

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMATICA



TESIS DE GRADO

**“HERRAMIENTA VISUAL DE EDICIÓN DE ONTOLOGÍAS PARA
UNA WEB SEMÁNTICA”**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA MENCIÓN:
INGENIERIA DE SISTEMAS INFORMÁTICOS

POSTULANTE : ERIKA MAYRA KENTA CHIPANA

TUTOR : LIC. LUISA VELÁSQUEZ LÓPEZ MSC.

REVISOR : LIC. JORGE HUMBERTO TERÁN POMIER MSC.

LA PAZ - BOLIVIA

Dedicatoria

A mi mamá y papá.

Agradecimientos

A la Licenciada Luisa Velásquez por la colaboración en la estructuración del presente trabajo, en su organización y énfasis colaborativo.

Al Licenciado Jorge Terán, a quien considero es una persona impulsora de la investigación científica, cuyos conocimientos y amplia experiencia me proporcionaron esa orientación de carácter fundamental para todo el desarrollo del proyecto.

Agradecer a mis papás, quienes me dieron toda su comprensión, su apoyo y me inculcaron los valores necesarios para seguir el camino de excelencia.

A toda mi familia, mis hermanos, por su comprensión, por su inmenso cariño.

A mis tías, quienes estuvieron muy cerca para alentarme en los momentos de mayor dificultad a lo largo de todo el proceso.

A Paolito, por su apoyo a lo largo de todo este periodo de formación.

RESUMEN

En la labor de elevar el nivel de interoperabilidad entre estaciones de trabajo a nivel de la red de Internet, la Web, nace la propuesta de la Web Semántica con una visión más ambiciosa en cuanto a la comprensión Semántica, también llamada a nivel conceptos.

De esta forma nace la Ontología, definición que formaliza la reunión de toda esta gama de componentes tales como conceptos, atributos y relaciones, los cuales serán el sustento de nuestra Web Semántica. Así también se presentan formas de representar esta ontología, donde claramente sobresale el propuesto por la organización W3C, denominado lenguaje OWL.

Es en este contexto que la implementación de Sitios Web Semánticos, se convierte en una labor desde varios puntos de vista engorrosa, especialmente para los diseñadores cuyos objetivos se miden función del tiempo y la calidad que ellos desempeñen en esta labor.

La herramienta propuesta, opta por ofrecer la virtud de la representación gráfica de las ontologías como principal medio para subsanar el proceso de diseño, edición de las Ontologías, todo a raíz de un minucioso estudio de las capacidades de técnicas de visualización de datos.

La misma herramienta cuenta con capacidades que permiten la labor de edición aún más interactiva, gracias a las mismas recomendaciones de los mismos métodos de graficación, como el zoom, el dispersamiento de datos, filtrado.

Entonces la Herramienta de Edición de Ontologías, brinda la realización grafica de un esquema conceptual en primera instancia propuesta por RDF, y lo transforma directamente a un archivo OWL, con extensión owl, el cual esta listo, lo cual quiere decir validado, para su inserción directa al Sitio Web Semántico.

INDICE DE LA TESIS

RESUMEN.....	4
INDICE DE LA TESIS	5
LISTA DE GRAFICAS.....	9
LISTA DE TABLAS.....	10
1 PRESENTACIÓN.....	11
1.1 INTRODUCCIÓN.....	11
1.2 ANTECEDENTES.....	13
1.3 SITUACIÓN PROBLEMÁTICA.....	14
1.4 FORMULACIÓN DEL PROBLEMA.....	14
1.4.1 DEFINICIÓN DEL PROBLEMA.....	15
1.4.2 IDENTIFICACIÓN DE VARIABLES.....	15
1.4.3 OPERACIÓN DE VARIABLES.....	16
1.5 OBJETO DE ESTUDIO.....	16
1.6 JUSTIFICACIÓN.....	16
1.7 HIPÓTESIS.....	17
1.8 OBJETIVOS.....	17
1.8.1 OBJETIVO GENERAL.....	17
1.8.2 OBJETIVOS ESPECIFICOS.....	17
1.9 LÍMITES Y ALCANCES.....	18
1.9.1 LIMITES.....	18
1.9.2 ALCANCES.....	18
1.10 METODOLOGÍA.....	18
1.10.1 MÉTODOS Y MEDIOS DE INVESTIGACIÓN CIENTÍFICA.....	18
1.10.2 MÉTODOS Y MEDIOS DE INVESTIGACIÓN INFORMÁTICA.....	18
1.11 APORTES.....	19
1.11.1 APORTE TEÓRICO.....	19
1.11.2 APORTE PRÁCTICO.....	19
2 MARCO DE REFERENCIA.....	20
2.1 WEB ACTUAL.....	20
2.1.1 PROBLEMAS DE LA WEB ACTUAL.....	21
2.1.1.1 INTEROPERABILIDAD.....	21
Interoperabilidad técnica.....	21

