la búsqueda de información.

La forma en la que se procesará esta información no sólo será en términos de entrada y salida de parámetros sino en términos de su SEMÁNTICA.

Una web semántica es una web que contiene mucha información donde podemos relacionar diferentes recursos. Con la misma podemos solucionar problemas habituales como son los de realizar búsquedas en Internet.

La web semántica se basa en la idea de añadir metadatos semánticos y ontológicos a la Web. El objetivo es mejorar Internet ampliando la interoperabilidad entre los sistemas informáticos.

³⁵ <http://www.w3c.es/divulgacion/guiasbreves/websemantica>

3.2. RDF³⁶

(Resource Description Framework) [16]. Es un modelo estándar basado en XML, recomendado por la World Wide Web Consortium (W3C) para el intercambio de datos en la Web, con el fin de crear un formato que permitiera alcanzar la compatibilidad entre los diversos sistemas de metadatos, suministrando para ello una arquitectura genérica de meta información.

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

RDF utiliza el lenguaje XML como método para representar y transformar la información. RDF extiende la estructura de vinculación de la Web utilizando el nombre de la URI a través de la relación entre las cosas uniendo los dos extremos del enlace (conocido normalmente como "tripleta"). La capacidad que tiene RDF para procesar metadatos facilita la interoperabilidad entre diversas aplicaciones, proporcionando un mecanismo perfecto de intercambio de información a través del Web.

Definiciones del Modelo RDF

- "Sistema que permite la interoperabilidad entre aplicaciones mediante el intercambio de información legible por ordenador a través del Web" (*Brickley, Dan y Guha, R. V., 2000*).
- "Mecanismo que facilita la automatización de procesos susceptibles de ser realizados con recursos Web" (*Lassila, Ora, 1998*).

Características de RDF

La siguiente tabla según [16] muestra las diferentes características de RDF como son:

³⁶ <http://www.w3.org/TR/RDF-primer/>

Tabla 3.1. Características de RDF

Independencia	Dado que una propiedad es un recurso, toda organización independiente o incluso cada persona puede inventarlas.
Intercambio	Dado que las sentencias RDF se escriben en XML pueden ser fácilmente usadas para intercambiar información.
Escalabilidad	Las sentencias RDF son simples, registros con tres campos (Recurso, propiedad, valor) por lo que son fáciles de manejar y de usar para buscar objetos aun en volúmenes realmente grandes. La Web continúa creciendo constantemente, por eso la escalabilidad es importante.
Reificación.	La reificación se basa en poder describir otras declaraciones utilizando RDF.

3.2.1. COMPONENTES DE RDF

Los componentes de un RDF son según [17]:

- **URIs**
Las URIs sirven para hacer referencia a los recursos
- **Literales**
 - Sirven para representar los valores de datos
 - Interpretado por medio de tipos de datos
- **Los nodos vacíos**

Hablando de algo que no tiene un nombre o no se conoce su identidad.

3.2.2. EL MODELO DE DATOS RDF

Es un modelo para representar propiedades designadas y valores de propiedades. El modelo RDF se basa en principios perfectamente establecidos de varias comunidades de representación de datos. Las propiedades RDF representan también la relación entre recursos y por lo tanto, un modelo RDF puede parecer un diagrama entidad-relación.

El modelo de datos de RDF es una forma de sintaxis-neutral para representar expresiones RDF. La representación del modelo de datos se usa para evaluar la equivalencia en significado.

Este modelo de datos proporciona un marco conceptual para definir y utilizar metadatos, pero para definir metadatos necesita una sintaxis concreta para crear e intercambiar metadatos. Este modelo se forma de objetos (Recursos, Propiedades y Sentencias).

El modelo de datos según [18] consiste en tres tipos de objetos:

1. **Recursos:** Son nodos dibujados por óvulos o círculos. Todas las cosas descritas por expresiones RDF se denominan *recursos*. Todo recurso es identificado por una URI (Uniform Resource Identifier), un recurso puede ser una página web, un documento HTML, una imagen, etc.
2. **Propiedades:** Representadas mediante flechas o arcos. Una propiedad es una característica, atributo, o relación utilizado para describir un recurso. Cada propiedad tiene un significado específico, define los valores permitidos, los tipos de recursos y sus relaciones con otras propiedades.
3. **Sentencias:** Una sentencia es un recurso junto con una propiedad, más el valor de dicha propiedad para ese recurso es una sentencia RDF. Estas tres partes individuales de una sentencia como son sujeto, predicado y objeto.
 - Sujeto representa a un recurso específico.
 - Predicado representa a una propiedad.
 - Objeto representa el valor de la propiedad y puede ser otro recurso identificado entonces por una URI o bien un valor literal. Los nodos que representan cadenas de literales se representan con rectángulos.

El sentido de las flechas es importante. El arco o flecha siempre empieza en el sujeto y apunta hacia el objeto de la sentencia.

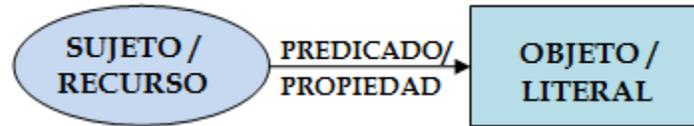


Figura 3.1. El modelo RDF

3.2.3. FORMAS DE REPRESENTACIÓN DE RDF

El modelo de datos RDF proporciona un marco abstracto y conceptual para definir y utilizar metadatos. Necesita también una sintaxis concreta para crear e intercambiar metadatos.

En este apartado se muestran algunas formas según [17] de representar el modelo de datos de RDF y partiendo de un ejemplo con la siguiente descripción:

Juan es una persona

- **Grafo RDF**

Un grafo representado en forma de gráfico de nodos y flechas, se basa en el modelo RDF en tripleta donde un **sujeto** (el recurso) tiene un **predicado** (propiedad) con un **objeto** determinado (recurso o literal).

Los **recursos** se representan por óvalos, las **propiedades** son las flechas o arcos etiquetados que unen los nodos (sujeto-objeto) y los **objetos** son nodos representados por óvalos (si es un recurso) o por un rectángulo (si es un literal).

El grafo puede ir creciendo conforme se le añadan más elementos a la descripción del recurso.



Figura 3.2. Grafo RDF Representación

- **Notation 3 (N3)**

Notation 3 según [19] se codifica en UTF-8 bytes [20] y su gramática está formada por sentencias que pueden ser directivas o declaraciones.

Las directivas usan el prefijo "@**prefix**" y proveen información para el acceso al vocabulario mediante namespace. N3 es mucho más compacto y fácil de leer que la notación XML RDF, además N3 sirve como el soporte para RDF basados en reglas.

La declaración está formada por sujeto, predicado y objeto, y se pueden hacer referencias a una URI. Sin embargo sólo el objeto puede ser además un literal.

```
@prefix : < http://www.prueba.org/Juan#> .
@prefix s: < http://www.prueba.org/schema#>.
:Juan a s:persona;
```

N3 presenta una ventaja significativa que es más fácil de escribir a mano y seguir, porque está basado en notación tabular.

Entonces, una tripleta se representa mediante nodos conectados por líneas con etiquetas. Los nodos representan recursos y las líneas con etiquetas las propiedades de esos recursos. Los 3 elementos de una tripleta se representan mediante URIs.

- **N-Triples**

El W3C desarrolló N-triples [21], un formato para una representación de RDF que está especialmente preparado para las colecciones de la prueba

ante las dificultades que provoca la flexibilidad de XML como sintaxis (XML/RDF) al querer comparar resultados de procesos de testing automatizado.

N-Triples es un subconjunto fijo de N3. Esta notación es muy simple, consiste en indicar el sujeto, predicado y objeto entre corchetes y en este preciso orden. Además al final de cada sentencia se termina con un punto “.”.

En realidad un documento de N-triples es una sucesión de triples de RDF formada del sujeto, predicado y objeto. Una sintaxis de N-triples ofrece dos variantes, con prefijo (@prefix) o sin él (formato estándar):

Con @prefix:

Esta variante es similar a la sintaxis N3, en donde primero se declaran los esquemas a utilizar mediante prefijos y namespaces, para luego mostrar las descripciones usando los prefijos declarados.

```
@prefix : < http://www.prueba.org/Juan# > .  
@prefix s: < http://www.prueba.org/schema#> .  
@prefix RDF: < http://www.w3.org/1999/02/22-RDF-syntax-ns#> .  
:JuanRDF:type s:persona;
```

En formato estándar:

El formato estándar ofrece una siguiente estructura:

- Si el objeto es un recurso
<URI sujeto><URI predicado><URI objeto>
- Si el objeto es un literal
<URI sujeto><URI predicado>”objeto”

Esta sería la representación del ejemplo utilizando esta sintaxis:

```
<http://www.prueba.org/Juan#Juan><http://www.w3.org/1999/02/22-RDF-syntax-s#type><http://www.prueba.org/schema#persona>
```

- **RDF/XML**

La sintaxis RDF/XML es la que ofrece mayor interoperabilidad semántica porque todos los datos permiten el acceso a su significado mediante su correspondiente namespace. Además al usar la sintaxis XML se hereda la forma representación del modelo de datos utilizando la expresividad e interoperabilidad sintáctica. Esta sería la representación del ejemplo:

```
<RDF:RDFXMLns="http://www.prueba.org/Juan#"
XMLns:s="http://www.prueba.org/schema#"
XMLns:RDF="http://www.w3.org/1999/02/22-RDF-syntax-ns#">
<s:personaRDF:about="#Juan">
</s:persona>
</RDF:RDF>
```

3.2.4. LENGUAJES DE CONSULTA PARA RDF

Una vez que la información ha sido almacenada en repositorios RDF se necesita de un lenguaje de consulta para sacar esa información. Existen tres tipos de niveles que se pueden analizar para los documentos RDF [18] se pueden analizar a tres niveles diferentes:

- A nivel sintáctico.
- A nivel semántico.
- A nivel de estructura.

A **nivel sintáctico** todo documento RDF es a su vez documentos XML. Cualquier documento RDF podrá ser consultado usando un lenguaje de consulta para RDF (por ej. Xquery, SPARQL).

A **nivel estructural**, el documento RDF es un conjunto de tripletas. En donde se han propuesto e implementado varios lenguajes de consulta RDF que buscan y recuperan tales tripletas de varias formas (ej. Squish, SPARQL).

A **nivel semántico** los lenguajes de consulta son:

- **RQL**³⁷
RQL permite el uso de variables para denotar los nombres de nodos (es decir, las clases) y los arcos (es decir las propiedades). Una de las principales características de RQL, y que lo distinguen de otros lenguajes de consulta RDF es su capacidad de consultar esquemas RDF y descripciones RDF (es decir, instancias) en una misma consulta.

- **RDQL**³⁸ (desarrollado por HP en su API Jena2)
La sintaxis RDQL (RDF Data Query Language) es similar a la de SQL. RDQL se relaciona de cerca con SquishQL, el cual a su vez está basado en RDQL, un lenguaje de bases de datos escalables creadas para trabajar con Servicios Web Semánticos.

- **SeRQL** (sucesor de RQL) de Sesame
Sesame RDF Query Language está siendo desarrollado por Aduna pertenece al grupo de Sesame, combina las mejores características de otros lenguajes de consulta RDF como son RQL, RDQL, N-Triples y añade alguna característica propia. Entre sus ventajas está el que soporta consultas anidadas (RDQL no las soporta).

- **SPARQL**³⁹
Es un estándar en el que está trabajando actualmente W3C, un lenguaje de scripting derivado de JavaScript que incluye comandos similares a SQL. Este lenguaje se adapta totalmente a las necesidades de RDF.

Todos estos lenguajes se analizan con más detalle en el Capítulo V sección 4.3.

³⁷ <http://139.91.183.30:9090/RDF/RQL/index.HTML>

³⁸ <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>

³⁹ <http://www.w3.org/TR/RDF-SPARQL-query/>

3.2.5. HERRAMIENTAS

En esta sección se hace una descripción de algunas herramientas que permiten la creación de ontologías utilizando la estructura para un RDF.

- **Sesame⁴⁰**

El Sesame es una arquitectura genérica, implementada con software abierto, que permite el procesamiento de datos y esquemas RDF. Esto incluye el análisis, el almacenamiento y la consulta de estos datos. Ofrece una API fácil de usar que se pueden conectar a todas las soluciones de almacenamiento líderes RDF. Los tipos de consultas que permite esta herramienta son: RQL, RDQL, SeRQL y SPARQL.

Sesame ha sido diseñado pensando en la flexibilidad. Por eso, Sesame soporta todos los principales formatos de archivos RDF, incluyendo RDF/XML, Turtle, N-Triples, Trig y Trix.

Las siguientes tablas muestran las características funcionales y técnicas de la herramienta Sesame como son:

Tabla 3.2. Características Funcionales de Sesame.

Completamente dirigida a Java 5	Todas las API de Java 5 se caracteriza por el uso, tales como las colecciones de escritos e iteradores.
Apoyo para el contexto/ procedencia	Lo que le permite realizar un seguimiento de las distintas unidades de datos RDF (como archivos, para instancias).
Adecuada operación/ retroceso de apoyo.	
Apoyo al Lenguaje de Consulta SPARQL.	

⁴⁰ <http://www.openrdf.org/index.jsp>

Protocolo HTTP	Incluye soporte para el protocolo SPARQL y los resultados de la consulta SPARQL en formato XML. Propuesta de Recomendaciones del W3C.
-----------------------	---

Tabla 3.3. Características Técnicas de Sesame

Lenguajes	Java
Bases de Datos	SPARQL
Sistemas Operativos	Windows, Linux
Licencia	Licencia BSD-2 (Generar su propia licencia)
API	Open RDF Sesame Core 2.5.0 API

Salida

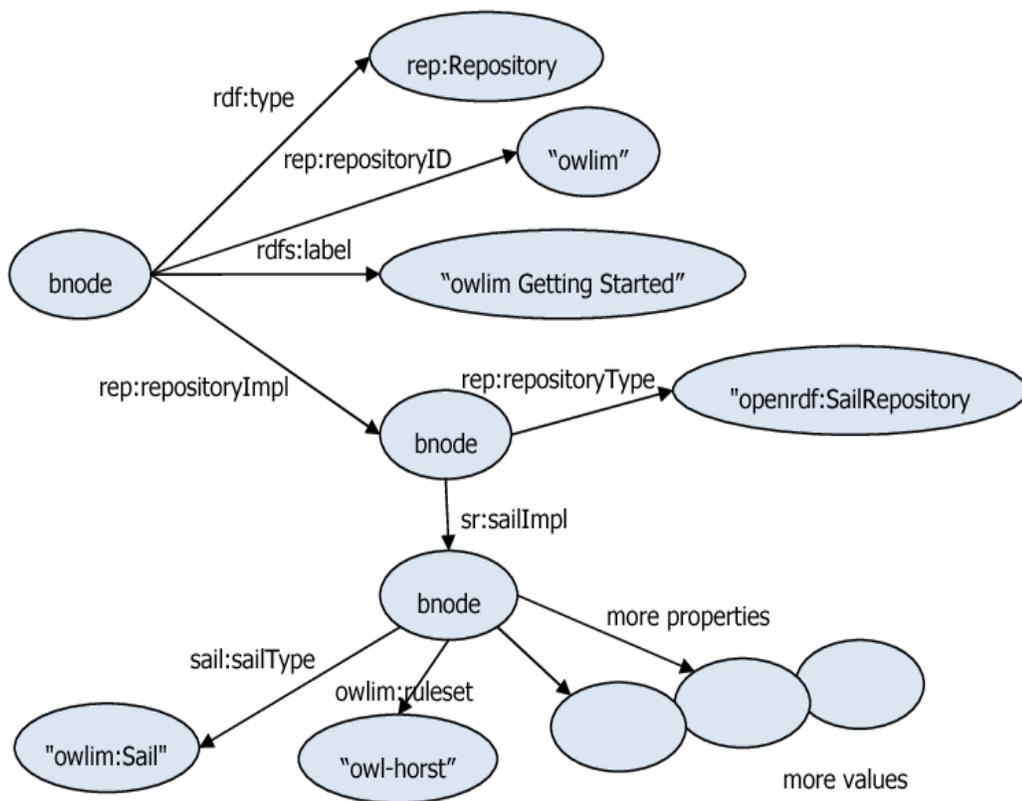


Figura 3.3. Salida de Sesame

- **Semantic Works Semantic Web⁴¹**

Altova Semantic Works 2011 es una herramienta editor de la Web Semántica innovadora para RDF y OWL. Gráficamente diseña instancias de documentos RDF, RDFS vocabularios y ontologías OWL, luego presenta salidas en RDF/XML o formatos de N-Triples.

Semantic Works a medida que se trabaja, se puede cambiar de la vista de diseño gráfico a la vista de texto para ver el correspondiente auto-generados RDF/XML o código N-triples, y se puede exportar el archivo de RDF/XML para N-triples o viceversa en cualquier momento. Esta herramienta permite la impresión de graficas de RDF y OWL.

Las siguientes tablas muestran las características funcionales y técnicas de la herramienta Semantic Works como son:

Tabla 3.4. Características Funcionales de SemanticWorks.

Creación de documentos visuales	Proporciona una interfaz intuitiva, gráfica y funcionalidad de arrastrar y soltar para crear y editar documentos RDF, RDFS, OWL Lite, OWL DL y OWL Full.
Editor Gráfico de RDF y RDFS	Ofrece un poderoso apoyo para el diseño y edición de documentos RDF y RDFS para implementar a aplicaciones de la Web Semántica de una manera intuitiva, visual.
Editor Gráfico de OWL	Permite exportar el mapa a un archivo XML, a un file textual en modo outline, incluyendo las anotaciones, en texto simple o formato RTF o Word.

⁴¹ <http://www.altova.com/semanticworks.HTML>

Documentación de la Web Semántica	Permite la impresión de graficas de documentos RDF y OWL, etc.
Generación de Código	Auto-genera código visual RDF/XML o N-triples basados en RDF/OWL.

Tabla 3.5. Características Técnicas de SemanticWorks.

Lenguajes	XML
Sistemas Operativos	Windows
Licencia	Código Cerrado (Free, 30-day trial)
API	No existe

Salida

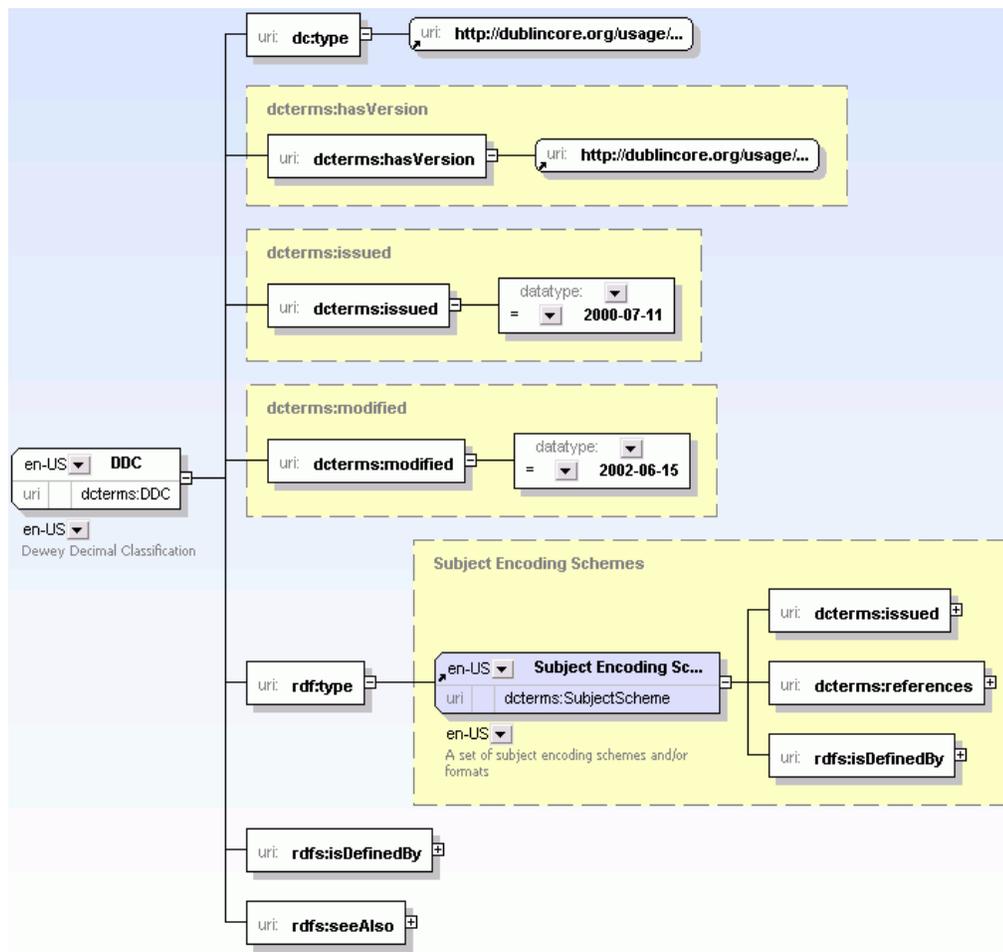


Figura 3.4. Salida de SemanticWorks⁴²

⁴² <http://www.altova.com/semanticworks/RDF-editor.HTML>

- **RDF Gravity**⁴³

RDF Grafity es una herramienta sencilla que permite la visualización gráfica de ontologías RDF/OWL. RDF Gravity utiliza una API grafica de JUNG y la herramienta de web semántica Jena.

La herramienta proporciona una visualización simple y poderosa de las estructuras de grafo RDF y la capacidad de filtrar y visualizar partes o fragmentos de gráficos RDF.

Las siguientes tablas muestran las características funcionales y técnicas de la herramienta RDF Gravity como son:

Tabla 3.6. Características Funcionales de RDF Gravity.

<p>Visualización grafica</p>	<p>RDF Gravity define un paquete de visualización en la parte superior de la API grafica de JUNG.</p> <p>Este paquete proporciona las siguientes funciones:</p> <ul style="list-style-type: none"> • Procesadores configurables para los bordes y nodos de un grafo. • Modelo de selección múltiple • Gestión de eventos del ratón y Zoom.
<p>Filtros globales y locales</p>	<p>La herramienta permite al usuario especificar los filtros globales y locales para tener puntos de vista específicos sobre el gráfico.</p>
<p>Búsqueda de texto completo</p>	<p>RDF Gravity proporciona una búsqueda de texto completo sobre los conceptos, propiedades y casos especificados en un archivo RDF.</p>
<p>Visualización de varios archivos RDF</p>	<p>RDF Gravity permite al usuario ver múltiples estructuras grafica contenida en varios archivos RDF a través de una única vista.</p>

⁴³ http://semweb.salzburgresearch.at/apps/RDF-gravity/user_doc.HTML

Es una plataforma que puede ser extendida con gráficos, diagramas, componentes animados para acceder a aplicaciones embebidas en sistemas de bases de conocimientos.

La plataforma Protégé soporta dos formas principales de modelado de ontologías a través de **Protégé-Frames** y editores **Protégé-OWL**.

Protégé pueden ser exportados a una variedad de formatos, incluyendo RDF, OWL y XML Schema, etc.

Las siguientes tablas muestran las características funcionales y técnicas de la herramienta Protégé como son:

Tabla 3.8. Características Funcionales de Protege.

Personalizable	Protégé puede ser personalizado para ofrecer un dominio fácil de apoyo para la creación de modelos de conocimiento y la introducción de datos. Posee gran cantidad de módulos para mejorar las funcionalidades de la herramienta.
Soporta Ontologías de modelado	<ul style="list-style-type: none"> • Protégé-Frames permite a los usuarios construir y poblar las ontologías que están basadas en fotogramas. • Protégé-OWL permite a los usuarios construir ontologías para la Web Semántica.
Versión Actual	Protégé 4.1 está disponible desde el 26 julio del 2011.

Tabla 3.9. Características Técnicas de Protege.

Lenguajes	Java.
Sistemas Operativos	Windows, Unix, Linux, Mac OS X.
Licencia	Código Abierto (Open Source).
APIs	API Protégé-Frames y API Protégé-OWL.

3.3. COMPARATIVA DE LAS HERRAMIENTAS

La siguiente tabla comparativa muestra un resumen de todas las herramientas que se menciona anteriormente, al finalizar se escogerá la mejor herramienta que se utilizara para la creación del documento RDF.

Tabla 3.10. Comparativa Resumen de las Herramientas para RDF.

Herramientas	Formatos que Soporta	Características Importantes
<ul style="list-style-type: none"> Sesame 	<ul style="list-style-type: none"> RDF, RDF/XML, Turtle, N-Triples, Trig y Trix 	<ul style="list-style-type: none"> Es abierto y gratuito Escalabilidad. Soporta el lenguaje SeRQL, SPARQL. Permite la exportación de esquemas completos o datos desde repositorios con formato RDF serializados XML. Esta implementado en Java. Consumo de tiempo en ejecución de una consulta.
<ul style="list-style-type: none"> Semantic Works 	<ul style="list-style-type: none"> RDF, RDF/XML, RDFS, OWL Lite, OWL DL y OWL Full. 	<ul style="list-style-type: none"> Es de uso comercial y la Interfaz es fácil de usar Escalabilidad. Utilizada para crear y editar documentos RDF, OWL, etc. No soporta lenguajes de consulta.
<ul style="list-style-type: none"> RDF Gravity 	<ul style="list-style-type: none"> RDF, RDF/XML, N-Triples. 	<ul style="list-style-type: none"> Es abierto y gratuito Utilizada para visualización grafica de ontologías RDF/OWL, pero no permite la creación o edición de ontologías. Permite especificar los filtros globales y locales para tener puntos de vista específicos sobre el gráfico.

<ul style="list-style-type: none"> • Protégé 	<ul style="list-style-type: none"> • OWL, RDF/XML, OWL/XML, OWL Funtional Syntax, Latex, Turtle. 	<ul style="list-style-type: none"> • Es abierto y gratuito. • Interfaz fácil de usar e intuitiva. • Facilita el trabajo con las clases y las instancias. • Ayuda a visualizar gráficamente ontologías. • Permite el trabajo colaborativo y su reutilización • Protege OWL se integra perfectamente con herramientas de clasificación. • OWLViz y Ontoviz pueden ser usados para visualizar gráficamente ontologías OWL. • Mayor rapidez en la introducción de datos, puesto que permite capturar todos los datos previamente introducidos. • No utiliza un lenguaje específico de interrogación, lo cual incrementa su sencillez.
---	---	--

Al finalizar el capítulo III, se analizan las diferentes herramientas para crear ontologías y documentos RDF, llegando a la conclusión que la herramienta Protégé es la más idónea para la solución de este proyecto de tesis, puesto que destaca en los siguientes aspectos.

- Permite el trabajo tanto con el formato OWL como RDF en sus diversas variaciones como turtle.
- Creación de forma gráfica de clases, propiedades de datos y objetos.
- Representación gráfica automática de la ontología
- Creación de individuos (población) de manera sencilla, integrando razonadores para comprobar la validez de los mismos.

CAPÍTULO IV:

DISEÑO DE LA SOLUCIÓN

DISEÑO DE LA SOLUCIÓN

En este capítulo se menciona con detalle el diseño de la solución para el desarrollo del mapa de conocimiento del gestor bibliográfico de la UTPL, en donde se explican los requerimientos necesarios, la arquitectura a utilizar, el lenguaje de consulta, así como las diferentes tecnologías involucradas.

4.1. REQUERIMIENTOS

La construcción de la plataforma semántica para la representación de información bibliográfica de los centros universitarios de la Universidad contempla los siguientes aspectos:

- Recolección de información de las bibliotecas de los centros universitarios de la UTPL.
- La información relativa a bibliografía debe ser migrada a tripletas RDF y almacenada en un sitio, que permita su recuperación.
- Se debe considerar la creación de un mecanismo de comunicación entre el almacén de tripletas y el aplicativo web que visualiza el mapa con la representación de la información.
- Se debe llevar un registro de los recursos más visitados por los usuarios, estos registros podrán ser utilizados en aplicativos futuros para la recomendación de contenido.