Interoperabilidad sintáctica.....	21
Interoperabilidad semántica.....	21
2.2 WEB SEMÁNTICA.....	22
2.2.1 DEFINICIÓN: WEB SEMÁNTICA.....	22
2.2.2 IMPORTANCIA DE LA WEB SEMÁNTICA.....	22
2.2.3 CONCEPTOS DE LA WEB SEMÁNTICA.....	24
2.2.3.1 LOS 7 NIVELES DE LA WEB SEMÁNTICA.....	24
2.2.4 EJEMPLOS DE USO DE LA WEB SEMÁNTICA.....	25
2.3 ONTOLOGÍAS.....	27
2.3.1 ESTRUCTURA DE UNA ONTOLOGÍA.....	27
2.3.2 TIPOS DE ONTOLOGÍAS.....	28
2.4 TECNOLOGÍAS PARA LA WEB SEMÁNTICA.....	29
2.4.1 RDF (Resource Description Framework, marco de descripción de recursos).....	29
2.5 LENGUAJES DE REPRESENTACIÓN DE ONTOLOGÍAS.-.....	30
2.5.1 DAM+OIL.....	30
2.5.2 RDFS (RDF Schema, esquema RDF).....	30
2.5.3 OWL (Web Ontology Lenguaje).....	30
2.5.3.1 SUBLENGUAJES DE OWL.....	31
▪ OWL Life.....	31
▪ OWL DL.....	31
▪ OWL Full.....	31
2.5.3.2 SINOPSIS DEL LENGUAJE OWL.....	32
➤ Sinopsis del OWL Lite.....	32
➤ Sinopsis OWL DL y Full.....	33
2.7 CONCEPTOS DE VISUALIZACIÓN.....	33
2.7.1 VISUALIZACIÓN.....	33
2.7.2. VIRI – INTERFACES VISUALES PARA LA RECUPERACIÓN DE LA INFORMACIÓN... 34	34
2.8 ¿COMO PARTICIPA LA HERRAMIENTA EN LA ETAPA DE DESARROLLO?.....	34
3 MARCO PRÁCTICO.....	36
3.1 DESCRIPCIÓN DE MÉTODOS.....	36
3.1.1 SIETE NIVELES DE LA VISUALIZACIÓN DE DATOS.....	37
➤ CLASIFICACIÓN.....	37
➤ FILTRADO.....	38
➤ MINERÍA O EXPLOTACIÓN.....	38
➤ REPRESENTACIÓN Y REFINACIÓN.....	38

➤ INTERACCIÓN.....	38
3.1.2 ELECCIÓN DE LA TÉCNICA DE VISUALIZACIÓN A UTILIZAR.....	39
i) JERÁRQUICAS:.....	39
ii) REPRESENTACIONES DE REDES:.....	39
iii) REPRESENTACIONES DE DISPERSIÓN:.....	39
iv) MAPAS:.....	40
3.1.2.1 ESTUDIO DE LOS MÉTODOS.....	40
3.1.2.2 CONCLUSION DEL ESTUDIO.....	41
3.2 MÉTODO CIENTÍFICO.....	41
3.2.1 MÉTODO ANALÓGICO-SINTÁCTICO.....	42
3.2.2 MÉTODO INDUCTIVO-DEDUCTIVO.....	42
3.2.3 MÉTODO ABSTRACCIÓN-CONCRECIÓN.....	42
3.3 DESCRIPCIÓN DE COMPORTAMIENTO.....	42
3.3.1 DESCRIPCION DE TECNOLOGIAS.....	42
3.3.1.1 JENA.....	42
3.3.1.2 ZVTM.....	44
3.3.1.3 PROTEGE-OWL 3.2.1.....	45
3.3.2 INTERACCIÓN ENTRE COMPONENTES.....	45
4 PROCESO DE INVESTIGACIÓN.....	46
4.1 CONSTRUCCIÓN DE UNA ONTOLOGIA POR PASOS.....	46
PASO 1: DOMINIO.....	47
PASO 2: CONSIDERAR LA REUTILIZACION DE ONTOLOGIAS.....	48
PASO 3: TERMINOS IMPORTANTES PARA LA ONTOLOGIA.....	48
PASO 4: CLASES.....	49
PASO 5: DEFINIR LAS PROPIEDADES DE LAS CLASES.....	51
PASO 6: DEFINIR FACETAS DE LOS SLOTS (PROPIEDADES).....	52
PASO 7: CREAR INSTANCIAS.....	53
4.2 HERRAMIENTAS QUE INTERVIENEN EN LA ELABORACIÓN DEL SOFTWARE... 53	
4.2.1 SOPORTE PARA PROYECTOS E INTERFACE DE USUARIO.....	54
4.2.1.1 PANELES DE EDICIÓN.....	54
4.2.2 ALMACENAMIENTO Y VALIDACIÓN DE DATOS SINTÁCTICOS.....	54
4.2.2.1 Validación de Datos en la edición de los Componentes.....	55
4.2.3 GRAFICADOR.....	55
4.2.3.1 ALMACÉN DE INFORMACIÓN GRÁFICA.....	55

4.2.3.2 REPRESENTACIÓN DE LOS COMPONENTES CLASE, PROPIEDAD, INSTANCIA Y PREDICADO.....	56
4.2.3.3 ESPACIO DE TRABAJO.....	57
4.2.3.4 VISTA GLOBAL Y PARTICULAR DEL ESQUEMA DE LA ONTOLOGÍA.....	58
4.2.3.5 FILTRO DE DATOS.....	60
4.2.3.6 ALMACENAMIENTO PERMANENTE DE DATOS GRAFICOS.....	61
4.2.3.7 ALMACENAMIENTO PERMANENTE DE GRAFICO.....	63
4.2.4 ANALIZADOR SEMÁNTICO.....	63
4.3 PRESENTACIÓN DE LA SIMULACIÓN.....	63
4.3.1 PRIMERA SIMULACIÓN.....	63
4.3.2 SEGUNDA SIMULACIÓN.....	69
5 DISCUSION Y CONCLUSIONES.....	72
5.1 HERRAMIENTA DE EDICION DE ONTOLOGIAS.....	72
5.2 CONCLUSIONES.....	73
BIBLIOGRAFIA.....	75
ANEXOS.....	77
ANEXO 1: División de las VIRI (Interfaces para la recuperación de la Información).....	77
ANEXO 2: Listado de técnicas de visualización y sus respectivas referencias en orden alfabético.....	77
ANEXO 3: Ejemplo de Ontologias en lenguaje OWL del ejemplo completo de la Página Web de la Facultad de Ciencias Puras y Naturales enlace Institucionalidad.....	78
ANEXO 4: Diagrama de Clases que Intervienen en la Serialización de documentos XML para Almacenamiento de Datos Gráficos permanente.....	82