De esta forma se detalla a continuación los diversos casos de uso tanto para la extracción de información como para la construcción del aplicativo, considerando la extracción de información como la representación.

4.1.1. CASOS DE USOS

El funcionamiento del aplicativo GERBI-MAP puede ser apreciado mediante el análisis de los Casos de Uso que se presentan a continuación:

- **OBTENCIÓN DE INFORMACIÓN**

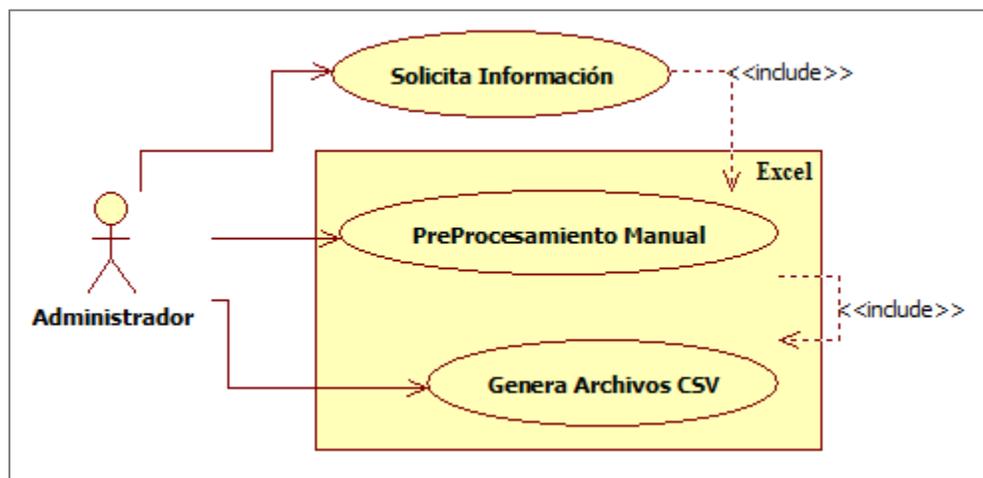


Figura 4.1: Vista de Caso de uso: Obtención de Información.

En la fase de recolección (obtención) de la información se puede apreciar las siguientes acciones:

Tabla 4.1. Especificación Caso de Uso: Obtención de la información.

Caso de Uso: Obtención de Información.	
Actores: Administrador.	
Actividades:	Solicitar Información, Pre-Procesamiento Manual y Generar los Archivos CSV.
Sinopsis:	El caso de uso comienza cuando el administrador solicita la información referente al material bibliográfico, luego de recibir la información se pasa al proceso de pre-procesamiento que consiste en realizar modificaciones manualmente de los datos. Finalmente luego de tener la información idónea para subir, el administrador exporta cada parte de los datos a archivos CSV.

Curso típico de eventos:	<ol style="list-style-type: none"> 1. Administrador solicita la información referente al material bibliográfico, obteniendo esta información en documentos no homogéneos (Excel, PDF). 2. Administrador realiza las modificaciones de los datos, hasta alcanzar una homogeneidad. 3. Administrador exporta cada parte de los datos a archivos separados por comas (CSV).
Extensiones:	<ul style="list-style-type: none"> • Si la información no está en archivos de Excel: convertir al formato requerido.

• **GENERACIÓN DE INDIVIDUOS**

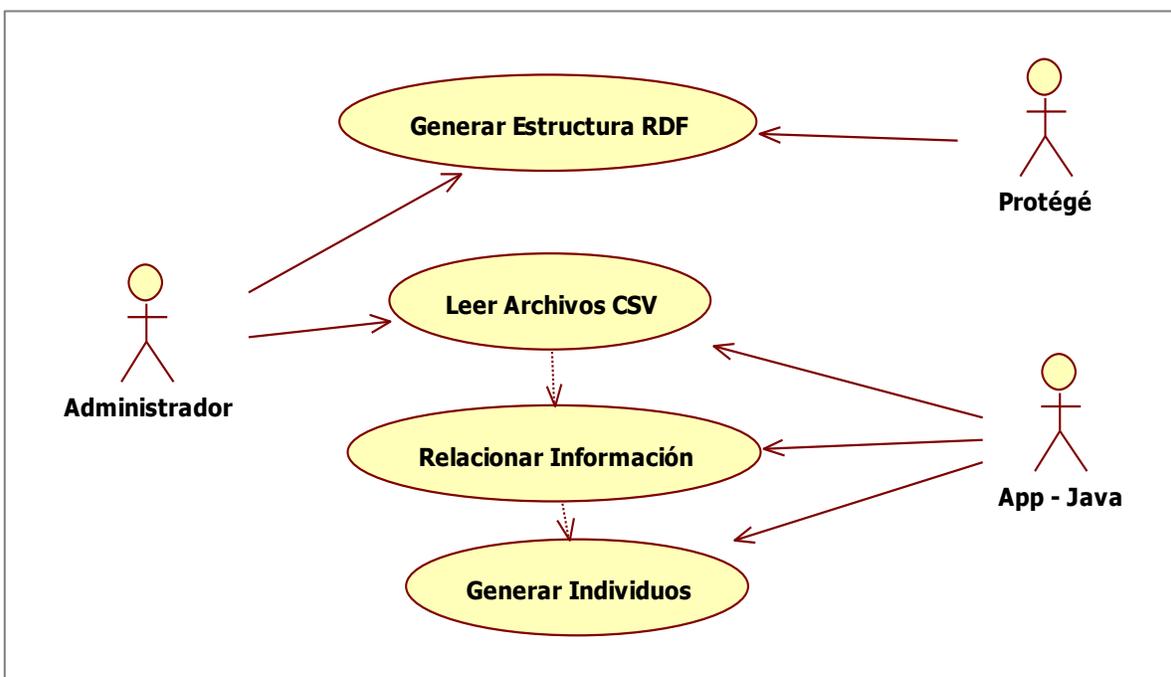


Figura 4.2. Vista de Caso de Uso: Generación de Individuos

El proceso de generación de Individuos en Tripletas RDF involucra lo siguiente:

Tabla 4.2. Especificación Caso de Uso: Generación de Individuos.

Caso de Uso: Generación de Individuos.	
Actores: Administrador, Protégé, App Java.	
Actividades:	Generar la Estructura del RDF, Leer los Archivos CSV, Relacionar la Información y Generar los Individuos.
Sinopsis:	El caso de uso comienza cuando el administrador genera una estructura básica de un individuo en Protégé, luego de tener la estructura se inicia la lectura de los archivos CSV generados desde el entorno Java; finalmente el aplicativo java relaciona la información de los archivos y genera los individuos en tripletas.
Curso típico de eventos:	<ol style="list-style-type: none"> 1. El Administrador se encarga de generar una estructura básica de un individuo en la herramienta Protégé, la misma que servirá de base para el proceso. 2. El Administrador inicia la lectura de los archivos CSV generados anteriormente desde un entorno de aplicación java. 3. El aplicativo Java relaciona la información de los archivos según parámetros definidos. 4. La aplicación Java genera los individuos en tripletas en base a la información procesada.
Extensiones:	<ul style="list-style-type: none"> • Si los archivos CSV a cargar no se encuentran o están en otro formato, el aplicativo java no generará los individuos.

- **CARGA DE DATOS**

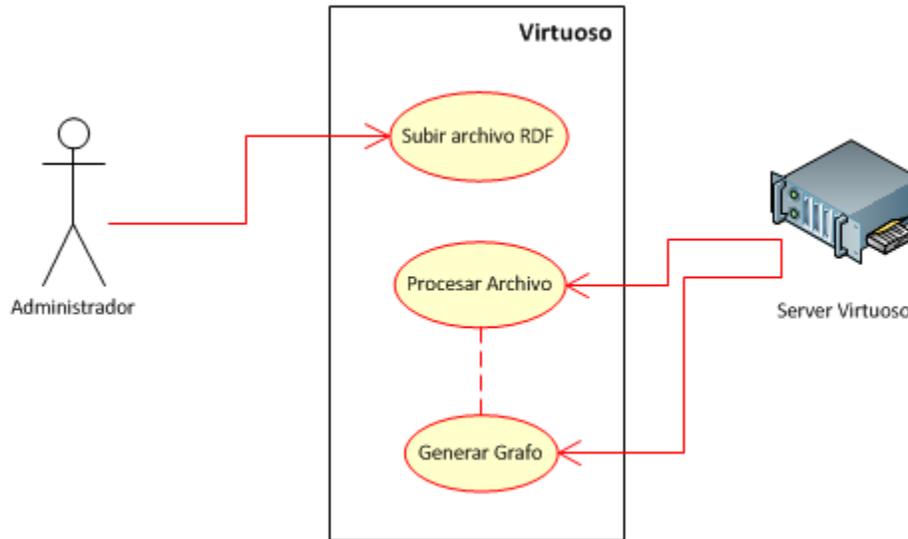


Figura 4.3. Vista de Caso de Uso: Carga de Datos

El proceso de cargar el archivo RDF al servidor Virtuoso involucra lo siguiente:

Tabla 4.3. Especificación Caso de Uso: Carga de Datos

Caso de Uso: Carga de Datos.	
Actores: Administrador, Server Virtuoso.	
Actividades:	Subir el archivo RDF, Procesar el archivo y Generar el grafo.
Sinopsis:	El caso de uso comienza cuando el administrador accede al servidor Virtuoso para subir el archivo RDF y generar el grafo que se utiliza para acceder a los datos.
Curso típico de eventos:	<ol style="list-style-type: none"> 1. El Administrador se encarga de subir el archivo RDF al servidor Virtuoso. 2. El servidor Virtuoso procesa el archivo y genera el grafo que es la dirección donde está almacenado el RDF, esta dirección se la utiliza en la aplicación GERBI-MAP.
Extensiones:	<ul style="list-style-type: none"> • Si el servidor virtuoso no está levantado o habilitado el puerto 8890 no se podrá acceder a los datos.

• VISUALIZACIÓN CENTROS UNIVERSITARIOS - MAPA

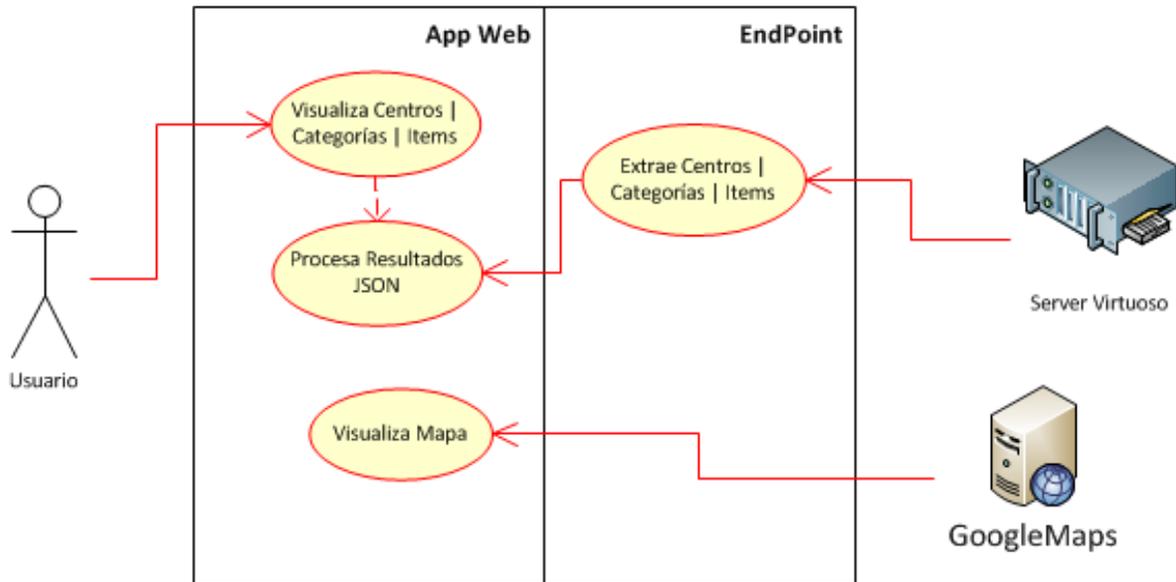


Figura 4.4. Vista de Caso de Uso: Visualización Centros universitarios - Mapa

El funcionamiento del aplicativo Web se detalla de la siguiente manera:

Tabla 4.4. Especificación Caso de Uso: Visualización Centros Universitarios – Mapa

Caso de Uso: Visualización Centros Universitarios – Mapa	
Actores: Usuario, GoogleMaps, Server Virtuoso.	
Actividades:	Visualizar el Mapa, Visualizar Centros Universitarios con las Categorías e Ítems, Extraer Centros, Categorías y Items, Procesar Resultados.
Sinopsis:	El caso de uso comienza cuando en el aplicativo web se muestra el mapa del Ecuador con los respectivos centros universitarios de la UTP, luego el usuario selecciona el centro para visualizar las categorías y los individuos existentes en cada centro asociado.

<p>Curso típico de eventos:</p>	<ol style="list-style-type: none"> 1. En el Aplicativo web se visualiza el mapa del Ecuador sobre la plataforma GoogleMaps. 2. El usuario observa los centros asociados, los mismos que son extraídos desde el EndPoint de Virtuoso mediante peticiones HTTP. 3. El usuario selecciona un centro asociado, la categoría y el ítem, los mismos que son extraídos desde el EndPoint de Virtuoso, en el aplicativo web se procesa los resultados mediante peticiones JSON.
<p>Extensiones:</p>	<ul style="list-style-type: none"> • Si no se carga el mapa del Ecuador: verificar el API key de la versión 2 de GoogleMaps. • Si no se muestran los centros universitarios y las categorías de ese centro: verificar las configuraciones del archivo getData.php.

• **REGISTRO - VISUALIZACIÓN DE ESTADÍSTICAS**

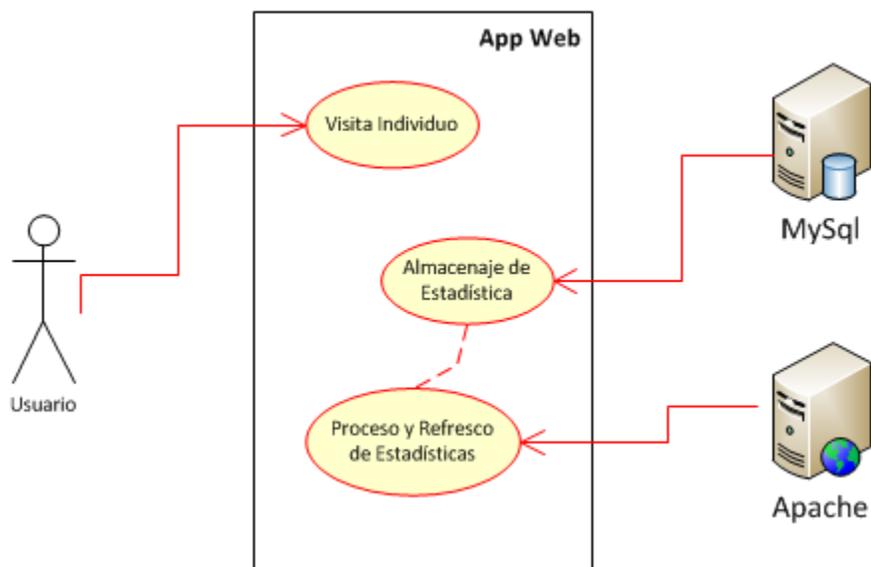


Figura 4.5. Vista de Caso de Uso: Registro - Visualización de Estadísticas

Para el registro y visualización de estadísticas se detalla de la siguiente manera:

Tabla 4.5. Especificación Caso de Uso: Registro - Visualización de Estadísticas

Caso de Uso: Registro - Visualización de Estadísticas	
Actores: Usuario, MySQL, Apache	
Actividades:	Visita Individuo, Almacén de Estadísticas y Actualización de las Estadísticas.
Sinopsis:	El caso de uso comienza cuando el usuario en el aplicativo web visita los individuos (Centro Asociado, Categoría, Ítem), y observa en la parte derecha del aplicativo GERBI-MAP como se incrementan las estadísticas de acuerdo a al tipo de visita.
Curso típico de eventos:	<ol style="list-style-type: none"> 1. En el Aplicativo web el usuario visualiza las estadísticas de los recursos visitados, relacionándose con un esquema de base de datos relacional sobre el motor MySQL. 2. El aplicativo web interactúa con la base relacional, registrando la estadística del recurso visitado por parte del usuario. 3. El servidor Apache interactúa con el aplicativo web en el proceso de actualización de las estadísticas para mostrarlas en la parte derecha del aplicativo.
Extensiones:	<ul style="list-style-type: none"> • Si el usuario selecciona un centro asociado: se incrementa la visita de ese centro. • Si el usuario selecciona una categoría: se incrementa el valor del individuo de esa categoría. • Si el usuario selecciona un ítem de una categoría: se incrementa el valor del elemento visitado. • Si no se actualizan las estadísticas: revisar las configuraciones del archivo ConexionBD.php.

4.2. ARQUITECTURA DEL PROYECTO

La arquitectura a utilizar para el desarrollo de la aplicación será una arquitectura en tres capas, por permitir mantener separados los ambientes de datos, proceso y presentación.

✓ **La Interfaz de Usuario (Capa de Presentación):**

Capa visible por el usuario, la misma que se encarga de permitir la interacción con la aplicación, aquí se ubican los componentes gráficos como el mapa y la visualización de los datos para la plataforma del gestor bibliográfico.

✓ **La Lógica (Capa de Aplicación):**

Se la considera una capa intermedia, puesto que esta se encarga de todo el procesamiento de la información proveniente de la capa inferior para ser mostrada en la capa superior. En este caso, como en muchos otros, esta se encuentra dentro del mismo ambiente que la capa de presentación, lo que no indica que sean dependientes.

✓ **El Almacenaje (Capa de Datos):**

Se encuentra conformada por los almacenes de datos, en donde se guardará la información necesaria para el funcionamiento de la plataforma. En este caso se utilizarán dos tipos diferentes, uno para almacenar información en formato de tripletas (TripleStore) y otro para almacenaje de datos relaciones.

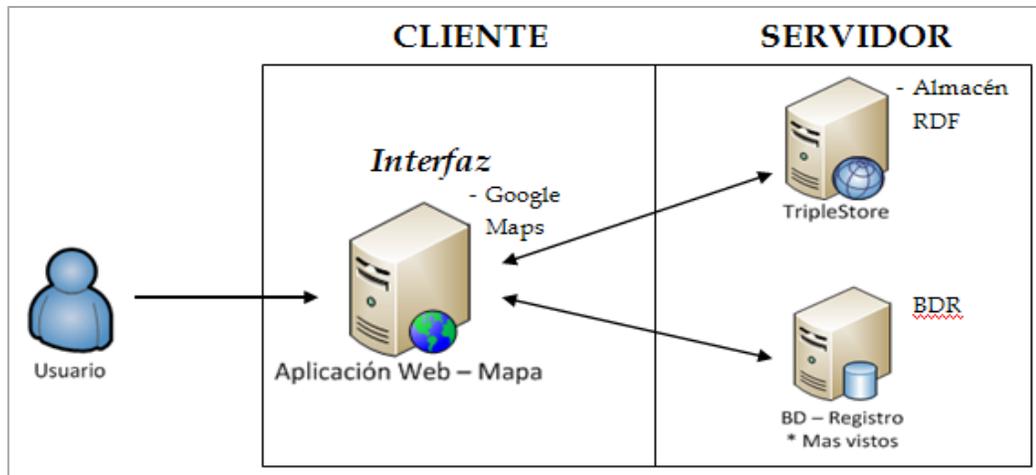


Figura 4.6. Arquitectura cliente-servidor

La siguiente Figura explica con más detalle la arquitectura de la solución.

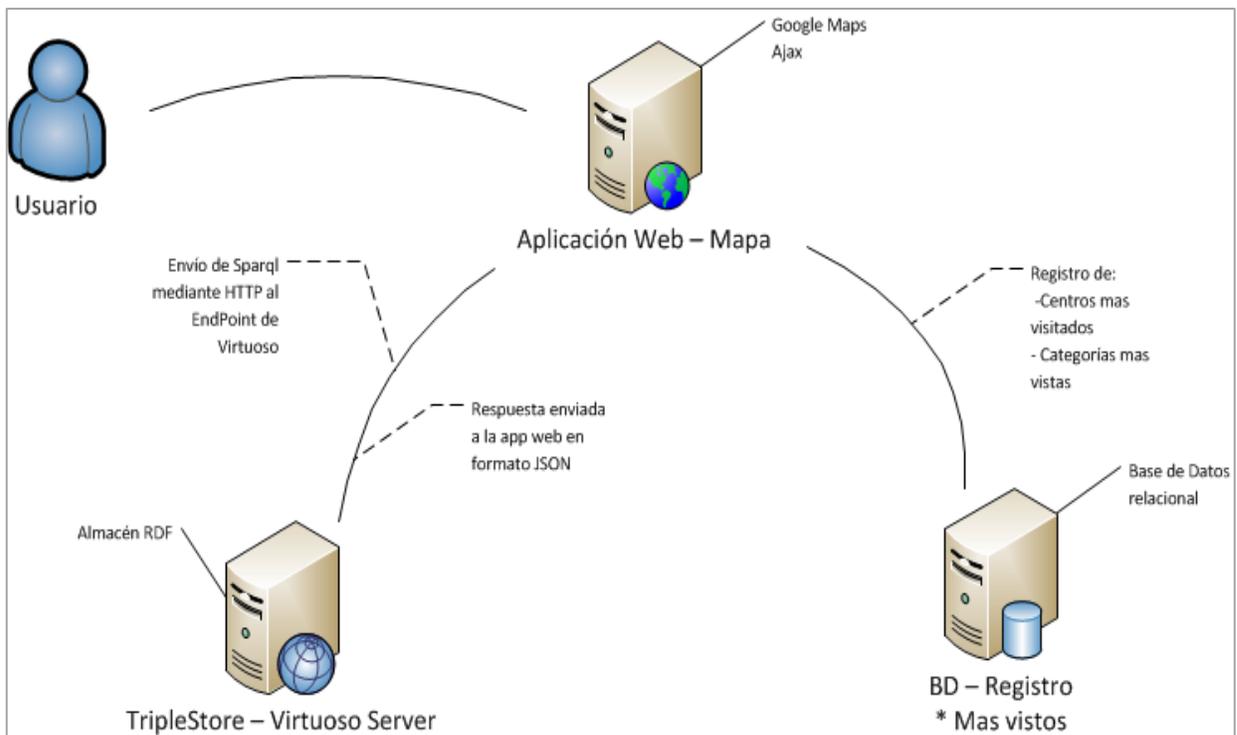


Figura 4.7. Arquitectura de la Solución

4.2.1. COMPONENTES

La arquitectura del proyecto está conformada por los componentes:

- **Aplicación Web**

La aplicación web, contiene la interfaz hacia el usuario final, así como la aplicación misma. Aquí se encuentra la integración con GoogleMaps para la visualización geo referenciada de los centros universitarios de la UTPL.

Toda la información será extraída mediante peticiones HTTP realizadas con tecnología Ajax tanto a la Base Relacional como al EndPoint del Servidor Virtuoso que almacena las estructuras RDF. Aquí también se encuentran los procedimientos encargados de transformar la información extraída para que sea visualizada por el usuario.

La aplicación está construida utilizando las tecnologías php, JavaScript y Ajax para la recuperación de información desde la capa de Datos.

- **Almacén de RDF**

La plataforma, debido a sus fines, utilizará un almacén de tripletas (RDF) sobre el cual se operarán búsquedas según los requerimientos del cliente y necesaria para conseguir el objetivo principal del proyecto.

En este caso se ha implementado haciendo uso de un servidor Virtuoso⁴⁷ en su versión OpenSource, el mismo que presta gran facilidad para su interacción con aplicaciones mediante peticiones HTTP, además contiene un EndPoint por medio del cual se recuperan las tripletas en formato JSON⁴⁸ (*Notación de Objetos de JavaScript*). Se utilizará este formato por la facilidad de interacción existente con las aplicaciones basadas en web.

⁴⁷ **Servidor Universal Virtuoso:** es un middleware y base de datos de motor híbrido que combina la funcionalidad de un RDBMS tradicional, base de datos virtual, RDF, XML, texto libre, servidor de aplicaciones Web y servidor de archivos, en un producto único servidor que ofrece.

De igual forma para su administración presenta una interfaz conocida bajo el nombre de CONDUCTOR que permite la carga de datos RDF a partir de un archivo. Lo que contribuye al objetivo de la plataforma.

- **Bases de Datos (Almacén Relacional)**

El uso de una base de datos relacional, será para almacenar los datos estadísticos obtenidos de las visitas realizadas a través del mapa de conocimiento que se mostrarán en la interfaz principal de la aplicación. Estos comprenden los centros universitarios de la universidad más visitados, lo más buscado y categorías más vistas. Estadísticas que son calculadas a medida que un usuario interactúa con la plataforma.

4.3. TECNOLOGÍAS

Las tecnologías que se utilizan para la implementación de la arquitectura web son las siguientes:

4.3.1. APLICACIÓN WEB

- **GoogleMaps⁴⁹**

Se utiliza el API de GoogleMaps para generar el mapa del Ecuador, sobre el cual se ubicaran diferentes marcadores para la representación de los centros universitarios de la UTPL. Gestionados a través de la combinación de las tecnologías que se mencionarán a continuación.

GoogleMaps es un servicio de *Google* que ofrece imágenes vía satélite de todo el planeta, combinadas, en el caso de algunos países, con mapas de sus ciudades, lo que unido a sus posibilidades de programación abierta ha dado lugar a diversas utilidades ofrecidas desde numerosas páginas web.

⁴⁸ **JSON:** es un formato ligero de intercambio de datos. Está basado en un subconjunto del Lenguaje de Programación JavaScript.

⁴⁹ http://www.googlemaps.es/?page_id=3

GoogleMaps **descansa poderosamente sobre la utilización de JavaScript**. La carga y el deslizamiento de imagen no podrían efectuarse sin este código.

GoogleMaps provee a los desarrolladores un API capaz de aprovechar los datos disponibles a través del servicio, en el seno de las propias aplicaciones.

- **Ajax**⁵⁰ (Asynchronous JavaScript y XML)

El uso de Ajax facilita que la recuperación de datos desde el servidor sea realizado sin necesidad de recargar la página nuevamente, lo que facilitará la consulta de datos de ambos almacenes de datos especializados en la arquitectura.

Ajax es una técnica para la creación de páginas, la importancia de ésta técnica está en que al utilizarla, donde su principal ventaja es el mantener una comunicación asíncrona con el servidor para la recuperación de información.

- **JavaScript**

Mediante la utilización de la librería **OpenLayers**⁵¹, la cual está construida totalmente en JavaScript, se gestionará la representación del mapa, así como las operaciones que se realizan sobre los marcadores.

JavaScript es un lenguaje interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

- **JSON**⁵² (JavaScript Object Notation)

Es un formato ligero de intercambio de datos. Por la gran aceptación de este formato, se lo ha considerado como el indicado para la comunicación entre la capa de datos y la capa de aplicación.

⁵⁰ <http://techlosofy.com/que-es-ajax/>

⁵¹ <http://openlayers.org/>

⁵² <http://JSON.org/JSON-es.HTML>

JSON está basado en un subconjunto del Lenguaje de Programación JavaScript. *JSON* es un formato de texto que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que *JSON* sea un lenguaje ideal para el intercambio de datos.

- **HTML- PHP**

Conforman la estructura principal, tanto para la representación de datos hacia el usuario como para el procesamiento de la información obtenida de la capa de datos. Constituyendo la interfaz principal de la aplicación Web.

HTML es el código que posibilita la creación y edición de documentos web. Es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web.

4.3.2. ALMACÉN DE RDF

- **RDF**⁵³

En este formato se almacenará toda la información extraída de los diferentes centros universitarios de la UTPL, la misma que será almacenada en el servidor Virtuoso⁵⁴.

RDF es un modelo estándar para el intercambio de datos en la Web. *RDF* tiene características que facilitan la fusión de datos, incluso si los esquemas subyacentes son diferentes, y se apoya específicamente la evolución de los esquemas en el tiempo sin necesidad de todos los consumidores de datos a ser cambiado.

⁵³ <http://www.w3.org/RDF/>

⁵⁴ <http://virtuoso.openlinksw.com/>

- **TripleStore**

Sitio de almacenaje de estructuras semánticas, representadas como tripletas, que representan información relativa a un tema en particular. El mismo puede ser accedido mediante lenguajes de consulta semánticos.

- **SPARQL⁵⁵**

Es un lenguaje de consulta semántico, que permite la extracción de tripletas desde el TripleStore para ser enviadas al aplicativo web.

SPARQL es un lenguaje de consulta orientado a la operación sobre LinkedData, que le permite operar sobre grafos o tripletas, propuesto por la W3C, actualmente goza de gran aceptación en la mayoría de TripleStores. En este caso Virtuoso brinda soporte para la versión 1.1 de SPARQL la misma que permite realizar operaciones de actualización y borrado.

4.3.3. BASES DE DATOS

- **MySQL⁵⁶**

Uno de los motores de base de datos relacional con mayor acogida en ambientes transaccionales, debido a su distribución libre y su estructura sencilla. Lo que facilita su integración al proyecto, para el manejo de estadísticas de visita.

4.3.3.1. DISEÑO DE DATOS:

Para el diseño de datos se utilizó el siguiente diagrama relacional; en donde los datos necesarios para mostrar las estadísticas de la parte inferior de la página principal, serán almacenados en base a la estructura mostrada en la Figura 4.8.

⁵⁵ <http://www.w3.org/TR/RDF-SPARQL-query/>

⁵⁶ <http://www.mysql.com>

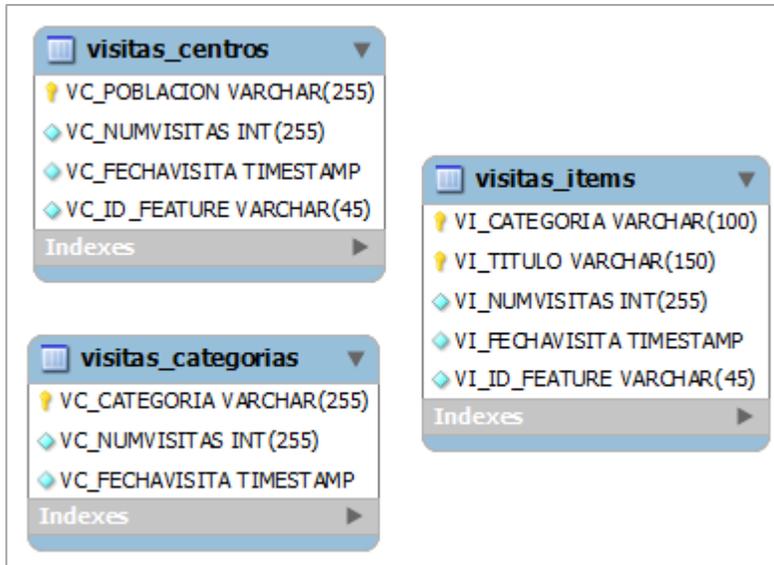


Figura 4.8. Diagrama Relacional de Base de Datos

Los campos NUM_VISITAS, serán incrementados en interacción directa con la aplicación, mediante la detección de un evento ‘clic’. De igual forma la fecha de este evento será registrada.

4.4. LENGUAJES DE CONSULTAS DE RDF

En el capítulo III de la sección 3.2.4 se analizaron los diferentes lenguajes de consultas para RDF como son: RQL, RDQL, SeRQL y SPARQL, en esta sección se hace una comparativa de estos lenguajes según [16] para seleccionar el más idóneo para el desarrollo de la solución.

RQL

RQL viene de las siglas en inglés “RDF Query Language” que significa Lenguaje de Consulta para RDF.

La principal característica de este lenguaje, y que los distinguen de otros lenguajes de consulta de RDF es su capacidad de consultar esquemas RDF y descripciones RDF en la misma consulta.

Además para construir la consulta RQL se deben manejar 3 cláusulas que son: SELECT, FROM y WHERE, donde:

- **SELECT:** Define la proyección sobre las variables de interés.
- **FROM:** Especificamos una parte del subgrafo RDF (un filtro).
- **WHERE:** Una expresión formada por la comparación de variables (o del resultado de otra Query).

Sentencia

```
SELECT vars
FROM Filters
WHERE Expressions
```

Figura 4.9. Estructura básica de una sentencia RQL.

RDQL

RDQL viene de las siglas en inglés “RDF Data Query Language” que significa Datos del Lenguaje de Consulta RDF. La sintaxis RDQL es similar a la de SQL.

Ha sido desarrollado por HP y es soportado por Jena2. De hecho, también podemos decir que RDQL es el lenguaje de consulta para grafos RDF de Jena2. RDQL trata RDF como datos y provee consultas con patrones de sentencias y restricciones sobre un modelo RDF. Se diseñó para ser usado en scripts y para experimentación en lenguajes de modelación de información.

Además para construir la consulta RDQL se deben manejar 5 cláusulas que son: SELECT, SOURCE, WHERE, AND y USING, donde:

- **SELECT:** define las variables que se requieren mostrar en el conjunto resultante.
- **WHERE:** define un patrón de subgrafo en términos de variables y constantes.
- **AND:** introduce un filtro en las variables; solamente los resultados que pasan el filtro son incluidos en el conjunto resultante de la consulta.

- **USING:** permite definir abreviaciones para espacios de nombre (NAMESPACES).

Sentencia

```

SELECT vars
  FROM documents
  WHERE Expressions
  AND Filters
  USING Namespace declarations

```

Figura 4.10. Estructura básica de una sentencia RDQL.

SeRQL

SeRQL viene de las siglas en inglés “Sesame RDF Query Language”, que significa Lenguaje de Consulta para RDF/RDFS el cual fue desarrollado por Aduna como parte del software Sesame.

SeRQL es un lenguaje declarativo de consulta de instancias y esquemas RDF, que aprovecha todas las características del modelamiento RDF.

Además para construir la consulta RDQL se deben manejar 6 cláusulas que son: SELECT, FROM, WHERE, LIMIT, OFFSET y USING NAMESPACE, donde:

- **SELECT:** podemos indicar que valores obtendremos y en qué orden.
- **FROM:** es opcional y determina la ruta de la expresión, es decir, define el camino o ruta de un grafo RDF.
- **WHERE:** es opcional y puede tener operadores lógicos.
- **LIMIT y OFFSET** también son optativas. Pueden ser usadas separada o conjuntamente en orden, para obtener un subconjunto de la consulta.
- **USING NAMESPACE:** es también opcional y puede contener declaraciones de namespaces.

Sentencia

```
SELECT (DISTINCT) campos  
FROM modelo de grafo  
WHERE  
LIMIT n  
OFFSET m  
USING Namespace
```

Figura 4.11. Estructura básica de una sentencia SeRQL

SPARQL

Es un lenguaje de consulta semántico el cual se encuentra basado en RDF. SPARQL viene del acrónimo en inglés de SPARQL Protocol and RDF Query Language, este es una recomendación para crear un lenguaje de consulta más robusto de ayuda para la web semántica.

La recomendación World Wide Web Consortium (W3C) para lenguajes de consulta sobre RDF es SPARQL.

SPARQL se basa en patrones de grafos. Los patrones más simples son las tripletas, los cuales son como las tripletas RDF pero con la posibilidad de tener una variable en el sujeto, predicado u objeto.

Además para construir la consulta RDQL se deben manejar 2 cláusulas que son: SELECT y WHERE, además existen algunos auxiliares de ordenamiento, agrupación y modificación, estos son: Distinct, Describe, Order By Desc, Order By Asc, Limit, Union y Offset.

- **SELECT:** La cláusula SELECT identifica las variables que aparecen en el resultado de la Query.
- **WHERE:** Está formada por tripletas.

Sentencia

La sintaxis básica de una consulta SPARQL [22] se muestra en la Tabla.

Tabla 4.6. Sintaxis básica de una consulta SPARQL.

Prologue (optional)	BASE <iri> PREFIX prefix: <iri> (repeatable)
Query Result forms (required, pick 1)	SELECT (DISTINCT) sequence of ?variable SELECT (DISTINCT)* DESCRIBE sequence of ?variable or <iri> DESCRIBE * CONSTRUCT { graph pattern } ASK
Query Dataset Sources (optional)	Add triples to the background graph (repeatable): FROM <iri> Add a named graph (repeatable): FROMNAMED<iri>
Graph Pattern (optional, required for ASK)	WHERE { graphpattern z}
Query Results Ordering (optional)	ORDER BY ...
Query Results Selection (optional)	LIMIT n, OFFSET m

4.4.1. COMPARATIVA

A continuación se realiza una comparativa según [16] de los lenguajes de consultas para RDF, en donde se detalla los criterios de evaluación:

- **Tipos de Datos.-** Son los diferentes tipos de datos que soporta cada lenguaje.
- **Operadores Matemáticos.-**Entre estos se encuentran las Operaciones Aritméticas, Comparadores, Operadores Lógicos, Funciones de Agrupación y Funciones de Cadena de caracteres.
- **Clases y Objetos.-** Hacen referencia a la recuperación de datos representados por conceptos y sus instancias.

- **Expresiones Condicionales.-** Hacen referencia a estructuras de control y de filtrado de la información a recuperar.
- **Agregación.-**son especialmente necesarias para identificar el mínimo y máximo de un conjunto de valores.
- **Recursión.-** Las consultas recursivas aparecen a menudo en los sistemas de información, típicamente si la relación subyacente es transitiva.
- **Namespaces.-** Los namespaces son una parte integral de cualquier lenguaje de consulta para los datos basados en Web.
- **Literales y tipos de datos.-** Los literales se utilizan para identificar valores tales como números y fechas por medio de una representación léxica.

Tabla 4.7. Comparativa Lenguajes de Consulta para RDF.

<i>Lenguajes</i> <i>Crterios</i>	RQL	RDQL	SeRQL	SPARQL
Tipos de Datos	Strings, Integers, reals, dates, enumerated, thesauri	Strings, integers, float.	Strings, float, integer	Boolean, String, double, RI, float, decimal, Integer, dateTime, literal,
Operadores Matemáticos	SI	SI	SI	SI
Clases y Objetos	Restringido	Restringido	Restringido	Restringido
Expresiones Condicionales	SI	SI	SI	SI
Agregación	SI	NO	NO	SI
Recursión	NO	NO	NO	SI
Namespaces	SI	Restringido	SI	SI
Lexical Space	SI	SI	SI	SI
ValueSpace	SI	Restringido	SI	SI

Observando la tabla comparativa se concluye que el lenguaje SPARQL es el más idóneo para realizar las consultas en el desarrollo del mapa de conocimiento, cuyo lenguaje es el más utilizado hoy en día y es recomendado por W3C.

4.4.2. PRUEBAS

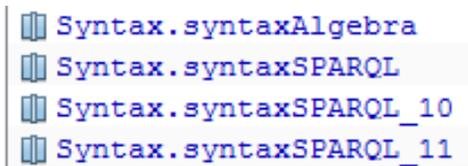
Como se ha mencionado en apartados anteriores, existen varios lenguajes de consulta que se pueden aplicar a Stores RDF para la recuperación de datos. De esta forma se realizarán pruebas como se detalla a continuación:

RDQL y RQL: Estos dos lenguajes tienen su origen por el año 2004, antes de la aparición de SPARQL, lo que ha provocado que hayan sido relegados y no exista soporte actual. En primeras instancias Jena⁵⁷ brindaba soporte en su versión 2.1 para el trabajo con RDQL (versión liberada por el año 2004) pero actualmente se encuentra en su versión 2.9, en la cual ya no se incluye esta característica. Inclusive la definición del lenguaje localizada en la siguiente dirección:

<http://jena.hpl.hp.com/2003/07/query/RDQL>

Ya no se encuentra disponible. Se ha examinado el API de Jena en su versión actual en busca de poder realizar consultas de este tipo, pero solamente el API gestiona el trabajo con SPARQL como se puede apreciar en la Figura 4.12. Este cambio se puede observar en el 'changelog' que se puede encontrar en [23]:

```
++ Remove RDQL from ARQ (code in Archive/ in SVN -- will not be updated).
```



```
Syntax.syntaxAlgebra
Syntax.syntaxSPARQL
Syntax.syntaxSPARQL_10
Syntax.syntaxSPARQL_11
```

Figura 4.12. Sintaxis permitidas en Jena

⁵⁷ <http://incubator.apache.org/jena/>

Por este motivo, estos dos lenguajes no serán utilizados para realizar la comparación, ni tampoco serán involucrados dentro de la construcción de la plataforma.

SeRQL: Las pruebas serán realizadas utilizando la plataforma Sesame⁵⁸ (detalle de la Instalación **Anexo A**), haciendo uso de una interfaz propia para el trabajo con consultas SeRQL. Esta plataforma posee requerimientos mínimos como:

- Java 5 o superior
- Un contenedor de Servlets Java que soporte Java Servlet API 2.4 and Java Server Pages (JSP) 2.0, o superior.

Esto debido a que Sesame se distribuye como dos archivos .war que deben ser desplegados en el contenedor de servlets para luego ser accedidos desde una interfaz web.

SPARQL: Se utilizará la versión Open Source de la plataforma Virtuoso (detalle de instalación **Anexo B**). La misma no posee requerimientos adicionales para su instalación ya que se utilizarán paquetes ya compilados (binarios) para la instalación. Solamente en el caso de compilar una propia versión sería necesario preparar un ambiente especial.

Se debe mencionar además que estos dos entornos serán probados sobre la plataforma Linux en su distribución Ubuntu en la versión 11.04 de 64 bits.

4.4.2.1. DATOS Y ESCENARIOS DE CONSULTAS

Para verificar la funcionalidad de estos dos lenguajes, se utilizará un documento RDF de autoría propia, el mismo que representará elementos bibliográficos como Libros y Tesis, en donde se encontrarán 5 individuos específicos de cada clase.

⁵⁸ <http://www.openRDF.org/>

El archivo ha sido creado utilizando la herramienta Protégé⁵⁹, las clases involucradas se las puede observar en la Figura 4.13.

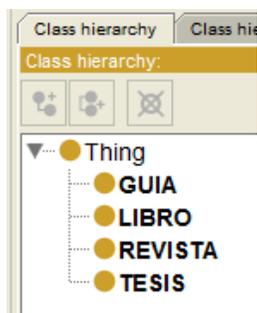


Figura 4.13. Clases de archivo RDF.

Las propiedades relacionadas con cada clase y que definirán a cada individuo que se creará, se muestran a continuación, en donde la columna *Propiedad (RDF)* determina como se ha representado el *Atributo* en el documento RDF por medio de Protégé.

Tabla 4.8. Atributos de Clase LIBRO.

LIBRO	
Atributo	Propiedad (RDF)
Autor	autor
Título	titulo
Sub-Título	subtitulo
Año	anio
Editorial	editorial
País	pais
Edición	Edición
Idioma	Idioma
ISBN	isbn
Sitio Web	Web
Área	area
Recurso Adicional	recAdc
Fecha de Ingreso	fechaIn

⁵⁹<http://protege.stanford.edu/>

Palabra Clave	keyWord
Resumen	resumen

Tabla 4.9. Atributos de Clase TESIS.

TESIS	
Atributo	Propiedad (RDF)
Título	titulo
Sub-Título	subtitulo
Autor	autor
Modalidad	modalidad
Nivel	nivel
Carrera	carrera
Director	director
Co-Director	codirector
Año	anio
Ciudad	ciudad
País	pais
Área	area
Recurso Adicional	recAdc
Fecha de Ingreso	fechaIn
Palabra Clave	keyWord
Resumen	resumen

En este caso la propiedad *nivel*, tomará valores como *licenciatura, ingeniería, postgrado o doctorado*.

Tabla 4.10. Atributos de Clase REVISTA.

REVISTA	
Atributo	Propiedad (RDF)
Nombre	nombre
Editorial	editorial
ISSN	issn
Año	anio

Número	num
Periodo De Publicación	perPublic
Fecha de Publicación	fechaPub
Recurso Adicional	recAdc
Fecha de Ingreso	fechaIn
Palabra Clave	keyWord
Resumen	resumen
Sitio Web	web

El valor de la propiedad *perPublic* hace referencia a periodos como *quincenal*, *mensual*, *trimestral*, etc.

Tabla 4.11. Atributos de Clase GUÍA.

GUÍA	
Atributo	Propiedad (RDF)
Nombre	nombre
Editorial	editorial
Institución	institución
Nivel	nivel
Carrera	carrera
Ciclo	ciclo
Año	anio
País	pais
Edición	edicion
Idioma	idioma

La representación de estas propiedades, mediante Protégé se puede observar en la Figura 4.14.

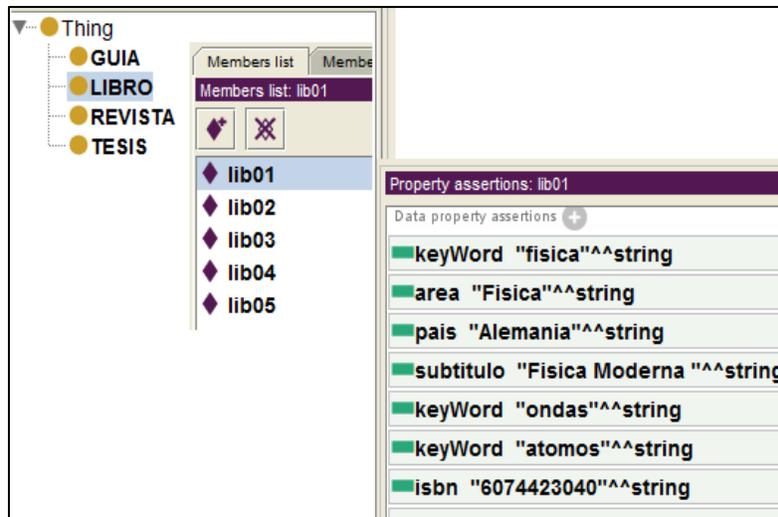


Figura 4.14. Clases y Propiedades del archivo RDF.

Toda la creación de individuos, cada uno con sus respectivas propiedades y valores, es almacenada en un archivo con extensión .owl, el mismo que contiene sintaxis RDF como se muestra en la Figura 4.15. Además, se ha definido en este archivo una dirección URI para la localización de los individuos, que será utilizada posteriormente al realizar la recuperación de los datos.

En este caso la dirección es:

<http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#>

```

<owl:NamedIndividual rdf:about="&dataTest;guia05">
  <rdf:type rdf:resource="&dataTest;GUIA"/>
  <nombre rdf:datatype="&xsd:string">Contabilidad</nombre>
  <editorial rdf:datatype="&xsd:string">UTPL</editorial>
  <institucion rdf:datatype="&xsd:string">UTPL</institucion>
  <nivel rdf:datatype="&xsd:string">Licenciatura</nivel>
  <carrera rdf:datatype="&xsd:string">Banca y Finanzas</carrera>
  <carrera rdf:datatype="&xsd:string">Químico Biológicas</carrera>

```

Figura 4.15. Contenido del archivo RDF.

En base a este archivo y su información (individuos específicos de cada clase), se han formulado un conjunto de 10 preguntas que serán respondidas haciendo uso de los lenguajes de consulta.

Consulta 01: *Todos los conceptos.*

Se extraerán todas las clases que actualmente se encuentran definidas y utilizadas dentro del Store.

Consulta 02: *Todos los libros existentes.*

Visualiza el atributo 'título' de todos los individuos cuyo tipo sea de la clase 'LIBRO' y que se encuentren en el Store.

Consulta 03: *Todos los libros del año 2009.*

Extrae y presenta el atributo 'título' de todos los individuos de clase 'LIBRO' que posean el atributo 'anio' con un valor igual a '2009'.

Consulta 04: *Todas las palabras claves de los libros en inglés.*

Mostrar el valor del atributo 'keyWord' de los individuos de tipo 'LIBRO' que posean el atributo 'idioma' igual a 'inglés'.

Consulta 05: *Titulo de tesis en modalidad presencial.*

Extrae el valor del atributo 'titulo' de todos los individuos del tipo 'TESIS', donde su atributo 'modalidad' sea igual al valor 'PRESENCIAL'.

Consulta 06: *Todas las tesis de doctorado.*

Extrae el valor del atributo 'titulo' de todos los individuos del tipo 'TESIS', donde el atributo 'nivel' es igual al valor 'DOCTORADO.'

Consulta 07: *El máximo número en revistas.*

Extrae el valor máximo del atributo 'numero' de entre todos los individuos del tipo 'REVISTA'.

Consulta 08: *Todas las palabras claves de las revistas que se publican mensualmente.*

Extrae el valor del atributo 'nombre' y 'keyWord' de todos los individuos que sean del tipo 'REVISTA' y además su atributo 'perPublic' sea igual al valor 'Mensual'.

Consulta 09: *Todas las guías publicadas por la UTPL.*

Extrae el valor del atributo 'nombre' de todos los individuos del tipo 'GUÍA' donde el atributo 'institución' coincide con el valor 'UTPL'.

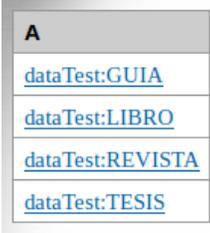
Consulta 10: *Cuántas guías pertenecen a 3^{er} ciclo.*

Contabiliza el número de individuos del tipo 'GUÍA', donde su atributo 'ciclo' es igual al valor '3'.

4.4.2.2. COMPARATIVA CONSULTAS SERQL Y SPARQL

A continuación se presenta una tabla comparativa de las distintas salidas de las 10 consultas realizadas:

Tabla 4.12. Comparativa consultas SeRQL y SPARQL

Consulta	SeRQL	SPARQL										
C 01	<p>Código:</p> <pre>SELECT A FROM {A} RDF:type {owl:Class} USING NAMESPACE lib = <http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#></pre> <p>Resultado:</p> 	<p>Código:</p> <pre>SELECT DISTINCT ?CONCEPT WHERE {[] A ?CONCEPT}</pre> <p>Resultado:</p> <table border="1"> <thead> <tr> <th>Concept</th> </tr> </thead> <tbody> <tr> <td>http://www.w3.org/2002/07/owl#Class</td> </tr> <tr> <td>http://www.w3.org/2002/07/owl#Ontology</td> </tr> <tr> <td>http://www.w3.org/2002/07/owl#DatatypeProperty</td> </tr> <tr> <td>http://www.w3.org/2000/01/rdf-schema#Datatype</td> </tr> <tr> <td>http://www.dirvocabulary.org/ontology/material/bibliografico/dataTest.owl#GUIA</td> </tr> <tr> <td>http://www.dirvocabulary.org/ontology/material/bibliografico/dataTest.owl#LIBRO</td> </tr> <tr> <td>http://www.dirvocabulary.org/ontology/material/bibliografico/dataTest.owl#REVISTA</td> </tr> <tr> <td>http://www.dirvocabulary.org/ontology/material/bibliografico/dataTest.owl#TESIS</td> </tr> <tr> <td>http://www.w3.org/2002/07/owl#NamedIndividual</td> </tr> </tbody> </table>	Concept	http://www.w3.org/2002/07/owl#Class	http://www.w3.org/2002/07/owl#Ontology	http://www.w3.org/2002/07/owl#DatatypeProperty	http://www.w3.org/2000/01/rdf-schema#Datatype	http://www.dirvocabulary.org/ontology/material/bibliografico/dataTest.owl#GUIA	http://www.dirvocabulary.org/ontology/material/bibliografico/dataTest.owl#LIBRO	http://www.dirvocabulary.org/ontology/material/bibliografico/dataTest.owl#REVISTA	http://www.dirvocabulary.org/ontology/material/bibliografico/dataTest.owl#TESIS	http://www.w3.org/2002/07/owl#NamedIndividual
Concept												
http://www.w3.org/2002/07/owl#Class												
http://www.w3.org/2002/07/owl#Ontology												
http://www.w3.org/2002/07/owl#DatatypeProperty												
http://www.w3.org/2000/01/rdf-schema#Datatype												
http://www.dirvocabulary.org/ontology/material/bibliografico/dataTest.owl#GUIA												
http://www.dirvocabulary.org/ontology/material/bibliografico/dataTest.owl#LIBRO												
http://www.dirvocabulary.org/ontology/material/bibliografico/dataTest.owl#REVISTA												
http://www.dirvocabulary.org/ontology/material/bibliografico/dataTest.owl#TESIS												
http://www.w3.org/2002/07/owl#NamedIndividual												
C02	<p>Código:</p> <pre>SELECT B FROM {A} RDF:type {lib:LIBRO}; lib:titulo {B} USING NAMESPACE lib = <http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#></pre>	<p>Código:</p> <pre>PREFIX :<http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#> SELECT ?titulo WHERE { ?lib a :LIBRO; :titulo ?titulo}</pre>										

	<p>Resultado:</p> <table border="1"> <thead> <tr> <th>B</th> </tr> </thead> <tbody> <tr> <td>"Fisica Universitaria"^^xsd:string</td> </tr> <tr> <td>"Leadership: Theory and Practice"^^xsd:string</td> </tr> <tr> <td>"RESUMEN CRONOLOGICO DE LA HISTORIA UNIVERSAL"^^xsd:string</td> </tr> <tr> <td>"Selección de Metodologías de Desarrollo para Aplicaciones Web"^^xsd:string</td> </tr> <tr> <td>"Publication Manual of the American Psychological"^^xsd:string</td> </tr> </tbody> </table>	B	"Fisica Universitaria"^^xsd:string	"Leadership: Theory and Practice"^^xsd:string	"RESUMEN CRONOLOGICO DE LA HISTORIA UNIVERSAL"^^xsd:string	"Selección de Metodologías de Desarrollo para Aplicaciones Web"^^xsd:string	"Publication Manual of the American Psychological"^^xsd:string	<p>Resultado:</p> <table border="1"> <thead> <tr> <th>titulo</th> </tr> </thead> <tbody> <tr> <td>Fisica Universitaria</td> </tr> <tr> <td>Leadership: Theory and Practice</td> </tr> <tr> <td>RESUMEN CRONOLOGICO DE LA HISTORIA UNIVERSAL</td> </tr> <tr> <td>Selección de Metodologías de Desarrollo para Aplicaciones Web</td> </tr> <tr> <td>Publication Manual of the American Psychological</td> </tr> </tbody> </table>	titulo	Fisica Universitaria	Leadership: Theory and Practice	RESUMEN CRONOLOGICO DE LA HISTORIA UNIVERSAL	Selección de Metodologías de Desarrollo para Aplicaciones Web	Publication Manual of the American Psychological
B														
"Fisica Universitaria"^^xsd:string														
"Leadership: Theory and Practice"^^xsd:string														
"RESUMEN CRONOLOGICO DE LA HISTORIA UNIVERSAL"^^xsd:string														
"Selección de Metodologías de Desarrollo para Aplicaciones Web"^^xsd:string														
"Publication Manual of the American Psychological"^^xsd:string														
titulo														
Fisica Universitaria														
Leadership: Theory and Practice														
RESUMEN CRONOLOGICO DE LA HISTORIA UNIVERSAL														
Selección de Metodologías de Desarrollo para Aplicaciones Web														
Publication Manual of the American Psychological														
C03	<p>Código:</p> <pre> SELECT B FROM {A} RDF:type {lib:LIBRO}; lib:titulo {B}; lib:anio {C} WHERE C = 2009 USING NAMESPACE lib = <http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#> </pre> <p>Resultado:</p> <table border="1"> <thead> <tr> <th>B</th> </tr> </thead> <tbody> <tr> <td>"Fisica Universitaria"^^xsd:string</td> </tr> <tr> <td>"RESUMEN CRONOLOGICO DE LA HISTORIA UNIVERSAL"^^xsd:string</td> </tr> <tr> <td>"Publication Manual of the American Psychological"^^xsd:string</td> </tr> </tbody> </table>	B	"Fisica Universitaria"^^xsd:string	"RESUMEN CRONOLOGICO DE LA HISTORIA UNIVERSAL"^^xsd:string	"Publication Manual of the American Psychological"^^xsd:string	<p>Código:</p> <pre> PREFIX :<http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#> SELECT ?titulo WHERE { ?lib a :LIBRO; :titulo ?titulo; :anio ?anio. FILTER(?anio = 2009)} </pre> <p>Resultado:</p> <table border="1"> <thead> <tr> <th>titulo</th> </tr> </thead> <tbody> <tr> <td>Fisica Universitaria</td> </tr> <tr> <td>RESUMEN CRONOLOGICO DE LA HISTORIA UNIVERSAL</td> </tr> <tr> <td>Publication Manual of the American Psychological</td> </tr> </tbody> </table>	titulo	Fisica Universitaria	RESUMEN CRONOLOGICO DE LA HISTORIA UNIVERSAL	Publication Manual of the American Psychological				
B														
"Fisica Universitaria"^^xsd:string														
"RESUMEN CRONOLOGICO DE LA HISTORIA UNIVERSAL"^^xsd:string														
"Publication Manual of the American Psychological"^^xsd:string														
titulo														
Fisica Universitaria														
RESUMEN CRONOLOGICO DE LA HISTORIA UNIVERSAL														
Publication Manual of the American Psychological														