LISTA DE GRAFICAS

FIGURA 2.1	Tres niveles de Interoperabilidad	21
FIGURA 2.2	La Web actual vs. La Web Semántica	23
FIGURA 2.3	W3C conceptos de la Web Semántica.....	24
FIGURA 2.4	FOAF-a-Matic.....	25
FIGURA 2.5	Buscador Swoogle	26
FIGURA 2.6	Buscador Kartoo	26
FIGURA 2.7	Ejemplo de una ontología	28
FIGURA 2.8	Comparación de un RDF con un Grafo Dirigido	29
FIGURA 2.9	Sublenguajes de OWL	31
FIGURA 2.10	Ejemplo de una ontología	32
FIGURA 2.11	Herramienta de Generación de Ontologías	35
FIGURA 3.1	Área de Estudio de las VIRI	37
FIGURA 3.2	Diagrama de Componentes	45
FIGURA 4.1	Página Web de la Facultad de Ciencias Puras y Naturales.....	47
FIGURA 4.2	Jerarquía de Clases.....	51
FIGURA 4.3	Jerarquía de Clases, Ejemplo de Ontología.....	52
FIGURA 4.4	Estructura de Almacén Temporal de Información Gráfica	56
FIGURA 4.5	Representación de Clase, Propiedad e Instancia.....	57
FIGURA 4.6	Espacio de Trabajo.....	57
FIGURA 4.7	Interacción de clases para acciones de Zoom, Radar y Navegador.....	58
FIGURA 4.8	Navegador del Esquema.....	59
FIGURA 4.9	Radar del Esquema.....	59
FIGURA 4.10	Algoritmo para filtro "visodFilter".....	60
FIGURA 4.11	Árbol estructura Almacén Permanente de Esquema Grafico.....	62
FIGURA 4.12	Nuevo Proyecto.....	64
FIGURA 4.13	Nueva Clase.....	64
FIGURA 4.14	Validación en la Clase.....	64
FIGURA 4.15	Crear Propiedad.....	65
FIGURA 4.16	Barra de Estado Panel de Trabajo.....	65
FIGURA 4.17	Nuevo Predicado.....	65
FIGURA 4.18	Insertar Clase.....	65
FIGURA 4.19	Barra de Herramientas	65
FIGURA 4.20	Predicado	66
FIGURA 4.21	Ejemplo en Esquema Grafico	66
FIGURA 4.22	Ejemplo en formato OWL.....	66
FIGURA 4.23	Ejemplo en formato OWL.....	68
FIGURA 4.24	Barra de Herramientas.....	69
FIGURA 4.25	Crear Instancia.....	69
FIGURA 4.26	Editor Instancia.....	69
FIGURA 4.27	Editor Instancia.....	70
FIGURA 4.28	Archivos resultantes.....	70
FIGURA 4.29	Archivo "ejemplo.owl".....	71
FIGURA 4.30	Exportar a archivo PNG.....	71
FIGURA 5.1	Interacción de Componentes, dentro del desarrollo de la Web Semántica.....	73
FIGURA Anexo1:	Área de Estudio de las VIRI.....	77
FIGURA Anexo3:	Ejemplo OWL Página Web F.C.P.N. enlace Institucionalidad.....	81
FIGURA Anexo4:	Diagrama de Clases de Serialización Documentos XML para Datos Gráficos.....	82

LISTA DE TABLAS

TABLA 1.1	Lista de Problemas.....	15
TABLA 2.1	Siete Niveles de la Web Semántica.....	24
TABLA 2.2	Características OWL Lite.....	32
TABLA 2.3	Agregación al OWL Lite para OWL DL y Full.....	33
TABLA 3.1	Comparación de Métodos.....	41
TABLA 4.1	Términos Importantes Ejemplo de Ontología.....	49
TABLA 4.2	Jerarquía de Clases, Ejemplo de Ontología.....	50
TABLA 4.3	Facetas de los Slots (propiedades) de la Clase Estudiante.....	53
TABLA 4.4	Ejemplo Instancias.....	53
TABLA Anexo2:	Listado de técnicas de visualización y sus respectivas referencias en orden alfabético.....	77
TABLA Anexo3:	Ejemplo OWL Página Web F.C.P.N. enlace Institucionalidad.....	78



1 PRESENTACIÓN

1.1 INTRODUCCIÓN.

La Web Semántica a través de las ontologías, le dan un significado a la información hasta ahora solo expuesta al entendimiento de los humanos de forma variada e imprecisa.

La principal tarea de la Web Semántica, es que ofrece la forma de organizar la información, haciendo la función de comunicación más eficiente, lo cual beneficia no solo a los usuarios consumidores, sino, a los usuarios que proveen aplicaciones o servicios, tales como buscadores, servicios de comercio electrónico, páginas comerciales, etc.

Tales campos de implementación hoy en día se convirtieron no solo en grandes redes de comunicación mundial, sino en motores activos de desarrollo empresarial. Es así, que la incorporación de la Web Semántica tiene aspectos estratégicos en varios campos: en cuanto al desarrollo de una economía sólida, en cuanto al nivel de comunicación, por tanto de desarrollo social, en cuanto al desarrollo científico, sin mencionar la calidad de trabajo en el entorno empresarial.

El éxito de las metas propuestas para el desarrollo de la Web Semántica, lleva un alto grado de responsabilidad por parte de los desarrolladores de páginas Web, ya que se incorpora una tarea fundamental para ellos, y es la elaboración de sus propias bases de conocimiento, denominadas ontologías.

Por supuesto, es innegable que esta labor es difícil de emprender por diferentes causas, una de ellas es, para muchos desarrolladores, el proceso de análisis conceptual de su propia

información, lo cual se convierte a veces en moroso, engorroso, lento, comparable al análisis que se realiza al diseñar una base de datos común, inclusive, podemos decir que comparando con este, el desarrollo de ontologías es más amplio, más minucioso y complicado por el nivel de detalle que este maneja.

Recordemos que el diseño de dichas ontologías, asegurarán su éxito, de acuerdo a la dirección a la que se orienta la página Web en sí. Es por esto, que encontramos una gama de ontologías bastante diversificadas en general, muchas de las cuales contienen estructuras muy complejas. Por este hecho, necesitan ser vistas con varios niveles de abstracción y en diferentes contextos para ser entendidos claramente.

Por otro lado, al momento de aplicar una ontología, es muy probable hacer el uso de otra ontología ya elaborada, tal que se pueda recuperar dicha estructura en su totalidad o en parte. Surgiendo así, otra posible tarea de carácter complejo.

En otro punto, la tarea de soporte de cualquier base de conocimientos en general, ha sido tema de investigación, no obstante, específicamente la tarea de implementar ontologías se conjunciona con el ciclo de vida de la propia página Web, por tanto diremos el mantenimiento de esta base de conocimientos en específico es de carácter relevante para llevarse a cabo con un buen procedimiento, ya que debe ser acorde al mantenimiento que debe llevar la página Web. Asimismo, si hablamos de soporte, también tenemos que referirnos, a un adecuado manejo de la documentación de estas ontologías. En esta parte, además de la formalización, es un hecho que una presentación agradable no tendrá muchos inconvenientes en el proceso de aprendizaje de las mismas por otros usuarios.

Por último, mencionemos una tendencia en la investigación científica acerca de modelos de visualización de ontologías orientado a los grandes espacios de información, lo cual significa también que existe un área abierta dentro del campo de ciencias de la computación a ser explorada actualmente.

El propósito del presente trabajo de investigación, con la elaboración de esta herramienta, es ofrecer al desarrollador Web la simplicidad que se busca en el diseño exitoso de la estructura de una base de conocimientos como son las ontologías, para su incorporación en un proyecto de Web Semántica, de manera que pueda acceder a realizar tareas complementarias como documentación, mantenimiento, soporte de la misma.

En el capítulo 1, de forma general se hace una breve introducción del área informática en la cual se desea incursionar, poniendo énfasis en la situación actual de la problemática y como ingresa nuestra propuesta en este contexto.

En el capítulo 2, se recopila información acerca de los conceptos generales dentro de la Web Semántica, dando un realce a los conceptos referentes a Ontologías y Visualización de Datos.

En el capítulo 3, observamos las metodologías empleadas dentro del campo científico y metodologías dentro del área de la Visualización de Datos, de tal forma que se explica como se llevo el proceso de elaboración del presente trabajo.

En el capítulo 4, encontramos un detalle del proceso de diseño de Ontologías, así como la participación de la herramienta dentro de este diseño y una explicación modular de dicho trabajo.

1.2 ANTECEDENTES.

Puesto que la introducción de la Web Semántica a creado polémica en cuanto al éxito de su incursión a nivel mundial, de alguna manera, la tendencia de incorporar datos adicionales en las páginas Web, con una funcionalidad diferente a la que pudiera proporcionar una base de datos, se ha implantado silenciosamente a través de distintas tecnologías, tales como los micro formatos, los XML (eXtensible Markup Lenguaje), entre otros; inclusive la aparición de las páginas de estilos.

En Bolivia no existen pruebas visibles de proyectos de Web Semántica, con relevancia, sino solo proyectos universitarios, que no son de alguna forma conocidos.

Por lo mismo, tampoco se realizó ningún trabajo de conocimiento público en cuanto a editores de ontologías con estas características, sin embargo, a nivel mundial existen algunas herramientas, las cuales siguen en la labor de investigación. E aquí algunos ejemplos:

- VisualKii (<http://www.visualkii.com/>) es un proyecto múltiple plataforma de programación visual basado en Java. Este tiene librerías para procesamiento de modelos RDF, N3 y N-

TRIPLES por flujo de datos visual y pasos de procesamiento de adaptación. También incluye soporte para consultas SPARQL.

- Isaviz, (<http://www.w3.org/RDF/~isaviz>), es una herramienta de creación visual para navegar y crear modelos RDF representados como grafos. Desarrollado por Emmanuel Pietriga de INRIA y fomentado por W3C y Xerox Research Centre Europe, bajo licencia W3C.
- Protégé, (<http://protege.stanford.edu/>), es considerado el de mayor prestigio, elaborado bajo soporte de la Universidad de Stanford, es de licencia pública.

Cabe resaltar que estas y otras herramientas no cuentan con un equilibrio entre eficiencia y usabilidad, haciendo visible muchas veces la complejidad con la que fueron diseñadas. También se ve las funcionalidades incompletas que individualmente presentan, algunas las cuales van incorporándose por medio de software adicionales. No esta por demás mencionar el hecho de que todas estas herramientas no están en español, lo cual hace difícil y hasta imposible su comprensión para usuarios de poca experiencia con el idioma inglés.

1.3 SITUACIÓN PROBLEMÁTICA.

La implementación de nuevas tendencias científicas (tal el caso de los Sistemas Basados en Conocimiento), dejan un duro trabajo a los investigadores de todo el mundo. Es así que en Bolivia, por las diferentes características de un país en vías de desarrollo, este trabajo se hace más complejo. De esta manera la implementación de un proyecto de Web Semántica puede apreciarse como un objetivo de complejidad importante, lo cual disminuye el entusiasmo de involucrarse con el mismo. Esta situación se puede repetir con proyectos en otras áreas científicas, y repercute en cuanto al nivel de investigación científica de nuestro país.

Es así, que se distingue el hecho de que la solución propuesta, permita involucrar a desarrolladores que investigan, en una implementación de un proyecto que eleve el nivel científico de su proyecto.

1.4 FORMULACIÓN DEL PROBLEMA.

En general mencionaremos una lista de problemas y soluciones:

TABLA 1.1 Lista de Problemas

Problema	Causa	Efecto	Solución
Necesidad de introducir la propuesta de la Web Semántica.	Transformar un sitio Web con un nivel alto en interoperabilidad semántica.	Rehacer el diseño de la metadata de la Web.	Incrementar el número de sitios adecuados a esta propuesta.
Baja implementación de sitios construidos con ontologías.	Complejo análisis conceptual en el desarrollo de estructuras de una ontología.	Pérdida de datos importantes a la hora de implementar la propuesta.	Coadyuvar al análisis conceptual mediante un esquema visual de la estructura de una ontología.
No existen suficientes sitios Web con sustento Semántico de carácter robusto.	Poco entendimiento de manejo de ontologías.	No se realiza la visita a la Web.	Incremento de sitios Web con respaldo de una ontología.
Web Semántica con lentitud en intromisión a los sitios nacionales.	Mayor tiempo y esfuerzo en la elaboración de ontologías.	Proyectos de Web Semántica no fructíferos.	Rápido procesamiento de estructuras de una ontología.
No existen los suficientes sitios que implementen Web Semántica con un nivel de mantenimiento adecuado.	Difícil comprensión de ontologías desarrolladas por otros diseñadores, lo cual le cierra al desarrollador la opción de un diseño eficiente.	No se realiza la visita a la Web.	Coadyuvar con el proceso de mantenimiento de una Web Semántica.
Necesidad de comprender el diseño de una Web Semántica para otros desarrolladores.	Falta de documentación formal y entendible de complicadas redes de conceptos.	Se pierde la oportunidad de conocer el sitio.	Reportes comprensibles.
Necesidad de sitios de Web Semántica más adecuados a la realidad en que vivimos.	Recursos y herramientas no están en idioma español.	No se realiza la visita a la Web.	Implementar el software en idioma español.
Necesidad de acceder a la red con estándares mundiales.	Bajo nivel de formalización con estándares W3C a nivel web.	Instituciones mundiales no visitan la Web.	Recursos con estándares.

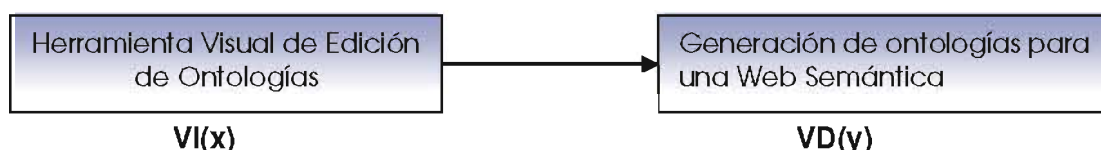
[Fuente: Velásquez L., Datos Propios]

1.4.1 DEFINICIÓN DEL PROBLEMA.

De la lista de problemas, extraemos el siguiente problema de investigación:

¿La herramienta visual de edición de ontologías permitirá proporcionar una fácil y ágil generación de una ontología para una Web Semántica?

1.4.2 IDENTIFICACIÓN DE VARIABLES.



1.4.3 OPERACIÓN DE VARIABLES.

Concepto:	Herramienta Visual de Edición de Ontologías, es aquella que sirve de medio para conseguir el objetivo de edición esperada, puede ser una operación que establece formalmente una relación sintáctica relevante entre una visión gráfica y un lenguaje.
Variable:	$VI(x)$ Herramienta visual de edición de ontologías.
Indicador:	Número de etiquetas efectivamente transpuestas a OWL.
Instrumento:	Graficador, analizador semántico.
Valor:	Porcentaje de efectividad en el número de equivalencias efectivas que forman parte del proceso de transformación.

Concepto:	Generación de ontologías para su utilización en una Web Semántica, significa elaborar documentos en formato OWL, los cuales puedan ser adheridos a la Web y así cumplir con objetivos como elevar el nivel de interoperabilidad semántica del sitio.
Variable:	$VD(y)$ Generación de ontologías para una Web Semántica.
Indicador:	Documentos conformes al formato OWL.
Instrumento:	Pruebas de validación de OWL
Valor:	Resultado de pruebas de validación OWL.

1.5 OBJETO DE ESTUDIO.

El proceso de generación a partir de un esquema o diagrama visual, hacia una ontología en formato del lenguaje OWL.

1.6 JUSTIFICACIÓN.

El **diseñador de un proyecto de Web Semántica**, se adhiere una tarea a las establecidas, es la de diseñar una ontología exclusiva. Esta labor puede llegar a ser fácil o complicarse, de acuerdo al tipo de información que empleará dicha página y del nivel de especificación, pero la importancia radica en establecer esta ontología con la mayor precisión y claridad, las cuales son características comparables a las utilizadas en una base de datos.

Recordemos que las ontologías son formas de representar el conocimiento, es entonces que se pretende colaborar en la fácil implementación, centrando el esfuerzo del diseñador en el esquema y no así en el código implementado en los archivos.

En un proyecto de implementación de una Web Semántica el diseñador puede acceder a la forma mas cómoda de visualización, a través de opciones de configuración, obteniendo no solamente el diseño de forma gráfica apta para una documentación agradable a la vista, sino también al fácil mantenimiento de trabajos ya realizados.

La relevancia científica, es el empleo de los algoritmos de análisis semántico, a través de los cuales cualquier lenguaje puede representarse, conjuntamente la representación gráfica mediante modelos de visualización, serán empleados de tal forma que trabajen en conjunto.

Es un **aliciente que impulsa la transición de una web cualquiera a una Web Semántica**, de forma menos traumática, lo cual significa mayor número de usuarios (humanos, máquinas), en el campo de la Web Semántica.

1.7 HIPÓTESIS.

Hi: La herramienta visual de edición de ontologías genera una ontología a partir de un esquema gráfico, con el fin de mejorar la eficiencia en la etapa de desarrollo de una Web Semántica.

1.8 OBJETIVOS.

1.8.1 OBJETIVO GENERAL.

Desarrollar una herramienta de edición visual que posibilite la construcción esquemática de ontologías, ofreciendo la facilidad de convertir un esquema visual en un documento OWL a implementar en la práctica, verificando su correcta aplicación.

1.8.2 OBJETIVOS ESPECIFICOS.

- Describir las definiciones de importancia de la Web Semántica.
- Realizar un estudio sobre la división del lenguaje de OWL.
- Realizar un estudio sobre técnicas de visualización del conocimiento.
- Implementar en la herramienta de edición visual de ontologías, módulos que permitan interactuar al usuario con el esquema gráfico.
- Seleccionar las librerías adecuadas para el desarrollo de la herramienta de edición visual de ontologías.

1.9 LÍMITES Y ALCANCES.

1.9.1 LIMITES.

Se construirá una herramienta de generación de ontologías OWL contemplando entornos visuales e interactivos para una edición de esquema tipo mapa conceptual, el cual permitirá su conversión a una ontología en OWL, lista para adjuntar al sistema Web en sí.