C04	<p>Código:</p> <pre> SELECT B, D FROM {A} RDF:type {lib:LIBRO}; lib:titulo {B}; lib:idioma {C}; lib:keyWord {D} WHERE C = "Ingles" USING NAMESPACE lib = <http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#> </pre> <p>Resultado:</p> <table border="1"> <thead> <tr> <th>B</th> <th>D</th> </tr> </thead> <tbody> <tr> <td>"Leadership: Theory and Practice"^^xsd:string</td> <td>"Autoestima"^^xsd:string</td> </tr> <tr> <td>"Leadership: Theory and Practice"^^xsd:string</td> <td>"Liderazgo"^^xsd:string</td> </tr> <tr> <td>"Leadership: Theory and Practice"^^xsd:string</td> <td>"Reglas"^^xsd:string</td> </tr> <tr> <td>"Publication Manual of the American Psychological"^^xsd:string</td> <td>"Enfermedad"^^xsd:string</td> </tr> <tr> <td>"Publication Manual of the American Psychological"^^xsd:string</td> <td>"Manual"^^xsd:string</td> </tr> <tr> <td>"Publication Manual of the American Psychological"^^xsd:string</td> <td>"Medicina"^^xsd:string</td> </tr> </tbody> </table>	B	D	"Leadership: Theory and Practice"^^xsd:string	"Autoestima"^^xsd:string	"Leadership: Theory and Practice"^^xsd:string	"Liderazgo"^^xsd:string	"Leadership: Theory and Practice"^^xsd:string	"Reglas"^^xsd:string	"Publication Manual of the American Psychological"^^xsd:string	"Enfermedad"^^xsd:string	"Publication Manual of the American Psychological"^^xsd:string	"Manual"^^xsd:string	"Publication Manual of the American Psychological"^^xsd:string	"Medicina"^^xsd:string	<p>Código:</p> <pre> PREFIX :<http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#> SELECT ?titulo, ?palClave WHERE { ?lib a :LIBRO; :titulo ?titulo; :keyWord ?palClave; :idioma ?idioma. FILTER(regex(str(?idioma),"Ingles"))} </pre> <p>Resultado:</p> <table border="1"> <thead> <tr> <th>titulo</th> <th>palClave</th> </tr> </thead> <tbody> <tr> <td>Leadership: Theory and Practice</td> <td>Autoestima</td> </tr> <tr> <td>Leadership: Theory and Practice</td> <td>Liderazgo</td> </tr> <tr> <td>Leadership: Theory and Practice</td> <td>Reglas</td> </tr> <tr> <td>Publication Manual of the American Psychological</td> <td>Enfermedad</td> </tr> <tr> <td>Publication Manual of the American Psychological</td> <td>Manual</td> </tr> <tr> <td>Publication Manual of the American Psychological</td> <td>Medicina</td> </tr> </tbody> </table>	titulo	palClave	Leadership: Theory and Practice	Autoestima	Leadership: Theory and Practice	Liderazgo	Leadership: Theory and Practice	Reglas	Publication Manual of the American Psychological	Enfermedad	Publication Manual of the American Psychological	Manual	Publication Manual of the American Psychological	Medicina
B	D																													
"Leadership: Theory and Practice"^^xsd:string	"Autoestima"^^xsd:string																													
"Leadership: Theory and Practice"^^xsd:string	"Liderazgo"^^xsd:string																													
"Leadership: Theory and Practice"^^xsd:string	"Reglas"^^xsd:string																													
"Publication Manual of the American Psychological"^^xsd:string	"Enfermedad"^^xsd:string																													
"Publication Manual of the American Psychological"^^xsd:string	"Manual"^^xsd:string																													
"Publication Manual of the American Psychological"^^xsd:string	"Medicina"^^xsd:string																													
titulo	palClave																													
Leadership: Theory and Practice	Autoestima																													
Leadership: Theory and Practice	Liderazgo																													
Leadership: Theory and Practice	Reglas																													
Publication Manual of the American Psychological	Enfermedad																													
Publication Manual of the American Psychological	Manual																													
Publication Manual of the American Psychological	Medicina																													
C05	<p>Código:</p> <pre> SELECT B FROM {A} RDF:type {lib:TESIS}; lib:titulo {B}; lib:modalidad {C} WHERE C = "Presencial" </pre>	<p>Código:</p> <pre> PREFIX :<http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#> SELECT ?titulo WHERE { ?tesis a :TESIS; :titulo ?titulo; </pre>																												

	<p><i>USING NAMESPACE</i> <i>lib =</i> <i><http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#></i></p> <p>Resultado:</p> <table border="1"> <thead> <tr> <th>B</th> </tr> </thead> <tbody> <tr> <td>"Reconstrucción de Analisis vectorial de sonido"^^xsd:string</td> </tr> <tr> <td>"Construcción de Pavimentos rígidos basados en lineamientos ambientales"^^xsd:string</td> </tr> <tr> <td>"Levantamiento de especies de animales en peligro dentro del parque nacional podocarpus"^^xsd:string</td> </tr> </tbody> </table>	B	"Reconstrucción de Analisis vectorial de sonido"^^xsd:string	"Construcción de Pavimentos rígidos basados en lineamientos ambientales"^^xsd:string	"Levantamiento de especies de animales en peligro dentro del parque nacional podocarpus"^^xsd:string	<p><i>:modalidad ?modalidad.</i> <i>FILTER(regex(str(?modalidad),"Presencial"))}</i></p> <p>Resultado:</p> <table border="1"> <thead> <tr> <th>titulo</th> </tr> </thead> <tbody> <tr> <td>Reconstrucción de Analisis vectorial de sonido</td> </tr> <tr> <td>Construcción de Pavimentos rígidos basados en lineamientos ambientales</td> </tr> <tr> <td>Levantamiento de especies de animales en peligro dentro del parque nacional podocarpus</td> </tr> </tbody> </table>	titulo	Reconstrucción de Analisis vectorial de sonido	Construcción de Pavimentos rígidos basados en lineamientos ambientales	Levantamiento de especies de animales en peligro dentro del parque nacional podocarpus
B										
"Reconstrucción de Analisis vectorial de sonido"^^xsd:string										
"Construcción de Pavimentos rígidos basados en lineamientos ambientales"^^xsd:string										
"Levantamiento de especies de animales en peligro dentro del parque nacional podocarpus"^^xsd:string										
titulo										
Reconstrucción de Analisis vectorial de sonido										
Construcción de Pavimentos rígidos basados en lineamientos ambientales										
Levantamiento de especies de animales en peligro dentro del parque nacional podocarpus										
C06	<p>Código:</p> <p><i>SELECT B</i> <i>FROM</i> <i>{A} RDF:type {lib:TESIS};</i> <i>lib:titulo {B};</i> <i>lib:nivel {C}</i> <i>WHERE C = "Doctorado"</i> <i>USING NAMESPACE</i> <i>lib =</i> <i><http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#></i></p> <p>Resultado:</p> <table border="1"> <thead> <tr> <th>B</th> </tr> </thead> <tbody> <tr> <td>"Construcción de Pavimentos rígidos basados en lineamientos ambientales"^^xsd:string</td> </tr> </tbody> </table>	B	"Construcción de Pavimentos rígidos basados en lineamientos ambientales"^^xsd:string	<p>Código:</p> <p><i>PREFIX</i> <i><http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#></i></p> <p><i>SELECT ?titulo</i> <i>WHERE {</i> <i> ?tesis a :TESIS;</i> <i> :titulo ?titulo;</i> <i> :nivel ?nivel.</i> <i>FILTER(regex(str(?nivel),"Doctorado"))}</i></p> <p>Resultado:</p> <table border="1"> <thead> <tr> <th>titulo</th> </tr> </thead> <tbody> <tr> <td>Construcción de Pavimentos rígidos basados en lineamientos ambientales</td> </tr> </tbody> </table>	titulo	Construcción de Pavimentos rígidos basados en lineamientos ambientales				
B										
"Construcción de Pavimentos rígidos basados en lineamientos ambientales"^^xsd:string										
titulo										
Construcción de Pavimentos rígidos basados en lineamientos ambientales										

<p>C07</p>	<p>En SeRQL no posee funciones de agregación, ni de grupo como MIN, MAX, COUNT o AVG.</p>	<p>Código:</p> <pre> PREFIX :<http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#> SELECT MAX(?num) WHERE { ?tesis a :REVISTA; :nombre ?nombre; :num ?num} </pre> <p>Resultado:</p> <table border="1" data-bbox="1182 690 1327 781"> <tr> <td>callret-0</td> </tr> <tr> <td>46</td> </tr> </table>	callret-0	46
callret-0				
46				
<p>C08</p>	<p>Código:</p> <pre> SELECT B, C FROM {A} RDF:type {lib:REVISTA}; lib:nombre {B}; lib:keyWord {C}; lib:perPublic {D} WHERE D = "Mensual" USING NAMESPACE lib = <http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#> </pre>	<p>Código:</p> <pre> PREFIX :<http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#> SELECT ?nombre, ?palClave WHERE { ?tesis a :REVISTA; :nombre ?nombre; :keyWord ?palClave; :perPublic ?perid. FILTER(regex(str(?perid), "Mensual"))} </pre>		

	<p>Resultado:</p> <table border="1"> <thead> <tr> <th>B</th> <th>C</th> </tr> </thead> <tbody> <tr> <td>"National Geographic"^^xsd:string</td> <td>"Animales"^^xsd:string</td> </tr> <tr> <td>"National Geographic"^^xsd:string</td> <td>"Mundo"^^xsd:string</td> </tr> <tr> <td>"National Geographic"^^xsd:string</td> <td>"Naturalez"^^xsd:string</td> </tr> <tr> <td>"Vistazo"^^xsd:string</td> <td>"Deportes"^^xsd:string</td> </tr> <tr> <td>"Vistazo"^^xsd:string</td> <td>"Farandula"^^xsd:string</td> </tr> <tr> <td>"Vistazo"^^xsd:string</td> <td>"Politica"^^xsd:string</td> </tr> <tr> <td>"Oracle"^^xsd:string</td> <td>"Hardware"^^xsd:string</td> </tr> <tr> <td>"Oracle"^^xsd:string</td> <td>"Mercado"^^xsd:string</td> </tr> <tr> <td>"Oracle"^^xsd:string</td> <td>"Software"^^xsd:string</td> </tr> </tbody> </table>	B	C	"National Geographic"^^xsd:string	"Animales"^^xsd:string	"National Geographic"^^xsd:string	"Mundo"^^xsd:string	"National Geographic"^^xsd:string	"Naturalez"^^xsd:string	"Vistazo"^^xsd:string	"Deportes"^^xsd:string	"Vistazo"^^xsd:string	"Farandula"^^xsd:string	"Vistazo"^^xsd:string	"Politica"^^xsd:string	"Oracle"^^xsd:string	"Hardware"^^xsd:string	"Oracle"^^xsd:string	"Mercado"^^xsd:string	"Oracle"^^xsd:string	"Software"^^xsd:string	<p>Resultado:</p> <table border="1"> <thead> <tr> <th>nombre</th> <th>palClave</th> </tr> </thead> <tbody> <tr> <td>National Geographic</td> <td>Animales</td> </tr> <tr> <td>National Geographic</td> <td>Mundo</td> </tr> <tr> <td>National Geographic</td> <td>Naturalez</td> </tr> <tr> <td>Vistazo</td> <td>Deportes</td> </tr> <tr> <td>Vistazo</td> <td>Farandula</td> </tr> <tr> <td>Vistazo</td> <td>Politica</td> </tr> <tr> <td>Oracle</td> <td>Software</td> </tr> <tr> <td>Oracle</td> <td>Hardware</td> </tr> <tr> <td>Oracle</td> <td>Mercado</td> </tr> </tbody> </table>	nombre	palClave	National Geographic	Animales	National Geographic	Mundo	National Geographic	Naturalez	Vistazo	Deportes	Vistazo	Farandula	Vistazo	Politica	Oracle	Software	Oracle	Hardware	Oracle	Mercado
B	C																																									
"National Geographic"^^xsd:string	"Animales"^^xsd:string																																									
"National Geographic"^^xsd:string	"Mundo"^^xsd:string																																									
"National Geographic"^^xsd:string	"Naturalez"^^xsd:string																																									
"Vistazo"^^xsd:string	"Deportes"^^xsd:string																																									
"Vistazo"^^xsd:string	"Farandula"^^xsd:string																																									
"Vistazo"^^xsd:string	"Politica"^^xsd:string																																									
"Oracle"^^xsd:string	"Hardware"^^xsd:string																																									
"Oracle"^^xsd:string	"Mercado"^^xsd:string																																									
"Oracle"^^xsd:string	"Software"^^xsd:string																																									
nombre	palClave																																									
National Geographic	Animales																																									
National Geographic	Mundo																																									
National Geographic	Naturalez																																									
Vistazo	Deportes																																									
Vistazo	Farandula																																									
Vistazo	Politica																																									
Oracle	Software																																									
Oracle	Hardware																																									
Oracle	Mercado																																									
<p>C09</p>	<p>Código:</p> <pre> SELECT B FROM {A} RDF:type {lib:GUIA}; lib:nombre {B}; lib:institucion {C} WHERE C = "UTPL" USING NAMESPACE lib = <http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#> </pre>	<p>Código:</p> <pre> PREFIX :<http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#> SELECT ?nombre WHERE { ?tesis a :GUIA; :nombre ?nombre; :institucion ?inst. FILTER(regex(str(?inst),"UTPL"))} </pre>																																								

	<p>Resultado:</p> <table border="1" data-bbox="483 284 814 500"> <tr> <td>B</td> </tr> <tr> <td>"Antropología"^^xsd:string</td> </tr> <tr> <td>"Cristologia"^^xsd:string</td> </tr> <tr> <td>"Contabilidad"^^xsd:string</td> </tr> </table>	B	"Antropología"^^xsd:string	"Cristologia"^^xsd:string	"Contabilidad"^^xsd:string	<p>Resultado:</p> <table border="1" data-bbox="1186 284 1381 459"> <tr> <td>nombre</td> </tr> <tr> <td>Antropología</td> </tr> <tr> <td>Cristologia</td> </tr> <tr> <td>Contabilidad</td> </tr> </table>	nombre	Antropología	Cristologia	Contabilidad
B										
"Antropología"^^xsd:string										
"Cristologia"^^xsd:string										
"Contabilidad"^^xsd:string										
nombre										
Antropología										
Cristologia										
Contabilidad										
<p>C10</p>	<p>SeRQL no posee funciones de agregación, ni de grupo como MIN, MAX, COUNT o AVG.</p>	<p>Código:</p> <pre> PREFIX :<http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl#> SELECT COUNT(?nombre) WHERE { ?tesis a :GUIA; :nombre ?nombre; :ciclo ?ciclo. FILTER(?ciclo = 3)} </pre> <p>Resultado:</p> <table border="1" data-bbox="1186 1027 1344 1128"> <tr> <td>callret-0</td> </tr> <tr> <td>3</td> </tr> </table>	callret-0	3						
callret-0										
3										

4.4.3. SELECCIÓN

En base a las consultas realizadas y el análisis de los lenguajes en el apartado anterior se puede llegar a la siguiente tabla comparativa.

Tabla 4.13. Resultados de SeRQL & SPARQL

	SeRQL	SPARQL
Sencillo	si	si
Funciones de Agregación	no	si
Actualización	no	si (SPARQL 1.1)
Abreviaturas	si (namespaces)	si (prefix)
Soporte	no	si
Estándar	no	propuesto por la w3c
Stores que lo soportan	Sesame	todos los conocidos

Como se puede apreciar en los resultados, el trabajo con SeRQL, aunque resulte fácil y entendible, resulta ser una solución poco escalable a medida que se desee integrar el Store con otros, o permitir el consumo de aplicaciones externas. Puesto que el avance de SPARQL lo ha llevado en estos días ha convertirse en prácticamente un estándar para la consulta y modificación de datos enlazados.

Otro punto destacable para la elección de SPARQL es la existencia de funciones de agregación como MAX, MIN y otras que facilitan llevar a cabo operaciones comunes. De igual forma la actualización de datos en el Store es realizada de forma directa haciendo uso de SPARQL Update que se encuentra en la versión 1.1 de SPARQL.

En base a estos criterios, se ha considerado el trabajo con SPARQL como lenguaje de consulta para la plataforma.

4.5. DISEÑO DE INTERFACES

La plataforma, para permitir la interacción con el usuario presenta una interfaz principal donde se podrá acceder a los datos disponibles en el TripleStore. Esta interfaz puede ser observada en la Figura 4.16. Todos los detalles que se comentan a continuación así como las figuras mostradas corresponden al diseño previo, con lo cual, en un apartado futuro se implementarán en la interfaz.

En la parte superior estará colocada la imagen representativa de la plataforma, junto al nombre de la misma. En la parte central se localiza un mapa sobre el cual se ubicarán cada uno de los centros universitarios de la UTPL, los mismos que serán representados con un icono en particular.

En la parte inferior se colocan tres listados que representan a los centros con mayor cantidad de visitas, las categorías más vistas de todos los centros y los elementos más visitados. Junto a estas categorías se ubica un número que hace referencia a la cantidad de elementos (recursos) de esta categoría almacenados en el TripleStore (RDF) este valor será extraído efectuando una consulta SPARQL, mientras que los otros serán extraídos de la BD Relacional.

Además al hacer 'clic' sobre uno de los centros que aparecen en el listado, este será mostrado sobre el mapa.

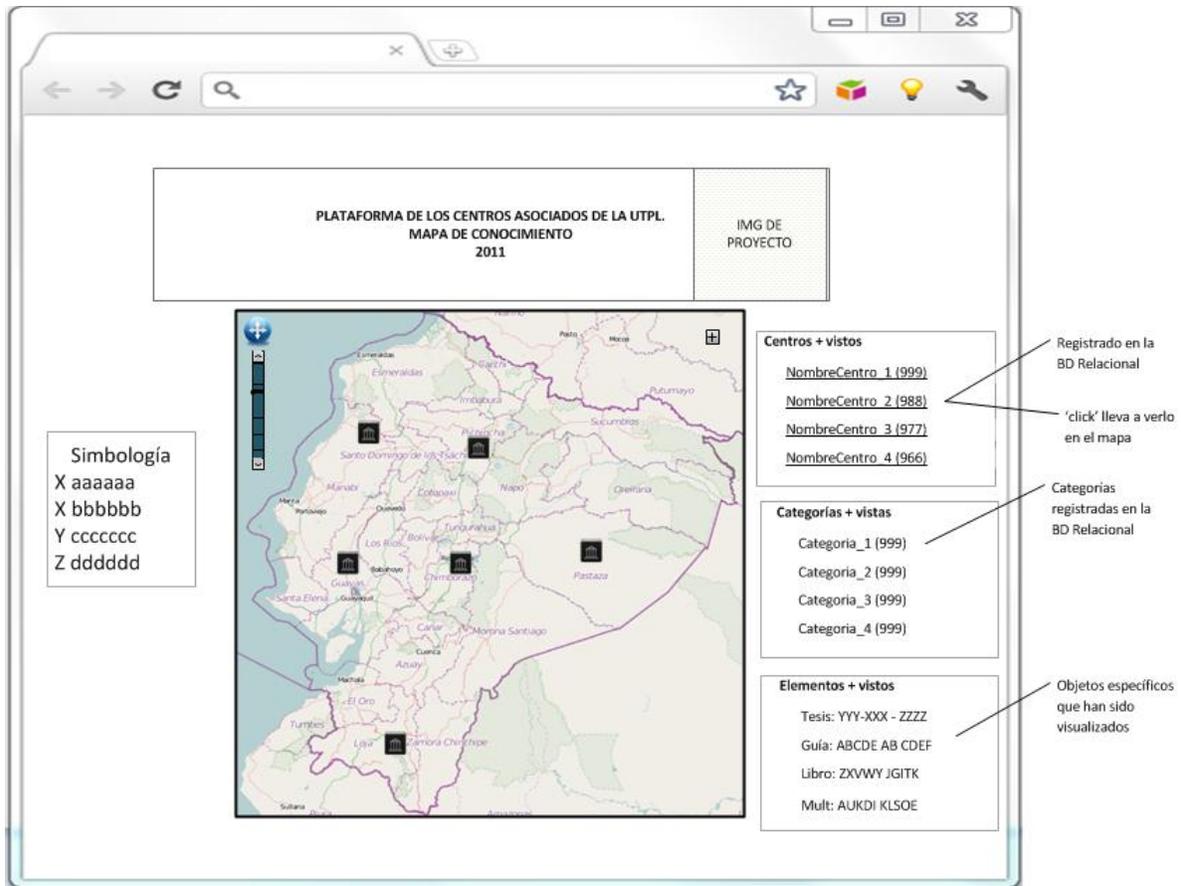


Figura 4.16. Interfaz Principal.

Al hacer 'clic' sobre uno de los centros universitarios representados por un icono, se desplegará un 'PopUp' como el que se muestra en la Figura 4.17.

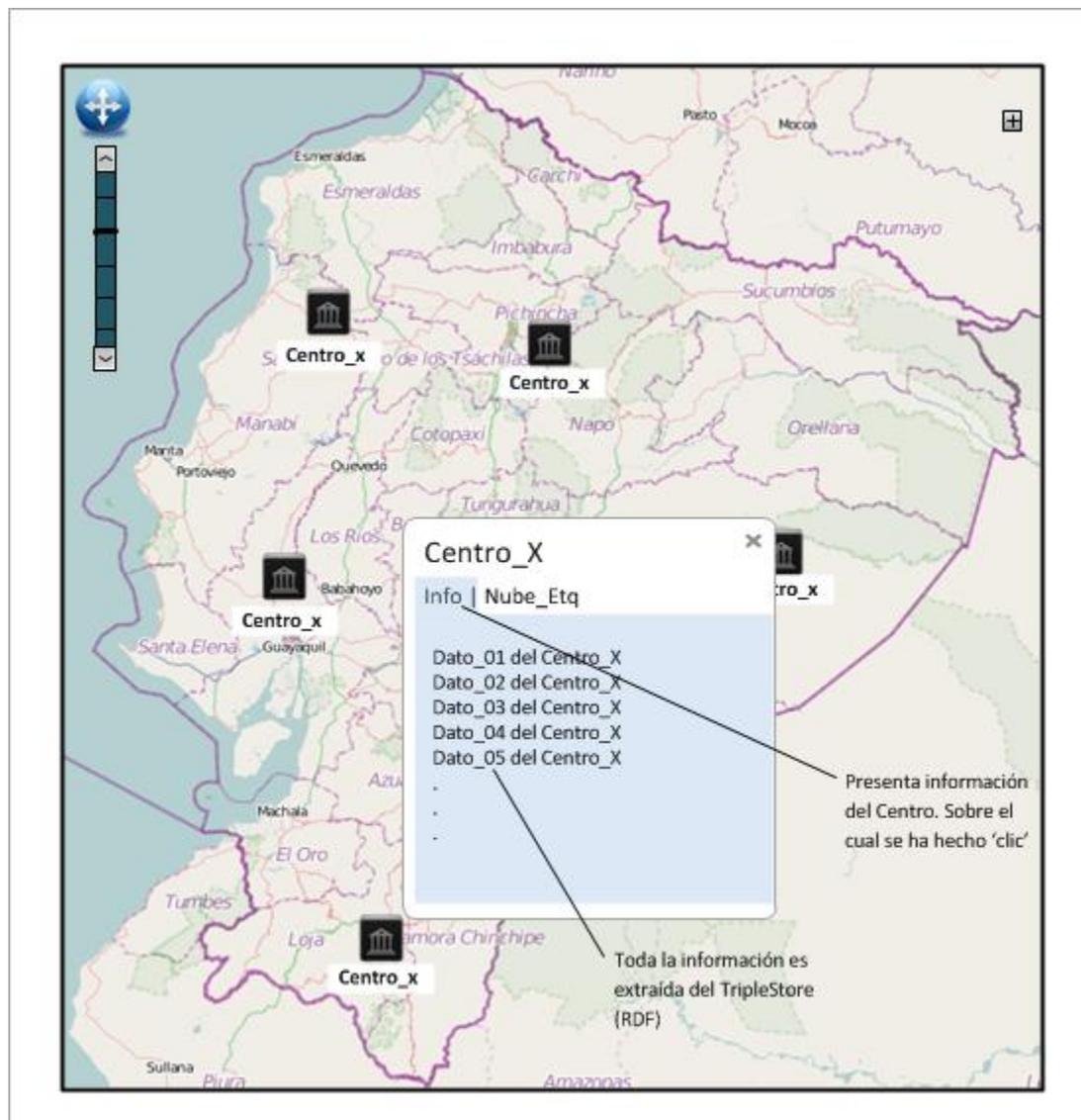


Figura 4.17. Interfaz seleccionando Centro Universitario.

Bajo la palabra 'Info' se desplegará un conjunto de información relativa al Centro Asociado, información que ha sido almacenada en el TripleStore en formato RDF.

Dentro de este mismo PopUp se mostrará información relativa a todas las categorías existentes en este Centro Asociado, que se muestra en la Figura 4.18.

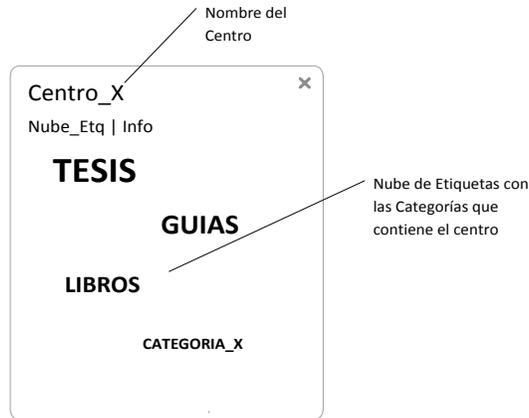


Figura 4.18. Categorías del Centro Universitario

Por cada categoría se mostrará un conjunto de 5 elementos con solamente dos atributos cada uno, como se muestra en la Figura 4.19, para visualizar otros elementos es necesario utilizar los enlaces 'anteriores' y 'siguientes' que permitirán visualizar otros 5 elementos.

Debajo de cada uno de los elementos se encuentra ubicado un link con la leyenda 'mas+', el mismo permitirá visualizar todos los atributos de este elemento en un PopUp Diferente.

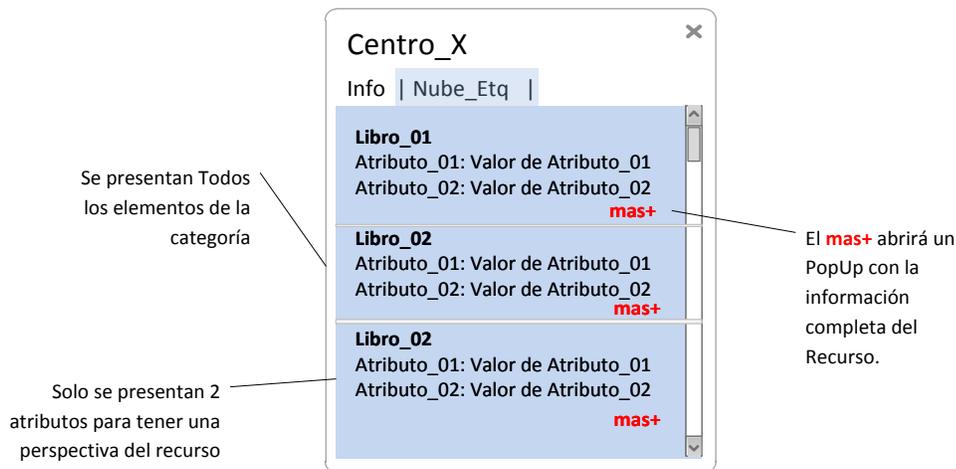


Figura 4.19. PopUp - Centro Universitario.

En el PopUp mostrado luego de hacer 'clic' sobre el enlace 'mas+' de un elemento, puede ser encontrado en la Figura 4.20. Se mostrarán todos los atributos relacionados con este elemento y que están definidos en la estructura

RDF, en el caso de ser un recurso descargable, como una guía o tesis. Se presentará un link en la parte inferior derecha con la leyenda 'Descargar'.

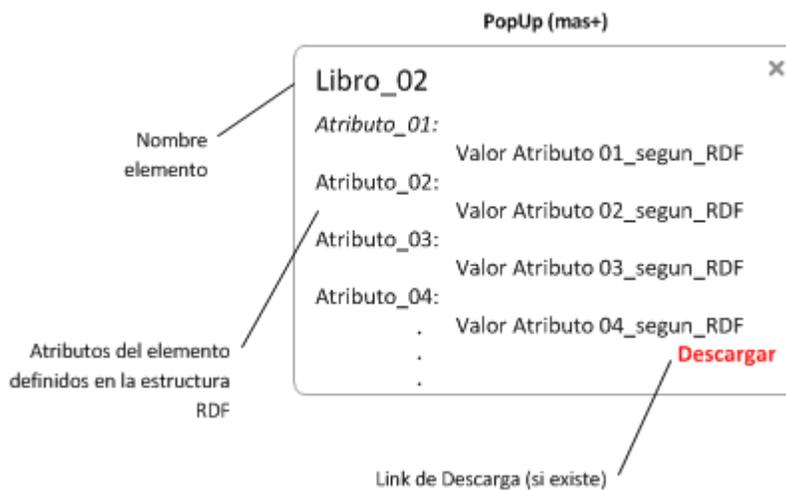


Figura 4.20. PopUp (mas+) - Centro Universitario.

Se debe mencionar que toda la información será extraída del TripleStore, en este caso Virtuoso Server, mediante el lenguaje de consulta SPARQL haciendo uso de una petición HTTP. Esta petición será realizada mediante Ajax y la respuesta será recibida en formato JSON que será posteriormente procesada y graficada desde JavaScript.

**CAPÍTULO V:
DESARROLLO DE
APLICACIÓN**

DESARROLLO DE LA APLICACIÓN

La construcción de la plataforma GERBI-MAP ha conllevado la integración de varias tecnologías con un solo fin, haciendo posible la representación de información almacenada como tripletas (RDF) de una forma totalmente transparente para el usuario final. Todo este proceso se lo detalla en el presente capítulo.



Figura 5.1. Proceso previo de la información

La secuencia seguida para el tratamiento de la información puede ser observada en la Figura 5.1, donde se pueden distinguir cuatro estados principales, antes de poseer un Store de Tripletas totalmente funcional.

5.1. RECOLECCIÓN DE INFORMACIÓN

En base a los objetivos planteados, el presente proyecto hace uso de información relativa a los centros universitarios de la UTPL, así como de información referente a recursos bibliográficos de los mismos, para su posterior representación de la información sobre un mapa del conocimiento.

La información utilizada para la representación de los centros, así como la información relativa a los mismos, ha sido obtenida de la Unidad Civil Geología y Minas de la UTPL, conocida como UCG. De esta forma se ha facilitado el acceso a un archivo shapefile⁶⁰ con información relativa a la geoLocalización de los centros universitarios.

⁶⁰ http://www.geogra.uah.es/gisweb/practica-vectorial/Formato_Shapefile.htm

5.1.1. CONVERSIÓN DE COORDENADAS

Como se especificó en capítulos anteriores, la representación de los centros universitarios será realizada sobre un mapa geo-referenciado, en este caso GoogleMaps, donde este hace uso de un sistema de coordenadas decimales bajo el estándar WGS84⁶¹, pero el archivo shapeFile recibido de la Unidad UCG posee un sistema de coordenadas UTM⁶², teniendo que realizar la conversión respectiva.

Para ello se ha hecho uso de un aplicativo desarrollado por Bashir Research, bajo el nombre de *Coord-converter* y que se basa en el trabajo de las librerías conocidas como SEDRIS⁶³. Transformando una coordenada UTM a WGS84 como se muestra en el siguiente ejemplo:

UTM

X(este): 708401.778356262

Y(norte): 9642292.92573448

WGS84

Latitud: -3.23451199999883

Longitud: - 79.1245126500035

⁶¹ http://en.wikipedia.org/wiki/World_Geodetic_System

⁶² http://en.wikipedia.org/wiki/Universal_Transverse_Mercator_coordinate_system

⁶³ <http://www.sedris.org/>

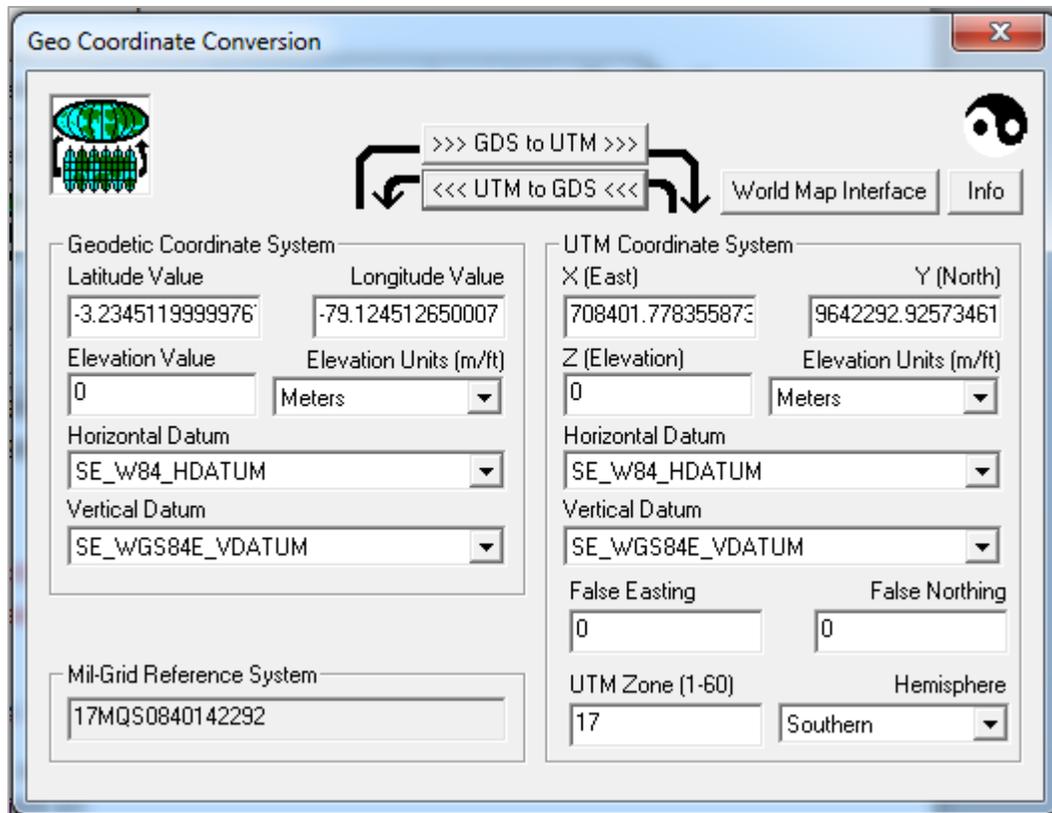


Figura 5.2. Conversor de Coordenadas

Este proceso tuvo que ser aplicado a los 29 centros seleccionados, que forman parte del proyecto GERBI-MAP, un listado completo de estos centros puede encontrarse en el **Anexo E**. La interfaz principal del aplicativo puede observarse en la Figura 5.2.

Al igual que la selección de los centros, se ha llevado a cabo la selección de 3 categorías bibliográficas principales, a las cuales se le ha sumado una cierta cantidad de individuos que las representen. La información de categorías e individuos ha sido facilitada por el Centro Asociado de Cariamanga, en un archivo de Excel.

De esta manera se ha seleccionado y dividido los individuos en las siguientes categorías:

Tabla 5.1. Individuos por Categoría.

Categorías	Individuos
LIBROS	2638
GUÍAS	169
TESIS	30
MULTIMEDIA	201

A su vez los individuos de todas las categorías han sido repartidos entre los 29 centros de forma aleatoria, permitiendo de esta manera tener un escenario totalmente funcional que represente el futuro levantamiento de información de cada centro para incrementar la población del Store.

5.2. EDICIÓN DE INFORMACIÓN

Siguiendo el proceso, una vez determinados los orígenes de la información, ser recolectada y definir los fines de la misma. Se debe considerar la estandarización de los datos, para la implementación de un proceso automático que permita crear el Store.

Esta estandarización fue aplicada a la información extraída de los dos archivos, tanto de los centros universitarios como de los datos bibliográficos. Se consideraron aspectos como:

- Definición de caracteres separadores
- Eliminación de caracteres extraños incluidos los símbolos '%', '\$', & que pueden presentar problemas al crear tripletas.
- Revisión ortográfica
- Separación de campos
- Cambio de simbología (números) por nombres reales.

Entre otros, se debe mencionar que el proceso de determinar qué información es relevante para el proyecto, involucró la mentalización de la estructura RDF a utilizar para el Store, en donde se definen en primera instancia las clases involucradas y sus relaciones, como se muestra en la Figura 5.3.

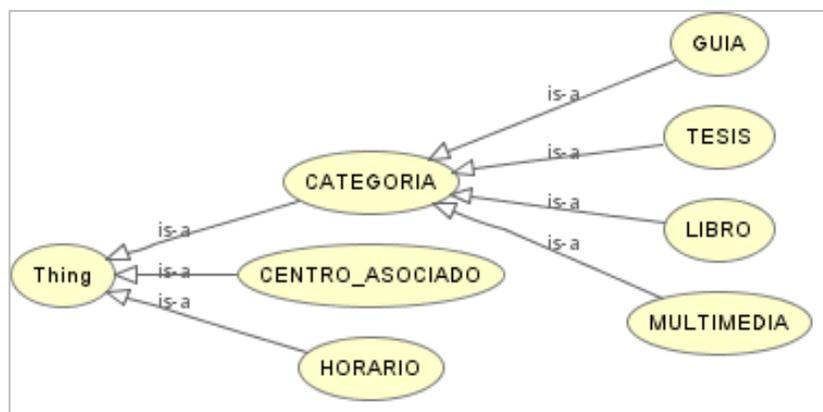


Figura 5.3. Diagrama RDF – GERBI_MAP.

El proceso de definición de estas entidades, fue realizado en Protégé, por las facilidades presentadas en capítulos anteriores. Considerando esta estructura y la información recolectada se procedió a definir el conjunto de propiedades que definirán a cada clase y permitirán hacer una relación entre las mismas.

Para esta definición, se consideraron dos tipos de propiedades existentes, propiedades de datos que pueden contener un valor específico (literal) y las propiedades objeto cuyo valor es un individuo de una clase, lo que permite crear relaciones o lazos entre las entidades.

Una lista de las propiedades de datos seleccionadas se muestra a continuación.

Tabla 5.2. Propiedades RDF.

Propiedades de Datos	
anio	horaI
area	isbn
autor	keyWord
canton	lat
colección	lon
coordinador	lugEvaluacion
dia	observacion
direccion	poblacion
edicion	provincia
editorial	resumen
email	sutTitulo
enlace	telefono

fax	tipoCentro
fechaP	titulo
horaF	uri

En este caso, la estructura RDF solo se ha definido para contener dos propiedades del tipo Objeto, en este caso estas se muestran a continuación:

Tabla 5.3. Propiedades tipo Objeto - RDF.

Propiedad	Tipo de Objeto	Descripción
bibliotecaIndv	CATEGORIA	Referencia a un individuo de cualquiera de las cuatro categorías.
horario	HORARIO	Enlaza un individuo horario.

Una vez definidas la estructura y propiedades a utilizar en el procesamiento de la información y posteriormente para la construcción del documento RDF, se ha considerado la definición de una URI para identificar los recursos, en este caso la URI será la siguiente:

<http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl>

Siendo este un caso demostrativo, se ha considerado la utilización de esta URI, tomando en cuenta la necesidad de ser redefinida en caso de expandir el modelo de datos a otras funcionalidades.

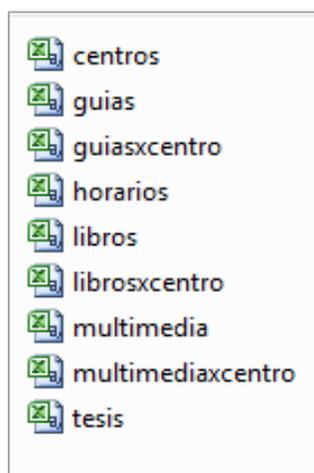


Figura 5.4. Archivos CSV

Todo este proceso de edición de información, ha sido llevado manualmente, puesto que ha sido necesario clasificar la información. El resultado de este proceso fue la creación de un documento de Excel con varias hojas en la cuales se agrupaban los datos específicos por cada categoría. Estos datos a su vez fueron almacenados como documentos separados por comas (CSV) para ser procesados por un aplicativo propio, que se explicará en el siguiente apartado.

5.3. PROCESAMIENTO DE INFORMACIÓN

Una vez definida y editada la información que será representada en formato RDF, se procede al procesamiento automático de conversión entre los archivos de CSV a uno completo de tripletas (RDF).

Para este proceso se ha creado una aplicación propia en el lenguaje java, en la misma que se ha especificado la estructura que manejará el documento así como los nombres de los archivos que contienen la información y el orden en que serán procesados.

La estructura de los individuos (RDF) ha sido extraída de Protégé y se puede apreciar en la Figura 5.5. Donde se aprecia una estructura general de un individuo de la categoría LIBRO.

```
<owl:NamedIndividual rdf:about = var_dirVocabulario>
  <rdf:type rdf:resource= var_dirVocabulario # TIPO_INDV/>
  <edicion rdf:datatype="&xsd:int"> </edicion>
  <anio rdf:datatype="&xsd:int"> </anio>
  <isbn rdf:datatype="&xsd:string"> </isbn>
  <area rdf:datatype="&xsd:string"> </area>
  <subTitulo rdf:datatype="&xsd:string"> </subTitulo>
  <titulo rdf:datatype="&xsd:string"> </titulo>
  <autor rdf:datatype="&xsd:string"> </autor>
  <editorial rdf:datatype="&xsd:string"> </editorial>
  <keyWord rdf:datatype="&xsd:string"> </keyWord>
</owl:NamedIndividual>
```

Figura 5.5. Estructura de Individuo (general)

La salida del aplicativo luego de realizar el procesamiento, será una estructura RDF totalmente funcional, que contiene todos los elementos definidos así como toda la información que fue recolecta y seleccionada.

Una vez terminado el proceso se procederá a almacenar toda esta información en un TripleStore que permita la extracción de información mediante un lenguaje de consulta (SPARQL).

En la Figura 5.6, se puede apreciar los diversos métodos utilizados dentro del aplicativo encargado de realizar la migración, como se mencionó, el orden en que se ejecutan es importante, debido a que existen relaciones que se deben satisfacer.

```
public static void main(String[] args) {  
  
    new Principal().cargarLibros();  
    new Principal().cargarTesis();  
    new Principal().cargarGuias();  
    new Principal().cargarMultimedia();  
    new Principal().cargarCentros();  
  
}
```

Figura 5.6. Métodos de extracción.

El proceso llevado a cabo en cada uno de estos métodos, sigue el proceso de cargar el archivo .CSV, recorrer una a una sus líneas y realizar la separación de las mismas mediante el operador ‘;’, como se muestra en la Figura 5.7 (más detalle **Anexo D**).

```
public void cargarCentros() {  
    try {  
        FileInputStream fis = new FileInputStream(new File("c:/infoRDF/centros.csv"));  
        InputStreamReader is = new InputStreamReader(fis, "ISO-8859-1");  
  
        BufferedReader in = new BufferedReader(is);  
        String str;  
        while ((str = in.readLine()) != null) {  
            str = str.replace("&", "y");  
            String[] parts = str.split(";");  
        }  
    }  
}
```

Figura 5.7. Métodos de extracción.

5.4. ALMACENAMIENTO

Como se mencionó en secciones anteriores el TripleStore seleccionado para el almacenaje de datos es Virtuoso OpenLink en su versión OpenSource. Para este proyecto, el mismo ha sido instalado sobre la plataforma Linux, el proceso de instalación puede ser observado en el **Anexo B**.

Una vez instalado el servidor de tripletas, se debe definir un nombre para el grafo que se va a crear y donde se almacenará toda la información anteriormente procesada. En este caso el nombre del grafo será el siguiente:

<http://gerbi.utpl.edu.ec:8890/dataGerbiMap>

Este nombre será utilizado al momento de la creación del Store, así como al extraer información mediante SPARQL. El mismo representa una IRI (Internationalized Resource Identifiers)⁶⁴.

Para la creación del Store dentro del Servidor es necesario realizar el siguiente procedimiento:

1. Iniciar sesión en la interfaz principal 'CONDUCTOR' de Virtuoso OpenSource.
2. Ubicar la Pestaña 'Linked Data'.
3. Ubicar la Pestaña 'Quad Store Upload'.

En esta opción se debe definir el archivo que contiene las tripletas, en este caso el generado por la aplicación en el punto anterior, así como definir el nombre que tendrá. Las opciones se las puede observar en la Figura 5.8. Una vez especificados estos pasos se debe utilizar el botón 'Upload' para crear el Store.

⁶⁴ <http://www.w3.org/International/iri-edit/draft-duerst-iri.HTML>

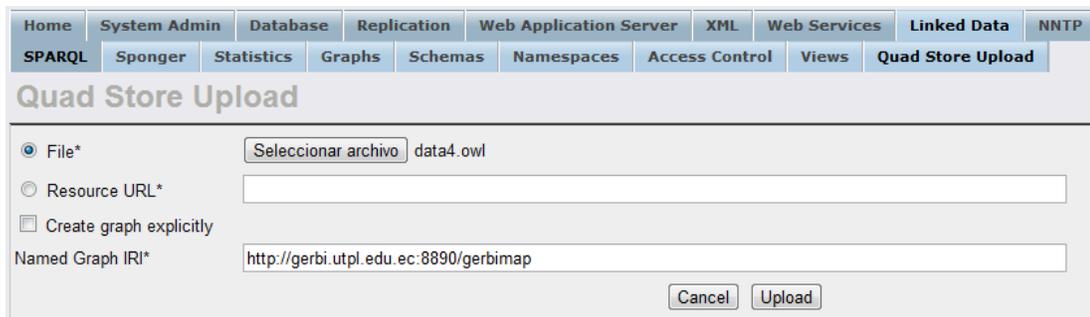


Figura 5.8. Creación de Store – Virtuoso

Una vez terminado el proceso de subida y creación del Store, se observa el nombre del grafo dentro de los disponibles, bajo la pestaña ‘Graphs’. Indicando así, que el proceso ha sido terminado con éxito y que es posible realizar consultas sobre el grafo para recuperar | modificar la información.



Figura 5.9. Grafo en la Lista de Virtuoso

5.4.1. VALIDACIÓN DE GRAFO - SPARQL

Una vez terminado con éxito el proceso de creación del grafo, es posible utilizar el EndPoint propio de Virtuoso para efectuar extracción de información mediante el uso del lenguaje de consulta SPARQL. Este EndPoint puede ser localizado en la siguiente dirección:

<http://gerbi.utpl.edu.ec:8890/SPARQL>

Es necesario especificar el nombre del grafo sobre el cual se va a realizar la consulta.

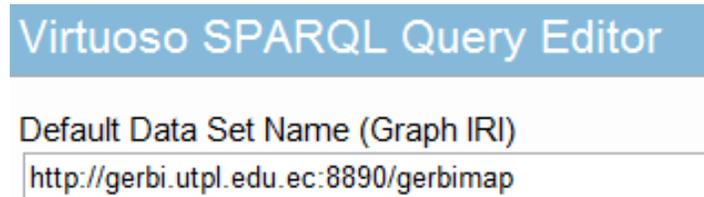


Figura 5.10. Especificación de Grafo – EndPoint

Para la demostración del funcionamiento del Store y el EndPoint se muestra a continuación algunas consultas de información mediante SPARQL y su resultado, en este caso para mejor visualización los resultados son presentados en formato HTML.

a. Extracción de todas las propiedades del individuo *guia101*

SPARQL

```

PREFIX
:<http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#>
SELECT ?verb ?pred
WHERE {
    :guia101 ?verb ?pred
}

```

Resultado

verb	pred
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#area	Ciencias de la Computación
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#autor	S.A.
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#edicion	1
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#editorial	U.T.P.L.
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#isbn	S/N
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#subTitulo	Guía Didáctica
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#titulo	Base de Datos I

b. Consulta de todos los individuos que son del tipo Libro

SPARQL

```

PREFIX
:<http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#>
SELECT ?indv

```

```

WHERE {
  ?indv a :LIBRO
}

```

Resultado

indv
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#libro0001
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#libro0002
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#libro0003
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#libro0004
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#libro0005
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#libro0006
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#libro0007

c. Extracción de todas las propiedades del centro22

SPARQL

```

PREFIX
:<http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#>
SELECT ?a ?b
WHERE {
  :centro22 ?a ?b
}
ORDER BY desc(?a)

```

Resultado

a	b
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#tipoCentro	Provincial
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#telefono	830189 832930 833986
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#provincia	Sucumbios
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#poblacion	NUEVA LOJA (LAGO AGRIO)
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#observacion	No registradas
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#lugEvaluacion	Esc. 22
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#lon	-76,88092
http://www.dirvocabulary.org/ontology/material/bibliografico/data3.owl#lat	0,08553

Como se puede apreciar en los resultados, al extraer la información desde el EndPoint de Virtuoso, esta es totalmente consistente y a su vez, permite ser consulta en base a diversos criterios.

5.4.2. ALMACENAMIENTO DE ESTADÍSTICAS

Para cumplir con otro de los objetivos del proyecto GERBI-MAP, el cual consiste en mostrar estadísticas de acceso a recursos. Se procede a la construcción de un esquema relacional de base de datos en donde serán almacenadas y posteriormente consultadas las estadísticas. Este diagrama puede ser observado en la Figura 5.11.

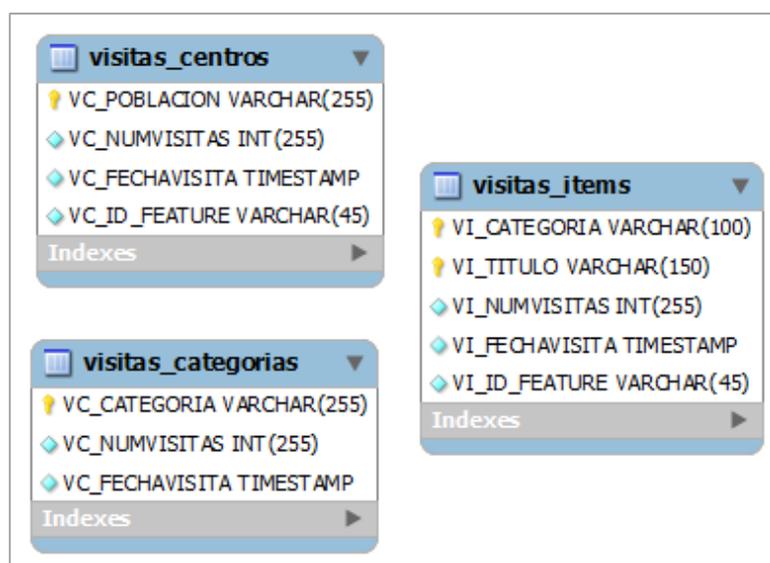


Figura 5.11. Registro de Estadísticas

Para el registro de estadísticas vinculadas con los centros e ítems se utiliza las entidades `visitas_centros` y `visitas_items`, que comparten una estructura similar, con los siguientes atributos:

- **VC_POBLACION:** Almacena un identificador único por cada centro, el mismo que referencia la población en donde se encuentra.
- **VC_NUMVISITAS:** Registra un contador, el mismo que inicia en 1, la primera vez que se visite un centro y luego se irá incrementando.

- **VC_FECHAVISITA:** Registra la última fecha en que se visitó el centro.
- **VC_ID_FEATURE:** Registra un identificador único, el mismo que sirve para localizar el centro sobre el mapa.

De igual manera para el registro de visitas según categorías, la entidad utilizada es **visitas_categorias**, con los atributos:

- **VC_CATEGORIA:** Indica el nombre de la categoría que se ha visitado.
- **VC_NUMVISITAS:** Registra un contador único por categoría, que se irá incrementando a medida que se visita una categoría.
- **VC_FECHAVISITA:** Registra la última vez en que se visitó la categoría.

Con la ayuda de este esquema el registro y consulta de estadísticas será realizado de una forma rápida y sencilla.

5.5. VISUALIZACIÓN DE INFORMACIÓN

La representación de la información recolectada será realizada sobre un mapa geo-referenciado (GoogleMaps), a su vez todo el contenido será modelado sobre una página web cuyo diseño se mostró en el capítulo anterior.

El esquema de funcionamiento, puede ser apreciado en la Figura 5.12. Hasta esta parte del proyecto se cuenta ya con el almacenamiento terminado y totalmente funcional, así mismo se han realizado pruebas de recuperación de información que han sido satisfactorias.

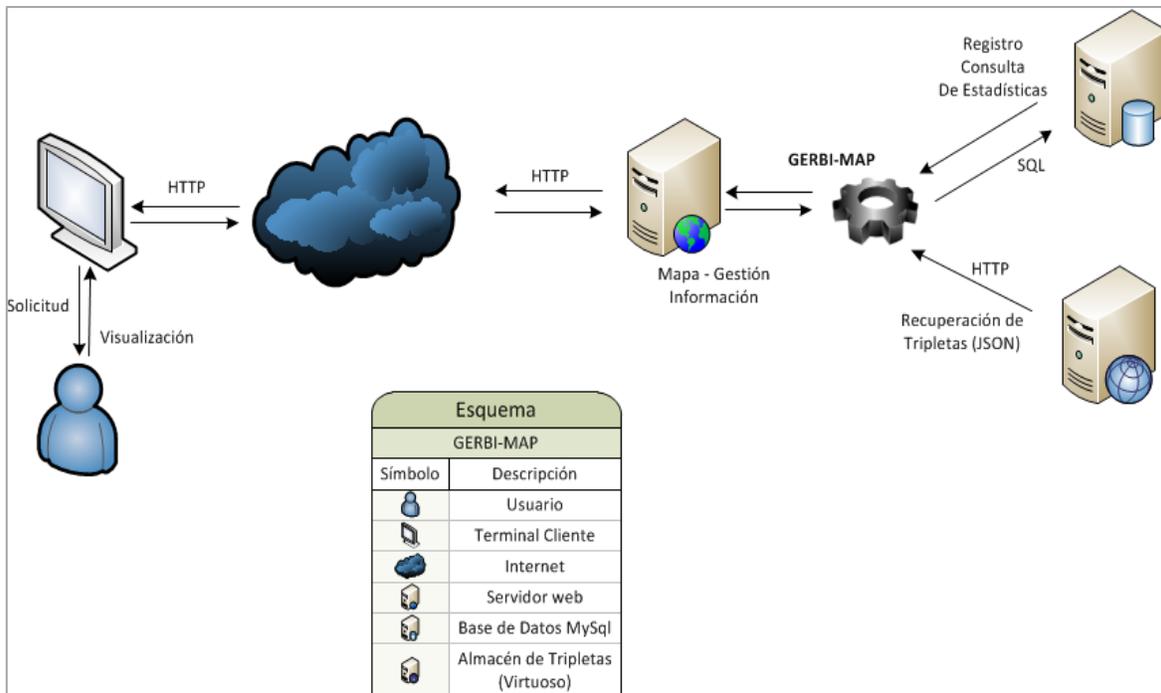


Figura 5.12. Esquema de Funcionamiento - GERBI-MAP

A continuación se mencionan todas las partes involucradas en el proceso de visualización de información, incluidas aquí también el registro de estadísticas y sus respectivas consultas.