La complejidad de las estructuras conceptuales, dependerán del usuario, pero se verá la manera de impedir incoherencias en dichas estructuras. Entonces diremos que las estructuras conceptuales a implementar, deberán ser coherentes, y siguiendo las normas estándar básicas para los sublenguajes OWL Lite y OWL DL.

En otro aspecto, el proceso inverso, es decir de OWL a un esquema visual, ya esta implementado en otras herramientas ya existentes, es así que no se ingresa en dicho objetivo.

1.9.2 ALCANCES.

La propuesta alcanza un procesamiento para la elaboración de Ontologías para una Web Semántica, contemplando la edición visual con un alto grado de navegabilidad entre objetos.

1.10 METODOLOGÍA.

1.10.1 MÉTODOS Y MEDIOS DE INVESTIGACIÓN CIENTÍFICA.

En cuanto a los métodos involucrados para el desarrollo, mencionaremos:

- **Método Analógico-Sintáctico**

Se contempla la necesidad de realizar un análisis y síntesis en varias etapas del proyecto.

- **Método Inductivo-Deductivo**

La inducción viene siendo la única forma de conseguir la deducción o deducciones propuestas en los objetivos.

- **Método Abstracción-Concreción**

Necesitamos tener una visión exacta del trabajo de investigación.

1.10.2 MÉTODOS Y MEDIOS DE INVESTIGACIÓN INFORMÁTICA.

Las herramientas involucradas en el trabajo son las siguientes:

- Lenguaje de programación Java Jdk1.6.0.
- IDE Netbeans 5.5.

- Librería en Java de código abierto: ZVTM para la graficación.
- Librería en Java Protege OWL, de código abierto, para estructuración de almacenes temporales.
- Librería en Java de código abierto: Jena, para el manejo de las ontologías.

1.11 APORTES.

1.11.1 APORTE TEÓRICO.

La visión objetiva del proceso en sí de una ontología es un aporte de importancia, el cual se distingue por proponer la transición entre lenguajes y esquemas como una tarea posible y además aceptable en algunas circunstancias.

Las técnicas de visualización a tomar, en la forma más eficiente, orientada a representación de ontologías y como se realiza la selección de acuerdo a las necesidades puntuales del usuario.

1.11.2 APORTE PRÁCTICO.

Como principal aporte se encuentra el software en sí, capaz de facilitar no solo en desarrollo, sino en el diseño de un complejo esquema conceptual. Esta tarea es la más importante para el éxito de un proyecto Web, es así que colaborará al diseñador reuniendo sus esfuerzos en el diseño mismo, y no en la forma de realizar este proceso.

También remarcando el hecho que en los países de habla hispana se esta incorporando muy lentamente dentro de la ciencia en si que ocupa la Web Semántica, una de las razones son las solicitudes de herramientas que hagan frente a las nuevas tendencias planteadas.

2 MARCO DE REFERENCIA.

2.1 WEB ACTUAL.

Tiene una estructura hipertextual: la World Wide Web (WWW) contiene una gigantesca cantidad de documentos de hipertexto. En general se dice que la red permite acceder a recursos Web (dispositivos físicos como impresoras, ordenadores, agendas electrónicas, etc., o dispositivos de software accesibles como páginas web, imágenes, videos, directorios, etc.) [Fuente: Artículo "El futuro de la Web"; Abián, Miguel Ángel].

"**Hipertexto**" fue introducido por Ted Nelson en 1965, con él se refería a una colección de documentos (nodos) con referencias cruzadas o enlaces que, mediante la ayuda de programas navegadores, permiten al lector moverse sencillamente de un documento a otro.

Es en este contexto que podemos definir la principal **función de la Web actual**:

- ✓ Acceso interactivo a documentos y aplicaciones, tanto para humanos como para máquinas (otras aplicaciones o programas). Cabe destacar que ambos usuarios, van en crecimiento.

Sin embargo, es un hecho que esta forma de comunicación (hipertexto) ha sido muy exitosa en ciertos puntos, pero, por su importancia, enfocaremos los principales problemas.

2.1.1 PROBLEMAS DE LA WEB ACTUAL

2.1.1.1 INTEROPERABILIDAD.

Capacidad de dos o más sistemas o componentes para intercambiar información y usar información que ha sido intercambiada [Fuente: Sociedad de Computación IEEE].

Desarrollamos los tres tipos de interoperabilidad, posteriormente un gráfico explicativo:

Interoperabilidad técnica.

Se refiere a la capacidad de los SI para intercambiar señales. [Fuente: Artículo El futuro de la Web, Abián, Miguel Ángel].

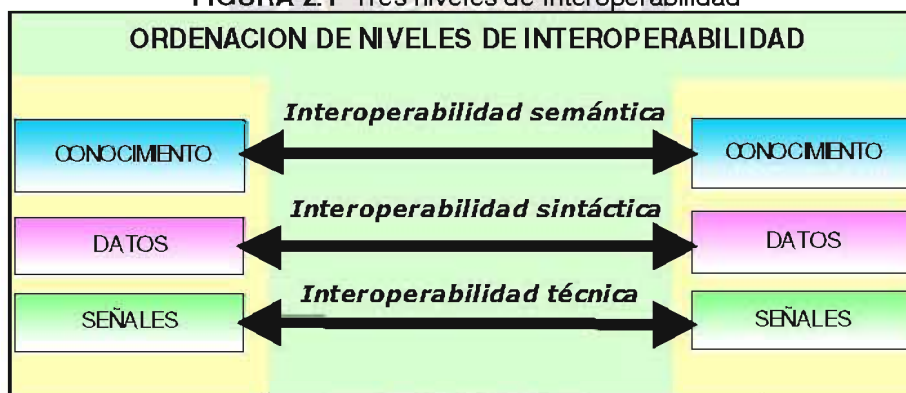
Interoperabilidad sintáctica.

Se refiere a la capacidad de los SI para leer datos procedentes de otros SI y obtener una representación que puedan usar (objetos, por ejemplo). [Fuente: Artículo El futuro de la Web, Abián, Miguel Ángel].

Interoperabilidad semántica.

Se refiere a la capacidad de los SI para intercambiar la información.