5.5.1. RECUPERACIÓN DE INFORMACIÓN

La comunicación entre el Store que contiene las tripletas y la aplicación web que gestionará el contenido, ha sido realizada mediante la utilización del protocolo HTTP contra el EndPoint de Virtuoso, de esta forma se puede enviar en una petición HTTP una consulta de SPARQL para que la misma sea devuelta en el formato especificado.

Para ello se debe formar una URL de forma similar a la siguiente:

[http://dirServidor:8890/SPARQL?\[string_consulta\]](http://dirServidor:8890/SPARQL?[string_consulta])

La misma que hace referencia al EndPoint del servidor, donde la parte que se describe como [string_consulta] deberá ser la concatenación de los siguientes parámetros:

- **default-graph-uri:** Indica el grafo por defecto que será utilizado en la consulta de SPARQL.
- **query:** Consulta SPARQL.
- **debug:** Indicador para mostrar información de error en caso que se produzca uno. En ambientes finales este parámetro se ajusta a 'off'.
- **timeout:** Especifica el tiempo máximo que una consulta puede demorar sobre el Store.
- **format:** Indica el formato en que se espera sea devuelta la respuesta, en este caso será especificado a JSON, colocando el parámetro a 'application/JSON'.

De esta forma se recibirá el resultado de la consulta en un formato similar al mostrado en la Figura 5.13.

```

"head": {
  "link": [
  ],
  "vars": [
    "a",
    "b"
  ]
},
"results": {
  "distinct": false,
  "ordered": true,
  "bindings": [
    {
      "a": {
        "type": "uri",
        "value": "http:
      },
      "b": {
        "type": "typed-
        datatype": "ht

```

Figura 5.13. Extracto de respuesta JSON

Dentro de la aplicación, este proceso se lleva a cabo por una función creada en PHP bajo el nombre de **simpleQueryGet**, la misma que recibe como parámetro de entrada la consulta en lenguaje SPARQL y realiza la petición HTTP mediante la función **file_get_contents**⁶⁵. Devolviendo de igual forma todo el contenido de la petición. La función puede ser observada en la Figura 54.

```
function simpleQueryGet($query) {

    $dirIpServer = "192.168.146.165";
    $grafo = "http://$dirIpServer:8890/testFinal";
    $format = "application/json";
    $endPoint = "http://$dirIpServer:8890/sparql/";
    $vocab = "http://www.dirvocabulary.org/ontology/material/";

    $params = array(
        "default-graph-uri" => $grafo,
        "query" => $query,
        "debug" => "on",
        "timeout" => "",
        "format" => $format
    );

    $querypart = "?";
    foreach ($params as $name => $value) {
        $querypart .= $name . '=' . urlencode($value) . "&";
    }

    $sparqlURL = $endPoint . $querypart;

    $salida = file_get_contents($sparqlURL);

    return utf8_encode($salida);
}
```

Figura 5.14. Función para petición HTTP al Store

De esta forma, se puede manipular posteriormente todo el contenido devuelto por el servidor. Proceso que será mostrado en el siguiente apartado.

⁶⁵ <http://php.net/manual/es/function.file-get-contents.php>

5.5.2. PROCESAMIENTO DE INFORMACIÓN

El procesamiento de la información, así como la petición previa, son realizadas en el lado del cliente (navegador), por tal motivo son realizadas utilizando el lenguaje JavaScript.

El primer paso para ello es realizar una petición al archivo PHP que contiene la función encargada de traer la información del Store. Para mejorar la eficiencia de este proceso se ha hecho uso del framework JQuery⁶⁶ el mismo que permite la utilización de Ajax para mejorar la carga de información del lado del servidor. La estructura de una petición puede ser observada en la Figura 5.15.

```
$.ajax({
  type: "POST",
  url: "getData.php",
  async: false,
  data: "q=" + querySparql ,
  success: function(datos) {
    //acción de éxito
  },
  error: function (obj, error, objError) {
    //acción de error
  }
});
```

Figura 5.15. Petición Ajax – JQuery

De esta forma se envía como parámetro, en una petición POST, la consulta de SPARQL, este parámetro es recuperado por el archivo PHP y enviado a la función respectiva para que se comunique con el Store de Virtuoso. Se debe notar que esta operación tiene integrados dos estados, para reconocer si la petición se pudo efectuar o no (sucess | error).

⁶⁶ <http://jquery.com/>

```
extract($_POST);  
  
echo simpleQueryGet($q);
```

Figura 5.16. Recuperación de Variable POST

Esta función es utilizada para recuperar información del Store en las siguientes ocasiones:

- Cargar todos los centros asociados
- Consultar las categorías que un determinado centro asociado posee
- Mostrar los Individuos de una determinada categoría y centro.
- Consultar toda la información de un Individuo.

Una vez detectado el éxito de una petición AJAX (función success) se procede a procesar la información, en primera instancia, se debe recalcar que la respuesta recibida, la cual se encuentra en la variable **datos**, es un objeto JSON, lo que facilita enormemente su manipulación.

En base a la estructura JSON mostrada en la Figura 5.13: Extracto de respuesta JSON, se puede determinar que los datos resultantes de la petición se encuentran bajo la directiva 'bindings' y vienen conformados como elementos de un Array. De esta forma para recorrer todos los resultados disponibles de una consulta SPARQL se debería realizar el proceso descrito en la Figura 5.17.

```
var rtArray = datos.results.bindings;  
for(var i=0; i<rtArray.length; i++){  
    //Extraer Individuo  
    var elemento = rtArray[i];  
    //extraer valores (atributos)  
    var atr1 = elemento.atr1.value;  
    var atr2 = elemento.atr2.value;  
    //procesar valores  
}
```

Figura 5.17. Extracción Atributos JSON

Lo que permite extraer todos los atributos definidos como respuesta en la consulta de SPARQL y recorrerlos hasta el final. Esto demuestra una gran interoperabilidad entre JavaScript - JSON - Virtuoso, lo que permite que la integración y el intercambio de información sea realizado de forma transparente.

5.5.3. VISUALIZACIÓN

Solventada la extracción de información desde el TripleStore, así como su procesamiento, la representación gráfica constituye el principal y último punto del proyecto.

Siendo necesario para ello la utilización de diversos elementos interrelacionados entre sí, que se detallan a continuación.

5.5.3.1. IMPLEMENTACIÓN MAPA

Como se ha mencionado anteriormente se hace uso del servicio de mapas de Google, sobre el cual se ubican cada uno de los centros universitarios de la UTPL seleccionados detalle en **Anexo E**. De esta manera la instanciación y creación del mapa, es realizada mediante el uso del framework OpenLayers. Para ello es necesario realizar la estructura mostrada en la Figura 5.18.

Donde se pueden apreciar las distintas configuraciones que se aplican al mapa, como la Barra de Zoom (PanZoomBar), deshabilitar el zoom con el scroll del mouse (zoomWheelEnabled). De igual forma se define el área máxima que se podrá visualizar (maxEntent) y el tipo de proyección que utilizarán las coordenadas (projection).

```

//Controles del mapa
var options = {
  controls : [
    new OpenLayers.Control.Navigation({
      zoomWheelEnabled : false
    }),
    new OpenLayers.Control.PanZoomBar(),
    new OpenLayers.Control.KeyboardDefaults(),
    new OpenLayers.Control.MousePosition()
  ],
  units: "m",
  numZoomLevels : 19,
  maxResolution : 'auto',
  restrictedExtent : extent,
  maxExtent: new OpenLayers.Bounds(
    -20037508.34, -20037508.34,
    20037508.34, 20037508.34)
  ,
  projection: "EPSG:4326"
};
map = new OpenLayers.Map('map',options);
var gmap = new OpenLayers.Layer.Google("Google Maps", {
  sphericalMercator:true
});
map.addLayers([gmap]);

```

Figura 5.18. Inicialización de Mapa

OpenLayers gestiona todos los elementos basándose en un estilo por capas (layers), de tal manera que se coloca a GoogleMaps como una capa. Indicando además que este servicio de mapas utiliza un sistema especial de coordenadas conocido como 'Spherical Mercator'. Mediante este proceso nuestro mapa está totalmente configurado y disponible, como se muestra en la Figura 5.19.

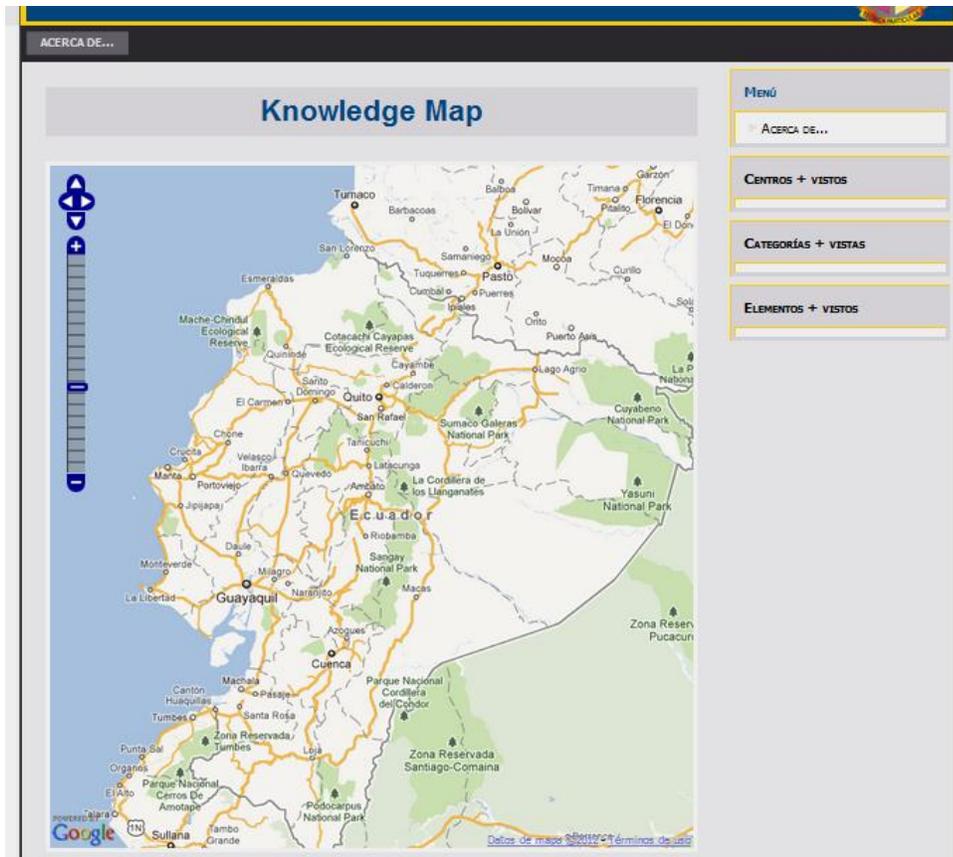


Figura 5.19. Instancia de Mapa inicial.

Se debe indicar, que dentro de las configuraciones iniciales del mapa, se ha utilizado la propiedad **maxExtent**, la misma que determina la máxima zona visible para el usuario, limitando con esta solamente la vista a todo el territorio nacional. La definición de este, se lo realiza mediante coordenadas geográficas como se muestra en la Figura 5.20. Donde se puede apreciar en la línea 1, se hace uso del objeto Bounds, propio del framework OpenLayers que permite definir un área geográfica, de esta forma se definen las coordenadas superior izquierda e inferior derecha que formarán el área visible, en la línea 2 y 3 respectivamente.

Una vez determinado esto, es necesario convertir estas coordenadas al sistema de coordenadas que GoogleMaps utiliza, para ello se hace uso de la función **transform**, indicando el sistema de origen y destino, como se aprecia en la línea 4.

```

1. var extent = new OpenLayers.Bounds();
2. extent.extend(new OpenLayers.LonLat(-93.12012,2.94434));
3. extent.extend(new OpenLayers.LonLat(-74.48730,-5.40527));
4. extent.transform( new OpenLayers.Projection( "EPSG:4326" ),
    new OpenLayers.Projection( "EPSG:900913" ));

```

Figura 5.20. Coordenadas para limitar a territorio nacional.

5.5.3.2. INDIVIDUOS

El proceso de ubicar individuos, también conocidos como Features⁶⁷ en OpenLayers es realizado también sobre una capa (layer), sin embargo es necesario previamente fijar un estilo que los mismos tendrán, en donde se pueden definir propiedades como color, tamaño, icono a mostrar e incluso incluir propiedades propias. En la Figura 5.21 se puede apreciar el estilo definido para los centros asociados.

```

// Estilo para centros asociados
var stl_centroAsociado = new OpenLayers.StyleMap( {
  externalGraphic : "${ico}",
  graphicWidth : 25,
  graphicHeight : 50,
  poblacion: "${pob}",
  canton: "${can}",
  provincia : "${pr}",
  tipoCentro : "${tp}",
  coordinador : "${co}",
  direccion : "${dir}",
  telefono : "${fono}",
  fax : "${fx}",
  email : "${em}",
  blog : "${bl}",
  lugEvaluacion : "${le}",
  observacion : "${ob}",
  horario: "${hor}",
  fontColor: "${favColor}",
  fontSize: "12px",
  fontFamily: "Courier New, monospace",
  fontWeight: "bold",
  labelAlign: "${align}",
  labelOffset: new OpenLayers.Pixel(0,-20)
}
);

```

Figura 5.21. Estilo centros universitarios – OpenLayers

⁶⁷ En adelante los términos individuos y features serán usados indistintamente, pues expresan el mismo significado.

Como se puede observar existen propiedades únicas como *tipoCentro* o *provincia* que servirán posteriormente para almacenar datos referentes a los centros. Una vez definido el estilo la representación gráfica consiste en crear un punto geográfico a partir de coordenadas de latitud y longitud, y añadir el individuo a la capa respectiva, como se muestra en la Figura 5.22.

```
// Posicion lon : lat
var point = new OpenLayers.Geometry.Point(ln,lt);
point.transform( new OpenLayers.Projection( "EPSG:4326" ),
    new OpenLayers.Projection("EPSG:900913") );
//creacion de centro con atributos propios
// ...
// ...
layer_centroAsoc.addFeatures( [centroAsocFeature] );
```

Figura 5.22. Creación de Centros

Realizado este procedimiento se puede visualizar cada uno de los elementos creados, geográficamente localizados sobre el mapa, con un icono representativo, como se muestra en la Figura 5.23. La simbología de estos puede ser encontrada en el **Anexo F**.

El agregar un globo de información (PopUp), es realizado de forma relativamente sencilla. En este globo se muestra información relativa al centro así como la nube de etiquetas en relación a las categorías del centro.

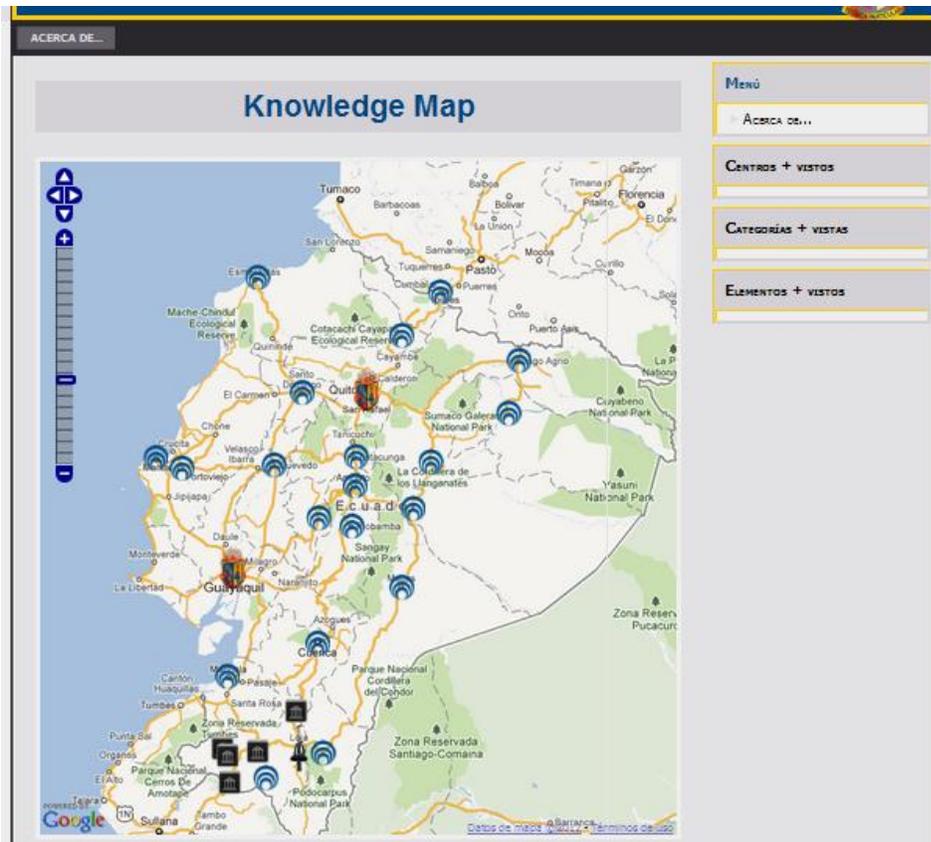


Figura 5.23. Centros universitarios sobre el GoogleMaps

Para indicar este comportamiento es necesario agregar una configuración adicional al mapa como se muestra en la Figura 5.24.

```

//Evento al hacer clic sobre un centro asociado
eventoSeleccion = new OpenLayers.Control.SelectFeature(
  [ lnz_centroAsoc ],
  {
    clickout: true,
    toggle: false,
    multiple: false,
    hover : false,
    onSelect : function(feature) {
      onCentroSelecc(feature) ;
    },
    onUnselect : function(feature) {
      onCentroUnselecc(feature) ;
    }
  }
);

```

Figura 5.24. Activar PopUp – Centro Asociado

Donde las funciones *onCentroSelecc* y *onCentroUnselecc* serán ejecutadas cuando se haga clic sobre un Centro Asociado y cuando se haga clic fuera de él, respectivamente. El funcionamiento de las mismas puede ser observado en el **Anexo G**.

Dentro de estos PopUps se mostrará la información relacionada con el Centro Asociado, así como una nube de etiquetas que contendrán las categorías que posee el centro asociado, tal como se visualiza en la Figura 5.25.



Figura 5.25. Vista del PopUp.

La nube de etiquetas permite visualizar las categorías de forma aleatoria y en movimiento, resaltando además las categorías que poseen mayor cantidad de individuos. Permitiendo así una mayor experiencia de usuario.

Listado de Ítems por Categoría

La presentación los individuos (ítems) que posee cada categoría asociada a cada centro se visualizan en forma paginada debido a que la cantidad de elementos por categoría es bastante grande.

Para poder conseguir este propósito se ha hecho uso de dos instrucciones de SPARQL que permiten la consulta de resultados basándose en un orden, como son: OFFSET y LIMIT.

Mediante estas se puede especificar la cantidad de resultados a devolver y donde debe iniciar, un ejemplo de estas consultas puede ser observado en la Figura 5.26.

Para su visualización solamente se ha considerado la presentación de dos propiedades por cada elemento, para facilitar la navegación. En caso de requerir observar a detalle todos los atributos de un individuo se ha facilitado la opción 'mas+' junto a cada detalle que permite visualizar en un PopUp toda la información relativa a este Individuo.

```
PREFIX :<dir_vocabulario_rdf#>
SELECT ?tit ?recur
WHERE {
  :centro1 :bibliotecaIndv ?recur.
  ?recur a ?tipo.
  ?recur :titulo ?tit.
  FILTER(?tipo = :LIBRO)
}
order by ?tit
offset 1 limit 5
```

Figura 5.26. Consulta SPARQL con las opciones offset y limit.

Se debe hacer notar que se ha utilizado dos tipos de PopUps, el primero propio del framework OpenLayers para presentar la información del centro, la nube de etiquetas y la lista de ítems, y un segundo tipo de PopUp para el que se ha utilizado el framework FancyBox⁶⁸ que permite fácilmente desplegar PopUp modales, los dos tipos pueden ser observados en la Figura 5.27.

⁶⁸ <http://fancybox.net/>

Se debe destacar que en el PopUp donde se muestra la información del Individuo se ha colocado dos propiedades URI y ENLACE, donde la primera direcciona al repositorio DSpace de la UTPL, donde se puede obtener más información sobre el material bibliográfico y la segunda permite tener un enlace desde donde se puede descargar el individuo en formato digital, estos dos atributos aplican para la categoría TESIS, donde el material se encuentra disponible para descarga.



Figura 5.27. PopUp de Listado e Información del individuo.

De igual manera, todos los elementos HTML involucrados, son insertados, reemplazados y actualizados mediante el framework JQuery, lo que permite hacer que la aplicación no se mantenga estática y no sea necesario el refresco completo de página.

Su uso es relativamente sencillo, como se muestra en la Figura 5.28, en donde se borra el contenido del *div* cuyo *id* es 'maxIndv' y acto seguido se coloca dentro de este otro, con contenido diferente.

```
$("#maxIndv").empty().append("<div>Estadísticas Centros</div>");
```

Figura 5.28. Manejo de HTML con JQuery

Al finalizar el desarrollo de la aplicación GERBI-MAP se concluye que se cumple a cabalidad con lo expuesto en los objetivos del trabajo de tesis.

5.5.3.3. ESTADÍSTICAS

El proyecto GERBI-MAP contempla la recolección de estadísticas de visita, tanto a centros universitarios representados sobre el mapa, como a recursos bibliográficos, en este caso, categorías e individuos.

Estas estadísticas permitirán mostrar en el aplicativo Web una visión general al usuario, que le permitirá guiarse en la visita al aplicativo. Y así mismo, podrán ser utilizadas en aplicativos futuros para la recomendación de contenido.



Figura 5.29. Estadísticas de búsqueda

El proceso para registrar las estadísticas, se considerará de la siguiente manera:

- Se considerará centro visitado si se ha hecho 'clic' sobre el mismo y se ha desplegado el PopUp respectivo.
- Se considera visita a una categoría, si se ha dado clic sobre una Categoría de la nube de etiquetas y se ha desplegado la lista de Items de la misma.
- Se considera una visita a un Ítem (individuo) si se ha hecho 'clic' en el enlace 'mas+' y se ha desplegado el PopUp respectivo.

En base a estas consideraciones se presenta en la parte derecha de la página, el respectivo conteo de visitas como se puede apreciar en la Figura 66. Todo el conteo y visualización de estas, es realizado mediante peticiones AJAX, lo que permite contabilizar las visitas en ese mismo instante y de igual forma actualizarlas, una petición de este tipo puede ser apreciada en la Figura 5.30. Tomando en cuenta que estos datos se almacenen en un ambiente relacional, como se indicó anteriormente.

```
function regVisitaItem(titulo, idF) {
    $.ajax({
        type: "POST",
        url:"php_connect/setEstadisticas.php",
        async: true,
        data: {
            flag: 3,
            cat: categSelect,
            tit: titulo,
            id: idF
        },
        success: function(datos){
            //nothing
        },
        error: function (obj, error, objError){
            alert('No fue posible registrar la visita a la categoria');
        }
    });
}
```

Figura 5.30. Estadísticas de búsqueda

DISCUSIÓN

DISCUSIÓN

La construcción del proyecto GERBI-MAP, ha permitido demostrar la integración de diversas tecnologías para obtener una solución capaz de brindar información de forma accesible.

De esta forma se ha podido definir a través de cada uno de los capítulos de la presente tesis, la arquitectura de la solución, la misma que utiliza diversas herramientas en un esquema web. La aplicación GERBI-MAP está enfocada en los siguientes módulos:

MIGRACIÓN DE DATOS

La migración de estos datos ha sido realizada en parte automática y en parte manual, puesto que los orígenes de datos han sido heterogéneos. En donde, se detectaron al inicio algunos inconvenientes y se buscó la mejor solución; un problema surgió en los orígenes de información no homogéneos, toda la información fue recibida en un documento Excel, posteriormente se realizó una limpieza para generar Archivos Separados por Comas (CSV); esta limpieza consistió en realizar correcciones ortográficas y la eliminación de caracteres raros (+,~,&?), luego se cambió el orden de las columnas y el relleno de campos faltantes en algunas celdas, así también se aplicó formatos homogéneos a fechas, números y texto, al final se separa la información en diversas hojas y se graba cada una con la extensión “.CSV”.

Posteriormente apareció otra dificultad, las coordenadas de los centros universitarios están en formato UTM y en un archivo .shp que para la solución de la aplicación no servían, para ello se realizó la conversión de las coordenadas de todos los centros universitarios del formato recibido (UTM) al formato admitido por GoogleMaps WGS84, para esto se utilizó una herramienta Coord-converter.

En vista de no existir una forma estándar de migrar estos datos a tripletas, se desarrolló un aplicativo propio en java, el cuál toma archivos CSV como entrada y genera una ontología OWL, solventando el problema en todos sus aspectos.

Luego de crear la ontología se procede a subir el archivo en un almacén de tripletas para posteriormente utilizarlo en la aplicación.

ALMACENAMIENTO

Para el almacenamiento del RDF se hizo uso del Store de Tripletas Virtuoso, que ha permitido que el trabajo con datos semánticos sea más fácil.

Uno de los mayores problemas que se presentó con el Store Virtuoso fue la instalación tediosa y complicada bajo el sistema operativo Windows, por tal razón se instaló Virtuoso en un sistema operativo libre como lo es Linux en su versión Ubuntu 10.4 donde no se presentaron inconvenientes. Buscar la manera de actualizar los datos del RDF fue una decisión complicada; la única solución por ahora es modificar los archivos .CSV y volver a generar el RDF, este problema de actualización/eliminación de información del RDF está considerado para trabajos futuros de la aplicación.

Luego de tener los datos (RDF) en un almacén de tripletas como lo es Virtuoso, ahora se tiene que unir el aplicativo web con Virtuoso, seguidamente se detalla el proceso de comunicación.

COMUNICACIÓN

La comunicación entre el aplicativo Web y el Store de Tripletas (Virtuoso) constituyó un elemento fundamental. Un problema surgió porque no hay estándares establecidos para la comunicación entre aplicativos de este tipo, para lo cual se utilizó peticiones HTTP hacia el EndPoint del Store, enviando y procesando todos

los resultados mediante el formato de intercambio JSON, realizando este proceso mediante el lenguaje JavaScript y la tecnología Ajax.

Finalmente luego de tener toda la solución de la aplicación GERBI-MAP se procede a realizar la interfaz.

INTERFAZ

El desarrollo de una interfaz amigable para el usuario, constituyó un reto, para lo cual se trabajó con algunos plugins: como JQuery para la interacción con los componentes HTML. Incluyendo además el plugin TagCanvas para crear una nube de etiquetas dinámica e interactiva.

Un problema serio fue la integración del plugin TagCanvas utilizado para la animación de las categorías de los centros asociados, por tal motivo se tuvo que modificar el código para poder acoplarlo, este plugin sirve bastante en el aplicativo, porque mejora la funcionalidad y apariencia.

Luego de tener toda la interfaz y la aplicación funcionando se presentaron los problemas porque al inicio no se definió el CMS a utilizar por tal razón la aplicación fue realizada creando una propia plantilla e ir modificando y adaptando código; por tal motivo, no fue posible realizar el cambio del aplicativo GERBI-MAP a Drupal un sistema de Gestión de Contenido Modular, por esta razón se creó un portal en drupal para todo el proyecto PUZZLE en donde está involucrada la aplicación GERBI-MAP.

Otro problema surgió en el servidor donde está subida la aplicación GERBI-MAP, al momento de abrir el link⁶⁹ del aplicativo no cargaba los archivos con extensión “.js” y “.CSS” mostrando la interfaz distorsionada, para corregir se tuvo que realizar varias pruebas de configuración del servidor y desactivar el modo de SELinux (*Seguridad*

⁶⁹ http://gerbi.utpl.edu.ec/gerbi_map/

Mejorada de Linux) con la siguiente línea de código: `[root]# /usr/sbin/setenforce 0`, mejorando completamente la aplicación.

También no mostraba el mapa del Ecuador por un error con el Key API de GoogleMaps, por tal razón, se volvió a generar el Key API de GoogleMaps en la versión 2.0 para acoplarlo nuevamente con el aplicativo.

Se debe mencionar que la construcción del aplicativo GERBI-MAP considera únicamente el almacenaje y consumo de la información en un ambiente semántico, más no temas como la actualización o eliminación de información del mismo.

En base a todo este proceso descrito a lo largo del proyecto, se puede considerar al mismo cómo la muestra de interacción entre componentes web existentes e información almacenada de forma semántica, brindando al usuario una interacción totalmente transparente.

Luego del trabajo dedicado a la implementación del aplicativo GERBI-MAP y solventando los inconvenientes antes mencionados se puede decir que el aporte que el presente trabajo deja, se basa en la demostración de la posibilidad de migrar y visualizar datos en tripletas RDF de manera transparente para el usuario, mencionando que para el proceso de migración de datos se desarrolló un aplicativo que permite leer archivos CSV y crear una ontología OWL; además, la forma de comunicación entre un almacén de tripletas (Virtuoso) y una aplicación web.

CONCLUSIONES

CONCLUSIONES

Al término del presente proyecto investigativo se puede concluir lo siguiente:

- El uso de tripletas RDF permite representar información de tipo bibliográfica y da la posibilidad de relacionar y nueva información.
- La forma de representación de conocimiento escogida depende del propósito de la aplicación, las estructuras semánticas mediante el uso de RDF permite construir repositorios de información accesibles y por ser un estándar propuesto por la W3C.
- La combinación del lenguaje de consulta SPARQL para la recuperación de información junto al formato de intercambio JSON, constituyen un mecanismo eficaz para interactuar directamente con un TripleStore.
- La integración de herramientas y tecnologías web disponibles como APIs, Ajax, JavaScript permitió el desarrollo del aplicativo más atractivo e interactivo al usuario.
- La recolección de estadísticas de acceso a cada uno de los recursos del aplicativo GERBI-MAP forma parte de información valiosa para futuros proyectos.
- La homogeneidad de configuraciones entre los ambientes de pruebas y el ambiente final de los servidores para el aplicativo permiten reducir fallas al momento de migrar el mismo.

RECOMENDACIONES

RECOMENDACIONES

En base a la investigación realizada y el estudio de los elementos que componen el presente proyecto se mencionan las siguientes recomendaciones:

- Para la selección de un lenguaje de consulta semántico, se debe considerar la aceptabilidad que posea en la web. Siendo de vital importancia acogerse a las recomendaciones hechas por estándares W3C.
- No toda la información disponible para un tema en específico puede ser representada de igual forma, por lo que la selección del mapa de conocimiento debe considerar tanto el tipo de información cómo lo que se quiere transmitir.
- Es recomendable hacer uso de estándares globales para la creación de servicios semánticos, de esta forma el servicio llega a ser interoperable y útil.
- Se recomienda no mostrar al usuario toda la información que se pueda, sino enfocar la extracción a contenido apropiado y orientado al fin de la aplicación, facilidad permitida por SPARQL.
- En relación a la geoLocalización de puntos, se recomienda la estandarización de coordenadas dentro de una misma plataforma, para impedir errores de conversión o precisión.
- Para realizar actualización de la información contenida en el Store, se recomienda hacer uso de SPARQL 1.1 o en su defecto realizar cambios en el archivo principal y efectuar nuevamente la carga del Store.

- Es importante recomendar el uso de ambientes y configuraciones similares al momento de migrar el aplicativo de un servidor a otro, para evitar contratiempos y mal funcionamiento del mismo.

TRABAJO FUTURO

TRABAJO FUTURO

Al término del presente trabajo sobre el aplicativo GERBI-MAP, se puede mostrar, la integración de diversas tecnologías, con el objeto de proporcionar la visualización de datos almacenados en un Store Semántico en una forma amigable.

Luego de todo el análisis realizado, se puede apreciar los procesos como: actualización/eliminación de los datos del Store quedan aún por cubrir y serían temas para profundizar en trabajos futuros, dando así una funcionalidad más dinámica y completa al Store y a la aplicación.

Aspectos relativos a la migración de los datos también son otros temas que se pueden ampliar, puesto que resultaría interesante determinar un aplicativo que sea capaz de extraer esta información de forma automática de los repositorios existentes y transformarla a tripletas, esto evitaría el tener que aplicar un pre-procesamiento manual a la información.

Es además importante mencionar que el aplicativo GERBI-MAP puede tener mayor impacto y usabilidad, al abrir la posibilidad de integrar componentes de información orientados a mostrar las preferencias tanto de usuarios como de los administradores al momento de subir la información. De esta forma se podría extraer patrones de conducta que permitirían determinar y proporcionar la información de interés del usuario en determinadas áreas.

Es así, que el presente proyecto abre la puerta para futuros proyectos de investigación muy amplios cuyos resultados contribuirán a pasar de una web de datos a una web de cosas cada vez más amigable e inteligente para el usuario.

GLOSARIO

GLOSARIO

- **Ajax:** Tecnología que permite realizar llamadas asíncronas hacia un servidor para extraer información.
- **API:** (Application Programming Interface) Es una interfaz de programación de aplicaciones, que permite la interacción entre componentes de software para una tarea específica.
- **Capa (Layer):** Espacio ubicado sobre el mapa, en el cual se pueden ubicar diferentes 'features' para poder visualizarlos.
- **EndPoint:** Nombre asignado a la dirección web del TripleStore donde se accede a los datos.
- **FancyBox:** Framework JavaScript para la presentación de PopUps dinámicos y con contenido personalizado.
- **Feature:** Elemento gráfico que se representa sobre un mapa, el cual puede ser manipulado para almacenar información.
- **GoogleMaps:** Servicio de Mapas geo-referenciados de Google, el mismo que se presta como un servicio gratuito, bajo ciertas restricciones.
- **Individuo:** Instanciación de una clase del vocabulario, cuyas propiedades tienen un valor específico.
- **IRI:** Identificador de recursos internacionalizados, a diferencia de URI puede contener caracteres Unicode/ISO 10646.
- **JSON:** Tecnología que utiliza un lenguaje estructurado para la representación de la información en el intercambio entre aplicaciones.
- **JQuery:** Framework JavaScript para manipulación de elementos DOM de HTML.
- **Linked Data:** Conocido también como datos enlazados, mantiene el principio de que la información debe ser enlazada para evitar la redundancia.

- **Ontología:** Formulación de información bajo un esquema conceptual de un determinado dominio, que facilita comunicación e intercambio de información.
- **OpenLayers:** Framework JavaScript para el manejo de mapas y objetos dentro de él.
- **OpenLinked Data (LOD):** Indica los procesos sobre los cuales se basa la publicación de información enlazada, para la construcción de la web de datos.
- **PopUp:** Ventana emergente lanzada por una aplicación, que muestra información relativa a una acción.
- **RDF:** Marco de Descripción de Recursos (Resource Description Framework), es un lenguaje descriptivo conceptual, diseñado para la representación de información en la web.
- **Semántica:** Representación de forma tal que la información adquiere significado.
- **SeRQL:** Lenguaje de recuperación de datos, diseñado específicamente para el trabajo con Sesame.
- **Sesame:** Almacén de tripletas mantenido por el grupo openRDF.org, el mismo que compete con otros sistemas propietarios.
- **Shapefile:** Archivo vectorial propio de aplicaciones geográficas como ArcGIS.
- **SPARQL:** Lenguaje de recuperación de datos, que se ha convertido en casi un estándar en la web, apoyado por la W3C. Actualmente se encuentra por su versión 1.1
- **Spherical Mercator:** Sistema de coordenadas utilizado en el servicio de mapas de Google para la representación de puntos sobre la superficie.
- **TripleStore (Store):** Almacén de tripletas, que utiliza lenguajes de consulta semánticos para la recuperación y actualización de la información.

- **Tripletas:** Estructura compuesta de tres partes, sujeto-predicado-objeto para representación de información en la web bajo un esquema RDF.
- **URI:** Identificador único de recurso, utilizado en la internet.
- **UTM:** Sistema de coordenadas geográficas bidimensional. (Universal Transverse Mercator).
- **Virtuoso:** Store de tripletas ampliamente utilizado por las facilidades que presta, como la interfaz de administración y la interacción con el EndPoint.
- **Vocabulario:** Estructura semántica que determina las clases, propiedades y relaciones que definirán los individuos.
- **Web semántica:** También conocida como la Web 3.0 determina la evolución de la red de datos a una red de cosas, de donde se pueda inferir conocimiento.
- **WGS84:** Sistema Geodésico Mundial 1984, permite localizar un punto sobre la superficie terrestre y que se ha convertido en un estándar.

REFERENCIAS BIBLIOGRÁFICAS

- [1] RC Chakraborty (2008), **Knowledge Representation - Issues, Predicate Logic, Rules**, Consultado (22/08/2011) en [<http://myreaders.files.wordpress.com/2008/04/ai-knowledge-representation-issues-predicate-logic-rules.pdf>].
- [2] Brachman & Levesque (2004), **Knowledge Representation and Reasoning**, Edit. El Sevier.
- [3] Dragan, G., Dragan, D., Vladan, D. & Bran S. (2006), **Model Driven Architecture and Ontology Development**, Edit. Springer.
- [4] Bonarini (2011), **Knowledge Representation Techniques**, Consultado (23/08/2011) en [<http://chrome.ws.dei.polimi.it/images/1/1d/2-KnowledgeTechniquesHandout.pdf>].
- [5] Haq, Shaiq (2011), **Knowledge Representation Techniques**, Consultado (24/08/2011) en [http://www.wecuw.edu.pk/downloads/ai/06_AI_Lecture_on_Knowledge_Representation_Techniques.pdf].
- [6] Clark, P & Porter, B (2004), **KM -The Knowledge Machine 2.0**, Consultado (26/08/2011) en [<http://www.cs.utexas.edu/users/mfkb/km/userman.pdf>].
- [7] Welty, C. A (1996), **An Integrated Representation for Software Development and Discovery**, Rensselaer Polytechnic Institute, New York.
- [8] Hellström, T., & Husted, K (2004), **Mapping knowledge and intellectual capital in academic environments-A focus group study**, Journal of Intellectual Capital, 5(1): Pág.165.
- [9] Davenport, T. H. & Prusak, L (2000), **Working Knowledge: How Organizations Manage What they Know**, Boston; Massachusetts: Harvard Business School Press, USA.
- [10] Dürsteler, Juan C (2004), **La revista digital de InfoVis.net**, Consultado (17/08/2011) en [<http://www.infovis.net/printMag.php?num=141&lang=1>].

- [11] Lasso Clemente, Iván (2008), **Proyecto Autodidactica**, Consultado (18/09/2011) en [\[http://www.proyectoautodidacta.com/comics/que-son-las-etiquetas-metadatos/\]](http://www.proyectoautodidacta.com/comics/que-son-las-etiquetas-metadatos/).
- [12] Rivero, S. (S.F.) (2006), **Mapas de conocimiento y repositorios de conocimiento estructural**, En CARBALLO, R. et. al. Innovación y gestión del conocimiento modelo, metodología, sistemas y herramientas de innovación. España: Díaz de Santos.
- [13] Justo L. & Araceli C (2009), **Investigación 3: Mapas de conocimiento, cartografía del conocimiento y uso de ontologías para su construcción**, Consultado (27/08/2011) en [\[http://negro.iing.mx.l.uabc.mx/~aracelijusto/investigaciones/investigacion3.pdf\]](http://negro.iing.mx.l.uabc.mx/~aracelijusto/investigaciones/investigacion3.pdf).
- [14] Davenport, T. & Prusak, L. Working Knowledge (1998), **How organizations manage what they know**, Boston; Massachusetts: Harvard Business School Press.
- [15] P. Hayes, T. Eskridge, M. Mehrotra, D. Bobrovnikoff, T. Reichherzer & R. Saavedra (2005), **COE: Tools for Ontology Development and Reuse, International Conference on Intelligence Analysis**, May 3-5, McLean, VA.
- [16] Lamas C. & María I (2006), **Proyecto: Lenguajes de consulta para Documentos RDF**, Consultado (10/10/2011) en [\[http://openaccess.uoc.edu/webapps/o2/bitstream/10609/640/1/38449tfc.pdf\]](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/640/1/38449tfc.pdf).
- [17] Pascal, Hitzler (2011), **Knowledge Representation for the Semantic Web**, Consultado (19/10/2011) en [\[http://www.semantic-web-book.org/w/images/9/94/W2011-03-RDF.pdf\]](http://www.semantic-web-book.org/w/images/9/94/W2011-03-RDF.pdf).
- [18] Medina, D. F (2003), **Universitat Oberta de Catalunya. Estudio del modelo de representación XML/RDF**, Consultado (10/10/2011) en [\[http://openaccess.uoc.edu/webapps/o2/bitstream/10609/1024/1/20847tfc.pdf\]](http://openaccess.uoc.edu/webapps/o2/bitstream/10609/1024/1/20847tfc.pdf).
- [19] Tim Berners-Lee, Dan Connolly & Sandro Hawke (2003), **Primer: Getting into RDF and the Semantic Web using N3**, Consultado (10/10/2011) en [\[http://www.w3.org/2000/10/swap/Primer.HTML\]](http://www.w3.org/2000/10/swap/Primer.HTML).

- [20] Tim Berners-Lee (1998), **Notation 3 Logic**, Consultado (10/10/2011) en [\[http://www.w3.org/DesignIssues/Notation3.HTML\]](http://www.w3.org/DesignIssues/Notation3.HTML).
- [21] Dave Beckett (2001), **N-Triples.W3C RDF Core WG Internal Working Draft**, Consultado (10/10/2011) en [\[http://www.w3.org/2001/sw/RDFCore/ntriples\]](http://www.w3.org/2001/sw/RDFCore/ntriples).
- [22] D. Beckett (2006), **SPARQL RDF Query Language Reference v1.8**, Consultado (08/02/2012) en [\[http://www.dajobe.org/2005/04-sparql/\]](http://www.dajobe.org/2005/04-sparql/).
- [23] Mailing list archives. Jena - Mail List: commit versión. Consultado (05/01/2012) en [\[http://mail-archives.apache.org/mod_mbox/incubator-jena-commits/201109.mbox/%3C20110905151105.3BD3A2388AC8@eris.apache.org%3E\]](http://mail-archives.apache.org/mod_mbox/incubator-jena-commits/201109.mbox/%3C20110905151105.3BD3A2388AC8@eris.apache.org%3E).

ANEXOS

Anexo A:

Instalación de Sesame

Sesame se encuentra alojado en OpenRDF.org, el cual es un proyecto de Aduna⁷⁰ el mismo que se ha encargado de mantenerlo y distribuirlo. En este caso descargaremos la última versión disponible desde el sitio:

<http://sourceforge.net/projects/sesame/files/Sesame%20/>

La misma que es **openRDF-sesame-2.6.2-sdk.tar.gz**. Es importante notar que se debe descargar el archivo que indica SDK puesto que dentro de este se encuentra un directorio bajo el nombre de war donde se encuentran dos archivos: *openRDF-sesame.war* y *openRDF-workbench.war* los mismos que serán utilizados posteriormente.

Es necesario además, instalar un contenedor Servlets Java, en este caso se hará uso de Apache Tomcat⁷¹, el mismo que cumple con los requisitos necesarios para el funcionamiento de Sesame.

Se utilizará la versión 7 que se encuentra bajo el nombre de:

apache-tomcat-7.0.23.tar.gz

Descomprimir este archivo en el directorio `/usr/local/` bajo el nombre de `tomcat7.0.23`. Para ello utilizamos la instrucción:

```
tar -xzvf apache-tomcat-7.0.23.tar.gz tomcat7.0.23
```

Una vez extraído, modificamos el archivo `tomcat-users.XML` que se encuentra en el directorio `conf`. Añadiendo las siguientes líneas:

⁷⁰ <http://www.aduna-software.com/>

⁷¹ <http://tomcat.apache.org/>

```

<role rolename="manager-gui"/>
<role rolename="admin-script"/>
<user username="admin" password="admin" roles="manager-gui"/>
<user username="admin" password="admin" roles="manager-gui"/>

```

De esta forma creamos un usuario **'admin'** que nos permitirá gestionar Tomcat desde su interfaz gráfica. Para levantar el servicio ejecutamos el archivo **startup.sh** ubicado en la carpeta **bin**. Mediante la instrucción:

```
sh startup.sh
```

De esta forma podremos acceder a la interfaz de administración desde la dirección:

<http://localhost:8080/manager/HTML>

Luego de colocar el usuario y clave creados anteriormente podremos observar los aplicativos existentes así como montar nuevos. Tan solo es necesario utilizar la función **'WAR file to deploy'**, seleccionamos los archivos de Sesame descargados anteriormente y elegimos la opción **'Deploy'**. Una vez desplegados se observarán como se muestra en la Figura 2.

Figura 1. Subir un archivo WAR. Tomcat

/openrdf-sesame	<i>None specified</i>	OpenRDF Sesame	true
/openrdf-workbench	<i>None specified</i>		true

Figura 2. Aplicaciones Desplegadas

Al acceder a la dirección <http://localhost:8080/openRDF-workbench> desde un navegador observaremos la interfaz principal que se muestra en la Figura 3. Donde

se pueden observar la lista de repositorios (stores) actuales y sobre los que se puede operar.

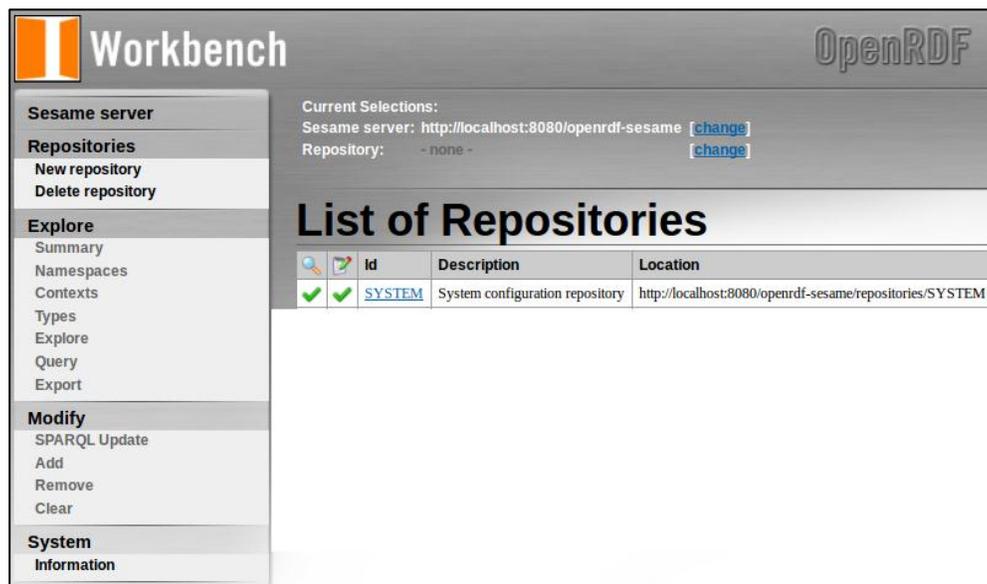


Figura 3. Interfaz Principal SeSame

Carga de Archivo

La acción de cargar el archivo RDF creado en el apartado anterior, se lo realiza con la opción 'New Repository' de la Interfaz principal de SeSame. Donde se debe colocar el ID con el cual se identificará al repositorio y que se utilizará en la extracción de información con SeRQL. En este caso nuestro repositorio se tendrá como identificador el valor de 'datosTest'.



Figura 4. Crear nuevo repositorio

Una vez creado nuestro Store (Repositorio) procedemos a cargar el archivo RDF, creado previamente. Desde la opción **Add** del menú **Modify** de la interfaz Principal. Utilizamos la función RDF Data File y localizamos el archivo RDF. Una vez localizado la opción **Upload** permitirá poblar el Store con estos datos.



Figura 5. Crear nuevo repositorio (Add RDF)

Para la realización de las consultas mediante SeRQL se utilizará una interfaz propia de Sesame que se puede encontrar en el menú **Explore**, denominada como **Query**. La misma que presenta una vista similar a la de la Figura 6. Donde se colocará la consulta en formato SeRQL y se mostrarán los resultados.

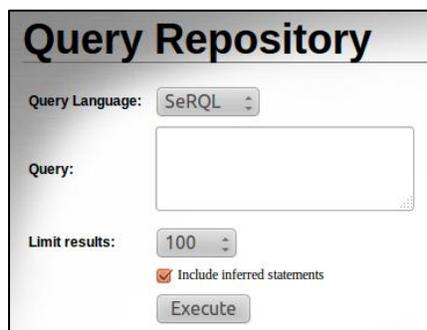


Figura 6. Espacio de Consulta – Sesame

Anexo B:

Instalación de Virtuoso OpenLink

En la presente instalación se describe el proceso realizado haciendo uso de los paquetes de instalación que se encuentran en los repositorios de la distribución de Linux Ubuntu. En esta distribución OpenLink Software⁷², mantiene Virtuoso OpenSource dentro de los repositorios oficiales, puesto que el servidor no contendrá ninguna configuración especial este método de instalación es adecuado, de lo contrario es recomendable la instalación mediante el uso del código fuente⁷³. Los pasos a seguir son los siguientes:

Actualizar los orígenes de datos

```
$ sudo apt-getupdate
```

Visualizar los paquetes disponibles, en caso de necesitar paquetes adicionales

```
$ apt-cache search '^virtuoso'
```

Descargar e Instalar

```
$ sudo apt-get install virtuoso-opensource
```

Durante este proceso de instalación se notificará de la creación de dos usuarios con privilegios de administrador en el servidor: “dba” y “dav” a los cuales se debe asignar una clave. En este caso se ha colocado los valores:

```
dbapasswd: `dba`
```

```
davpasswd: `dav`
```

⁷² <http://www.openlinksw.com/>

⁷³ <http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VOSUbuntuNotes#Building>

Estos usuarios permitirán el ingreso mediante la interfaz de administración, la cual puede ser ubicada en la siguiente dirección:

<http://gerbi.utpl.edu.ec:8890/>



Figura 7. Interfaz Principal Virtuoso

El ingreso a la interfaz de administración proporcionada por Virtuoso bajo el nombre de 'CONDUCTOR' puede ser realizado mediante la URL:

<http://gerbi.utpl.edu.ec:8890/conductor/>

Esta permite tener un control total del servidor, desde donde se pueden gestionar usuarios, servicios, paquetes, grafos entre otros.

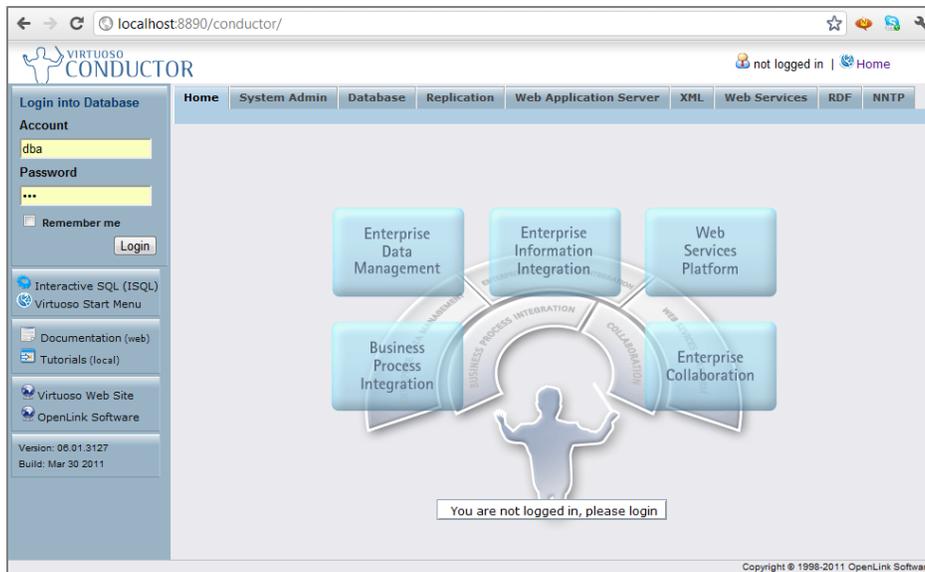


Figura 8. Interfaz de Administración – CONDUCTOR

El proceso de iniciar, detener u obtener información sobre el proceso que ejecuta el servidor puede ser llevado a cabo mediante la instrucción:

```
$ sudo /etc/init.d/virtuoso-opensource-6.1 [ start | stop | status]
```

En ocasiones, cuando el servidor ha sido detenido de forma incorrecta puede presentar problemas para iniciar nuevamente de forma automática, en estos casos, es recomendable iniciar el servicio mostrando la información de log. Para ello ejecutamos el siguiente comando, dentro del directorio *database*, el mismo que se encuentra dentro de la instalación de Virtuoso:

```
$ virtuoso-t -df
```

Este mostrará información útil sobre el problema, uno de los más comunes es causado por el archivo **virtuoso.lck**, el mismo que es creado en cada inicio del servidor, pero si no ha sido borrado previamente por el servidor la instancia no se levantará nuevamente. Si este fuera el problema, basta con eliminar el archivo **virtuoso.lck** del directorio *database* de forma manual, para que el servicio vuelva a iniciar.

Carga de Archivo

El proceso a seguir para la carga de datos desde un archivo de tripletas al Store, es realizado mediante la interfaz principal CONDUCTOR, mediante los siguientes pasos:

1. Bajo la pestaña RDF seleccionar la opción **RDF StoreUpload**, la cual nos permite efectuar la carga de datos desde un archivo, en este caso será el que se creó con la aplicación anteriormente.
2. Una vez seleccionado el archivo y definido el nombre del grafo, completar haciendo clic en el botón **“UPLOAD”**. El nombre del grafo definido será:

<http://gerbi.utpl.edu.ec:8890/dataGerbiMap>

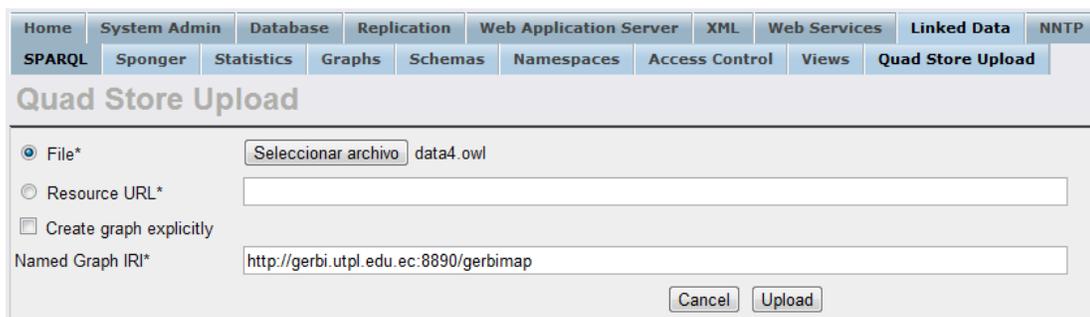


Figura 9. Carga de Archivo OWL

3. Una vez subidos los datos podremos comprobar la creación del grafo en la opción **Graphs**.



Figura 10. Grafo Creado

La extracción de la información será realizada mediante el uso de un **EndPoint** propio de virtuoso, instalación de virtuoso que se localiza en la dirección:

<http://gerbi.utpl.edu.ec:8890/SPARQL>

Anexo C:

Instalación de Apache - PHP- MySQL

A continuación se describe el proceso de instalación de estas tres herramientas en un entorno Linux (Ubuntu) mediante el uso de los repositorios que vienen con esta distribución.