FIGURA 2.1 Tres niveles de Interoperabilidad



[Fuente: Artículo: "El futuro de la Web" Abián, Miguel Ángel]

Es así que podemos concretar dos fundamentales carencias de la Web actual:

- No incorpora mecanismos que permitan el *proceso automático de la información*; esto es, que hagan que la información pueda ser procesada por máquinas. [Fuente: "El futuro de la Web", Abián, Miguel Ángel].
- En segundo lugar, la Web actual no incluye mecanismos para la *interoperabilidad completa* de los sistemas de información (SI) basados en la Web, carece de interoperabilidad semántica.

Este aspecto afecta directamente a la falta de entendimiento entre aplicaciones, e indirectamente también a los seres humanos, que cada día se hacen mas dependientes de dichas aplicaciones, o servicios. El ejemplo más común son los buscadores, los cuales, a pesar de los muchos esfuerzos por mejorar sus algoritmos de búsqueda, no tendrán forma de alcanzar los niveles de entendimiento esperados.

2.2 WEB SEMÁNTICA.

El **W3C** (sitio <http://www.w3.org/>) es el organismo que regula aspectos esenciales de la Web tales como el lenguaje (X)HTML con el cual se crean las páginas y los sitios web. Su director es el propio fundador de la Web, Tim Berners-Lee, por lo que sus recomendaciones, tienen un carácter normalizador de gran prestigio y enorme influencia a nivel mundial.

2.2.1 DEFINICIÓN: WEB SEMÁNTICA.

Tim Berners-Lee, el creador de la WWW tenía otra visión sobre lo que es hoy por hoy la web que conocemos. El pensó en una red de recursos que nos permitiera programar agentes que navegaran la infinitud de sitios pudiendo obtener la información que necesitamos sin tener que indicarle de donde obtenerla o que significado debe tener cada recurso, transformando finalmente esa información a un formato que sea fácilmente entendible por nosotros. Esa Web, que aún se encuentra en una fase de desarrollo, es lo que se conoce como la Web Semántica.

Es así que desglosamos la siguiente definición:

“La Web Semántica es un conjunto de iniciativas destinadas a realizar la organización, estructuración y codificación, de manera tal que se puedan realizar inferencias, es decir razonar, para así posibilitar un procesamiento sistemático, consistente”.

2.2.2 IMPORTANCIA DE LA WEB SEMÁNTICA.

Para resaltar su importante incursión en la Web actual, podemos hacer referencia dos visiones, La visión de la Inteligencia Artificial y Procesamiento Robusto. Los objetivos de ambas visiones nos dicen que la Web Semántica posibilita el razonamiento, y un proceso intensivo dentro de los sistemas ya existentes.

Ahora mencionemos el principal problema que atiende la Web Semántica: *“Dificultad para encontrar información”*. Puesto que, los buscadores son en todos los casos otros

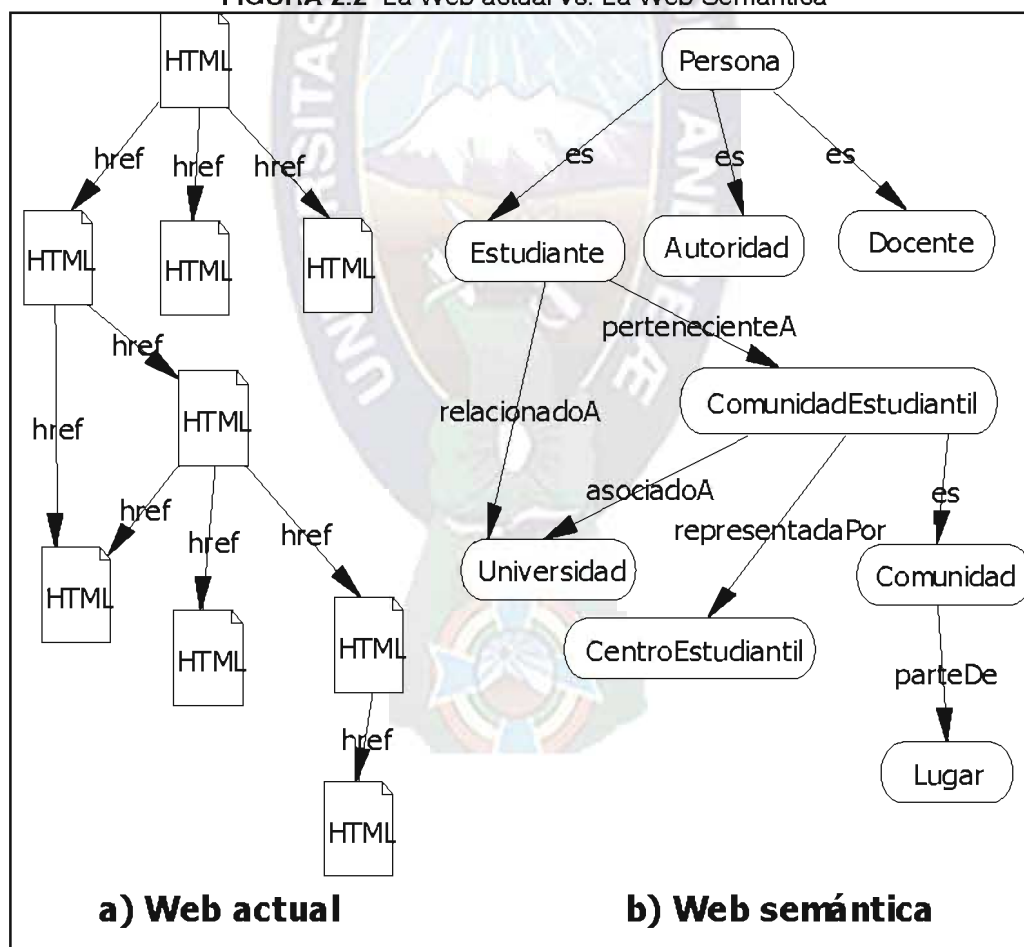
programas, y no el hombre, por el mismo hecho que la información en la web es de un inmenso tamaño.