Primero instalamos el servidor web Apache mediante el comando

```
$ sudo apt-get install apache
```

Esto instalará y levantará el servicio automáticamente, por defecto se puede acceder al servicio desde la dirección:

```
http://localhost/
```

Para que el servidor web reconozca un sitio es necesario mover todo el contenido a la carpeta

```
/var/www/
```

Para probar que el servicio esté ejecutándose se puede utilizar el siguiente comando:

```
$ service apache2 status
```

La instalación de PHP necesita ser realizada con el siguiente comando:

```
$ sudo apt-get install php5
```

De esta forma se realiza la integración del servidor web apache con php.

La base de datos MySQL es instalada de la siguiente forma:

```
$ sudo apt-get install MySQL-server  
$ sudo apt-get install php5-MYSQL
```

De esta forma se tiene instalado el servidor de base de datos relacional. Para realizar la carga de la estructura relacional, la cual se encuentra en un archivo, se lo realiza de la siguiente forma:

```
$ MySQL -u root -p <archivo.sql
```

Al final de todos estos pasos tendremos un entorno totalmente funcional para la aplicación.

Anexo D:

Migración de Datos

```
package principal;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Principal
{
    private String dirVocab = "http://www.utpl.edu.ec/gerbimap/rdf/dataTest.owl";
    public static int numHorario = 0;
    public static int numLibro = 0;
    public static int numGuia = 0;
    public static int numMult = 0;

    public static void main(String[] args)
    {
        new Principal().cargarLibros();
        //new Principal().cargarTesis();
        //new Principal().cargarGuias();
        //new Principal().cargarMultimedia();
        //new Principal().cargarCentros();
    }

    public void cargarLibros()
    {
        try {
            FileInputStream fis = new FileInputStream(new File("c:/infoRDF/libros.csv"));
            InputStreamReader is = new InputStreamReader(fis, "ISO-8859-1");//ISO-8859-1

            BufferedReader in = new BufferedReader(is);
            String str;
            while ((str = in.readLine()) != null) {
                //System.out.println(str);
                str = str.replace("&", "y");
                String[] parts = str.split(";");
                // 0 1 2 3 4 5 6 7 8 9
                //CodIndividuo;Area;AUTOR;Titulo;Subtitulo;isbn;EDICION;Editorial;Año;keyWord
                //generar el archivo en base a la estructura
```

```

String individuo = "<owl:NamedIndividual rdf:about=\"" + dirVocab + "#" + parts[0] +
    "\"> \n"
    + "<rdf:type rdf:resource=\"" + dirVocab + "#LIBRO\"/> \n"
    + "<anio rdf:datatype=\""&xsd:int\">" + getAnio() + "</anio> \n"
    + "<edicion rdf:datatype=\""&xsd:int\">" + parts[6] + "</edicion> \n"
    + "<autor rdf:datatype=\""&xsd:string\">" + parts[2] + "</autor> \n"
    + "<isbn rdf:datatype=\""&xsd:string\">" + parts[5] + "</isbn> \n"
    + "<area rdf:datatype=\""&xsd:string\">" + parts[1] + "</area> \n"
    + "<editorial rdf:datatype=\""&xsd:string\">" + parts[7] + "</editorial> \n"
    + "<keyWord rdf:datatype=\""&xsd:string\">" + parts[9] + "</keyWord> \n"
    + "<subTitulo rdf:datatype=\""&xsd:string\">" + parts[4] + "</subTitulo> \n"
    + "<titulo rdf:datatype=\""&xsd:string\">" + parts[3] + "</titulo> \n"
    + "</owl:NamedIndividual> \n";

    System.out.println(individuo);
    System.out.println("");
}
} catch (IOException ex) {
    Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);
}
}
}

```

```

public void cargarTesis() {
try {
    FileInputStream fis = new FileInputStream(new File("c:/infoRDF/tesis.csv"));
    InputStreamReader is = new InputStreamReader(fis, "ISO-8859-1");//ISO-8859-1
    BufferedReader in = new BufferedReader(is);
    String str;
    while ((str = in.readLine()) != null) {
        //System.out.println(str);
        str = str.replace("&", "y");
        str = str.replace("\\"", "");
        String[] parts = str.split(";");

        //0 1 2 3 4 5
        //ID;Titulo;Actor;Palabras clave;Fecha de publicación;Resumen;
        // 6 7 8
        //URI;Aparece en las colecciones;enlace;
        String individuo = "<owl:NamedIndividual rdf:about=\"" + dirVocab + "#" +
            parts[0] + "\"> \n"
            + "<rdf:type rdf:resource=\"" + dirVocab + "#TESIS\"/>"
            + "<fechaP rdf:datatype=\""&xsd:date\">" + parts[4] + "</fechaP> \n";
        String[] autores = parts[2].split("_");
        for (int i = 0; i < autores.length; i++) {
            individuo += " <autor rdf:datatype=\""&xsd:string\">" + autores[i] + "</autor>
                \n";
        }
    }
}
}

```

```

individuo += " <enlace rdf:datatype=\ "&xsd:string\">" + parts[8] + "</enlace>
  \n"
  + "<coleccion rdf:datatype=\ "&xsd:string\">" + parts[7] + "</coleccion> \n";
String[] keyWords = parts[3].split(",");
for (int i = 0; i < keyWords.length; i++) {
  individuo += " <keyWord rdf:datatype=\ "&xsd:string\">" + keyWords[i] +
    "</keyWord> \n";
}

individuo += " <resumen rdf:datatype=\ "&xsd:string\">" + parts[5] +
  "</resumen> \n"
  + "<titulo rdf:datatype=\ "&xsd:string\">" + parts[1] + "</titulo> \n"
  + "<uri rdf:datatype=\ "&xsd:string\">" + parts[6] + "</uri> \n"
  + "</owl:NamedIndividual> \n";
System.out.println(individuo);
System.out.println("");
}
} catch (IOException ex) {
  Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);
}
}

public void cargarGuias()
{
  try {
    FileInputStream fis = new FileInputStream(new File("c:/infoRDF/guias.csv"));
    InputStreamReader is = new InputStreamReader(fis, "ISO-8859-1");//ISO-8859-1
    BufferedReader in = new BufferedReader(is);
    String str;
    while ((str = in.readLine()) != null) {
      //System.out.println(str);
      str = str.replace("&", "y");
      String[] parts = str.split(",");

      // 0 1 2 3 4 5 6 7
      //codIndividuo;area;AUTOR;Titulo;Subtitulo;isbn;EDICION;Editorial
      //generar el archivo en base a la estructura
      String individuo = " <owl:NamedIndividual rdf:about=\ "" + dirVocab + "#" +
        parts[0] + "\"> \n"
        + "<rdf:type rdf:resource=\ "" + dirVocab + "#GUIA\"/> \n"
        + "<edicion rdf:datatype=\ "&xsd:string\">" + parts[6] + "</edicion> \n"
        + "<editorial rdf:datatype=\ "&xsd:string\">" + parts[7] + "</editorial> \n"
        + "<titulo rdf:datatype=\ "&xsd:string\">" + parts[3] + "</titulo> \n"
        + "<area rdf:datatype=\ "&xsd:string\">" + parts[1] + "</area> \n"
        + "<isbn rdf:datatype=\ "&xsd:string\">" + parts[5] + "</isbn> \n"
        + "<subTitulo rdf:datatype=\ "&xsd:string\">" + parts[4] + "</subTitulo> \n"
        + "<autor rdf:datatype=\ "&xsd:string\">" + parts[2] + "</autor> \n"

```

```

        + " </owl:NamedIndividual> \n";
        System.out.println(individuo);
        System.out.println("");
    }
} catch (IOException ex) {
    Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);
}
}

public void cargarMultimedia()
{
    try {
        FileInputStream fis = new FileInputStream(new
            File("c:/infoRDF/multimedia.csv"));
        InputStreamReader is = new InputStreamReader(fis, "ISO-8859-1");//ISO-8859-1
        BufferedReader in = new BufferedReader(is);
        String str;

        while ((str = in.readLine()) != null) {
            //System.out.println(str);
            str = str.replace("&", "y");
            String[] parts = str.split(";");

            // 0    1  2  3    4    5  6    7
            //codInvidivuo;area;AUTOR;Titulo;Subtitulo;isbn;EDICION;Editorial

            //generar el archivo en base a la estructura
            String individuo = " <owl:NamedIndividual rdf:about=\"\" + dirVocab + "#" +
                parts[0] + "\"> \n"
                + "<rdf:type rdf:resource=\"\" + dirVocab + "#MULTIMEDIA\"/> \n"
                + "<edicion rdf:datatype=\"&xsd:int\">" + parts[6] + "</edicion> \n"
                + "<isbn rdf:datatype=\"&xsd:string\">" + parts[5] + "</isbn> \n"
                + "<area rdf:datatype=\"&xsd:string\">" + parts[1] + "</area> \n"
                + "<editorial rdf:datatype=\"&xsd:string\">" + parts[7] + "</editorial> \n"
                + "<autor rdf:datatype=\"&xsd:string\">" + parts[2] + "</autor> \n"
                + "<subTitulo rdf:datatype=\"&xsd:string\">" + parts[4] + "</subTitulo> \n"
                + "<titulo rdf:datatype=\"&xsd:string\">" + parts[3] + "</titulo> \n"
                + " </owl:NamedIndividual> \n";

            System.out.println(individuo);
            System.out.println("");
        }
    } catch (IOException ex) {
        Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);
    }
}
}

```

```

public void cargarCentros()
{
try {
FileInputStream fis = new FileInputStream(new File("c:/infoRDF/centros.csv"));
InputStreamReader is = new InputStreamReader(fis, "ISO-8859-1");//ISO-8859-1
BufferedReader in = new BufferedReader(is);
String str;
while ((str = in.readLine()) != null) {
//System.out.println(str);
str = str.replace("&", "y");
String[] parts = str.split(";");
// 0 1 2 3 4 5 6 7
//NUM;LATITUD;LONGITUD;Categoria;Población;Cantón;Provincia;Coordinador;
// 8 9 10 11 12 13 14 15
//Telefono;Fax;Dirección;email;Blog;LugarEvaluación;Observaciones;Horario
String individuo = " <owl:NamedIndividual rdf:about=\"\" + dirVocab + "#centro"
+ parts[0] + "\"> \n"
+ "<rdf:type rdf:resource=\"\" + dirVocab + "#CENTRO_ASOCIADO\"/> \n"
+ "<lat rdf:datatype=\"&xsd;double\">" + parts[1] + "</lat> \n"
+ "<lon rdf:datatype=\"&xsd;double\">" + parts[2] + "</lon> \n"
+ "<direccion rdf:datatype=\"&xsd:string\">" + parts[10] + "</direccion> \n"
+ "<telefono rdf:datatype=\"&xsd:string\">" + parts[8] + "</telefono> \n"
+ "<tipoCentro rdf:datatype=\"&xsd:string\">" + parts[3] + "</tipoCentro> \n"
+ "<canton rdf:datatype=\"&xsd:string\">" + parts[5] + "</canton> \n"
+ "<lugEvaluacion rdf:datatype=\"&xsd:string\">Esc. " + parts[0] + "
</lugEvaluacion> \n"
+ "<observacion rdf:datatype=\"&xsd:string\">No
registradas</observacion> \n"
+ "<poblacion rdf:datatype=\"&xsd:string\">" + parts[4] + "</poblacion> \n"
+ "<provincia rdf:datatype=\"&xsd:string\">" + parts[6] + "</provincia> \n"
+ "<email rdf:datatype=\"&xsd:string\">centro" + parts[0] +
"@utpl.edu.ec</email> \n"
+ "<coordinador rdf:datatype=\"&xsd:string\">" + parts[7] +
"</coordinador> \n"
+ "<fax rdf:datatype=\"&xsd:string\">" + parts[9] + "</fax> \n";
ArrayList<String> horariosThis = cargarHorarios(Integer.parseInt(parts[0]));
for (String string : horariosThis) {
individuo += "<horario rdf:resource=\"\" + dirVocab + "#" + string + "\"/> \n";
}
ArrayList<String> librosThis = cargarRecursoLibros(Integer.parseInt(parts[0]));
for (String string : librosThis) {
individuo += "<bibliotecaIndv rdf:resource=\"\" + dirVocab + "#" + string +
\"\"/> \n";
}
ArrayList<String> guiasThis = cargarRecursoGuias(Integer.parseInt(parts[0]));
for (String string : guiasThis) {

```

```

        individuo += " <bibliotecaIndv rdf:resource=\"" + dirVocab + "#" + string +
            "\"/> \n";
    }
    ArrayList<String> multThis = cargarRecursoMult(Integer.parseInt(parts[0]));
    for (String string : multThis) {
        individuo += " <bibliotecaIndv rdf:resource=\"" + dirVocab + "#" + string +
            "\"/> \n";
    }

    //loja 1 al 15
    if (parts[0].equals("7")) {
        for (int i = 1; i <= 15; i++) {
            individuo += " <bibliotecaIndv rdf:resource=\"" + dirVocab + "#tesis" +
                complete4Digits(String.valueOf(i), 4) + "\"/> \n";
        }
    } else if (parts[0].equals("5")) { //cariamanga 16 al 30
        for (int i = 16; i <= 30; i++) {
            individuo += " <bibliotecaIndv rdf:resource=\"" + dirVocab + "#tesis" +
                complete4Digits(String.valueOf(i), 4) + "\"/> \n";
        }
    }
    }
    for (String string : multThis) {
        individuo += " <bibliotecaIndv rdf:resource=\"" + dirVocab + "#" + string +
            "\"/> \n";
    }
    individuo += " </owl:NamedIndividual> ";
    System.out.println(individuo);
    System.out.println("");
}

} catch (IOException ex) {
    Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);
}
}

public ArrayList<String> cargarHorarios(int centro)
{
    try {
        FileInputStream fis = new FileInputStream(new File("c:/infoRDF/horarios.csv"));
        InputStreamReader is = new InputStreamReader(fis, "ISO-8859-1");//ISO-8859-1
        BufferedReader in = new BufferedReader(is);
        String str;
        ArrayList<String> horariosT = new ArrayList<String>();
        String todosHorarios = "";
        while ((str = in.readLine()) != null) {
            //System.out.println(str);
            str = str.replace("&", "y");
            String[] parts = str.split(";");

```

```

        //0      1      2
        //1;LUNES;08H30 - 12H30;
        if (parts[0].equals(String.valueOf(centro))) {
            numHorario++;
            String idHorario = "horario" + numHorario;
            horariosT.add(idHorario);
            //procesar hora
            String[] horas = parts[2].replace("H", ":").split("-");
            String individuo = " <owl:NamedIndividual rdf:about=\"" + dirVocab + "#" +
                idHorario + "\"> \n"
                + "<rdf:type rdf:resource=\"" + dirVocab + "#HORARIO\"/> \n"
                + "<horaI rdf:datatype=\"" + "&xsd;time\">" + horas[0].trim() + "</horaI> \n"
                + "<horaF rdf:datatype=\"" + "&xsd;time\">" + horas[1].trim() + "</horaF> \n"
                + "<dia rdf:datatype=\"" + "&xsd:string\">" + parts[1] + "</dia> \n"
                + " </owl:NamedIndividual>\n";
            todosHorarios += individuo;
        }
    }
    System.out.println("\n\n\n" + todosHorarios + "\n\n\n");
    return horariosT;
} catch (IOException ex) {
    Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);
}
return null;
}

```

```

public ArrayList<String> cargarRecursoLibros(int centro) {
    try {
        FileInputStream fis = new FileInputStream(new
            File("c:/infoRDF/librosxcentro.csv"));
        InputStreamReader is = new InputStreamReader(fis, "ISO-8859-1");//ISO-8859-1
        BufferedReader in = new BufferedReader(is);
        String str;
        ArrayList<String> librosR = new ArrayList<String>();
        while ((str = in.readLine()) != null) {
            str = str.replace("&", "y");
            String[] parts = str.split(";");
            if (parts[0].equals(String.valueOf(centro)) && parts.length > 1) {
                //cantidad de libros = parts[1]
                int maxLibros = Integer.parseInt(parts[1]);
                for (int i = 0; i < maxLibros; i++) {
                    numLibro++;
                    librosR.add("libro" + complete4Digits(String.valueOf(numLibro), 4));
                }
            }
        }
    }
    return librosR;
}

```

```

    } catch (IOException ex) {
        Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);
    }
    return null;
}

public ArrayList<String> cargarRecursoGuias(int centro) {
try {
    FileInputStream fis = new FileInputStream(new
        File("c:/infoRDF/guiasxcentro.csv"));
    InputStreamReader is = new InputStreamReader(fis, "ISO-8859-1");//ISO-8859-1
    BufferedReader in = new BufferedReader(is);
    String str;
    ArrayList<String> guiaR = new ArrayList<String>();
    while ((str = in.readLine()) != null) {
        str = str.replace("&", "y");
        String[] parts = str.split(";");
        if (parts[0].equals(String.valueOf(centro)) && parts.length > 1) {
            int maxGuias = Integer.parseInt(parts[1]);
            for (int i = 0; i < maxGuias; i++) {
                numGuia++;
                guiaR.add("guia" + complete4Digits(String.valueOf(numGuia), 3));
            }
        }
    }
    return guiaR;
} catch (IOException ex) {
    Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);
}
return null;
}

```

```

public ArrayList<String> cargarRecursoMult(int centro) {
try {
    FileInputStream fis = new FileInputStream(new
        File("c:/infoRDF/multimedioxcentro.csv"));
    InputStreamReader is = new InputStreamReader(fis, "ISO-8859-1");//ISO-8859-1
    BufferedReader in = new BufferedReader(is);
    String str;
    ArrayList<String> multiR = new ArrayList<String>();
    while ((str = in.readLine()) != null) {
        str = str.replace("&", "y");
        String[] parts = str.split(";");
        if (parts[0].equals(String.valueOf(centro)) && parts.length > 1) {
            multiR.add("mult" + complete4Digits(parts[1], 3));
            int maxMult = Integer.parseInt(parts[1]);
            for (int i = 0; i < maxMult; i++) {

```

```

        numMult++;
        multiR.add("mult" + complete4Digits(String.valueOf(numMult), 3));
    }
}
return multiR;
} catch (IOException ex) {
    Logger.getLogger(Principal.class.getName()).log(Level.SEVERE, null, ex);
}
return null;
}
public String complete4Digits(String num, int max) {
    int limit = max - num.length();
    for (int i = 0; i < limit; i++) {
        num = "0" + num;
    }
    return num;
}

public int getAnio() {
    return 1900 + (int) ((Math.random() * 99) + 1);
}
}

```

Anexo E:

Listado de Centros Seleccionados

Categoría	Población
Asociado	ALAMOR
Asociado	CATACUCHA
Asociado	CÉLICA
Asociado	MACARÁ
Asociado	SARAGURO
Provincial	MANTA
Provincial	PORTOVIEJO
Provincial	MACHALA
Provincial	CARIAMANGA
Provincial	QUEVEDO
Provincial	SANTO DOMINGO DE LOS COLORADOS
Provincial	CUENCA
Provincial	GUARANDA
Provincial	ZAMORA
Provincial	RIOBAMBA
Provincial	AMBATO
Provincial	LATACUNGA
Provincial	MACAS
Provincial	PUYO
Provincial	TENA
Provincial	ESMERALDAS
Provincial	FRANCISCO DE ORELLANA(EL COCA)
Provincial	IBARRA
Provincial	NUEVA LOJA (LAGO AGRIO)

Provincial	SANTA CRUZ
Provincial	TULCÁN
Provincial_SEDE	LOJA
Regional	GUAYAQUIL
Regional	QUITO

Anexo F:
Simbología

Símbolo	Leyenda
	Centro Asociado
	Centro Provincial
	Centro Regional
	Sede UTPL

Figura 11. Simbología

Anexo G: Interfaz de Aplicativo

1. Vista General

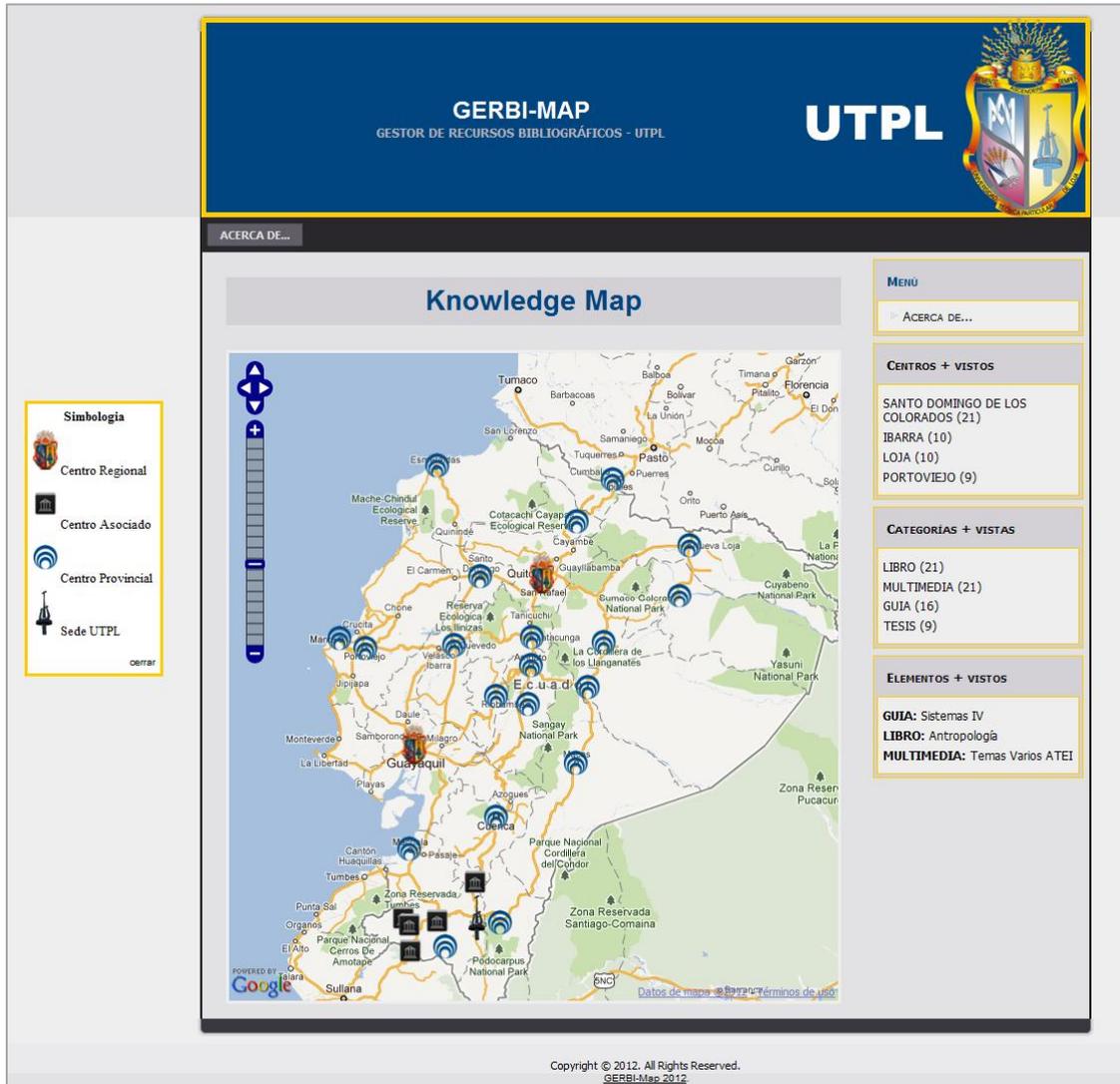


Figura 12. Interfaz Aplicativo: Vista General

2. Simbología



Figura 13. Interfaz Aplicativo: Simbología

3. Registro de Estadísticas



Figura 14. Interfaz Aplicativo: Estadísticas

4. PopUp



Figura 15. Interfaz Aplicativo: PopUp

5. Mapa de Etiquetas



Figura 16. Interfaz Aplicativo: Mapa de Etiquetas

6. Lista de Items



Figura 17. Interfaz Aplicativo: Listado de Items

7. Información de Individuo - PopUp

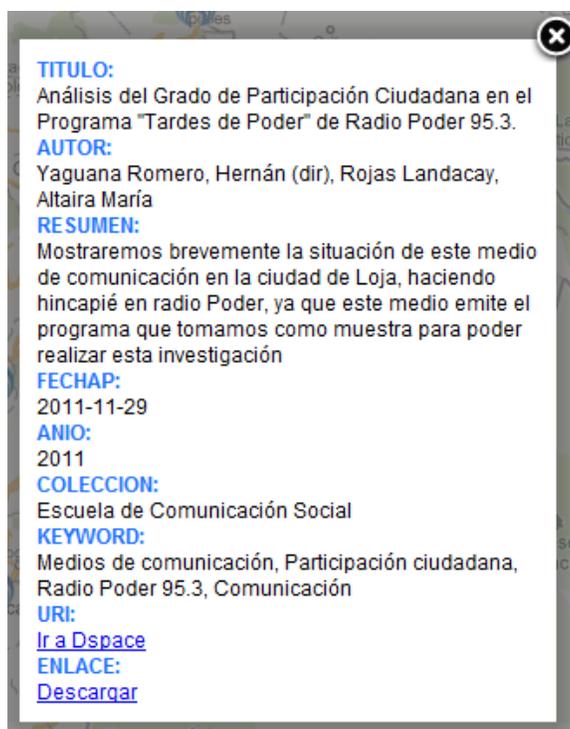


Figura 18. Interfaz Aplicativo: Información del Individuo

8. Acerca de...

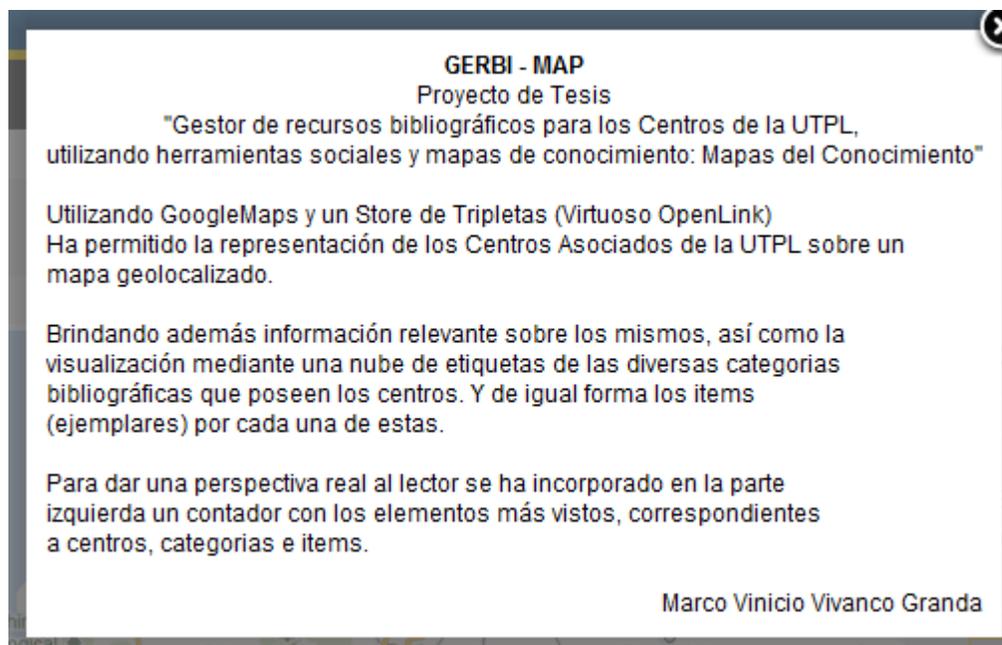


Figura 19. Acerca de