Al respecto, la siguiente explicación nos da a entender como funcionan los buscadores actuales, y nos dará una pauta de cómo se manifiestan los problemas:

- Escasa precisión en los resultados.
- Alta sensibilidad del vocabulario empleado en la búsqueda.

Es así, que a través de su estructura, los buscadores se hacen intérpretes del verdadero significado de una búsqueda, a través de un estilo de búsqueda inferencial, el cual permite ver a las páginas web no solo como colecciones de datos, sino que significa realmente la información, el significado de los datos. En el siguiente cuadro observamos como se observa una Web, a los ojos de un buscador inferencial.

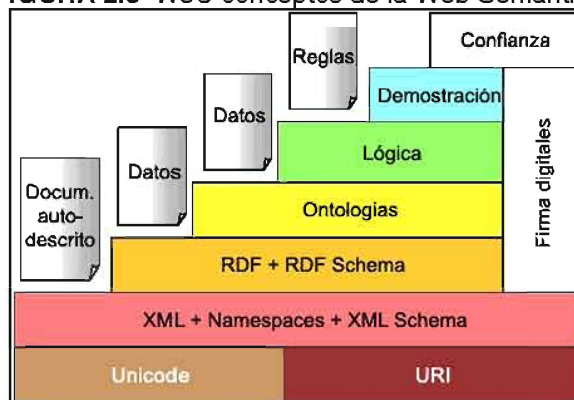
FIGURA 2.2 La Web actual vs. La Web Semántica



[Fuente: "La Web Semántica"; Castells, Pablo; Datos Propios]

2.2.3 CONCEPTOS DE LA WEB SEMÁNTICA.

FIGURA 2.3 W3C conceptos de la Web Semántica



[Fuente: W3C Consortium]

2.2.3.1 LOS 7 NIVELES DE LA WEB SEMÁNTICA

En este esquema observamos los 7 niveles con los cuales trabaja la Web Semántica, representados en la Figura 2.3. A continuación, destacamos una breve descripción de cada una:

TABLA 2.1 Siete Niveles de la Web Semántica

Niveles	Descripción
1 Unicode + URI	<i>Unicode</i> es un sistema internacional estándar que proporciona un número único para cada carácter, sin importar la plataforma ni el programa.
2 XML+NS+XML SCHEMA	<i>eXtended Markup Language (XML)</i> es un sistema que permite definir lenguajes de marcas para usos específicos. <i>Name Spaces (NS)</i> permite combinar diversos lenguajes de marcado creados con XML en un mismo documento. <i>XML Schema</i> sirve para definir tipos de documentos complejos en los que se pueden especificar tipos de datos, listas de componentes y restricciones similares a las del diccionario de datos típico de una base de datos.
3 RDF + rdfschema	<i>Resource Description Framework (RDF)</i> es un modelo de representación de metadatos que, entre otras cosas, permite representar recursos digitales tales como sitios o páginas web. RDF está concebido para representar cualquier clase de recursos (no solamente páginas publicadas en la web). <i>RDF Schema</i> , por su parte, es una extensión de RDF que aporta un lenguaje con mayor capacidad para representar relaciones semánticas complejas.
4 Vocabulario de Ontologías	Una <i>ontología</i> es una especificación formal de un dominio del conocimiento que, en su expresión más simple, se identifica con una taxonomía. Una taxonomía consiste en una jerarquía de conceptos y sus relaciones del tipo clase-subclase. Una ontología formaliza la relación de clase, añade otras relaciones y especifica propiedades para individuos y clases. <i>Vocabulario de Ontología</i> se refiere a una ontología concreta sobre un dominio concreto del conocimiento.
5 Lógica	En este contexto, <i>lógica</i> se refiere al estudio de las reglas formales que permiten determinar si un razonamiento se sigue de sus premisas. La lógica estudia, por tanto, la estructura de los razonamientos válidos. Se espera que los ordenadores del futuro puedan efectuar razonamientos sobre los recursos y servicios de la Web combinando los conocimientos expresados en las ontologías, los hechos declarados en los metadatos y la aplicación de reglas lógicas.

<p>6 Demostración</p>	<p>En este contexto, <i>Demostración</i> [matemática]. Se considera que un ordenador alcanza la máxima fiabilidad en sus razonamientos cuando es capaz de realizar demostraciones o, lo que es lo mismo a efectos prácticos, cuando es capaz de justificar el motivo por el cual tomó una decisión.</p>
<p>7 Confianza (+Firma Digital)</p>	<p>La última capa, <i>Confianza</i> debe servir para otorgar confianza a las transacciones en la Web a través que se llevarán a cabo no solamente entre usuarios y sitios web sino también entre programas de software; y todo ello tanto en el plano C2B (<i>consumidor a empresa</i>) como en el B2B (<i>empresa a empresa</i>). La <i>Firma Digital</i> proporcionará soporte específico a esta capa, tal como muestra el diagrama.</p>

[Fuente: artículo "La Web Semántica"; Codina, Lluís y Rovira, Cristófol].

2.2.4 EJEMPLOS DE USO DE LA WEB SEMÁNTICA.

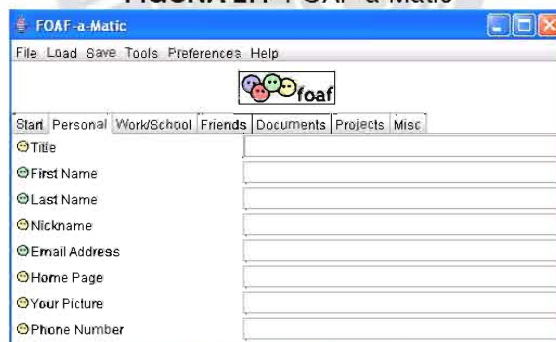
Aplicaciones:

- **Museo Suomi** (sitio: <http://museosuomi.cs.helsinki.fi/>), utiliza tecnologías de la Web Semántica.
- **Proyecto ANOTA** (sitio: <http://anota.ibit.org/>), pretende utilizar las tecnologías de anotaciones semánticas y de anotaciones externas para uso de las PYMES. Objeto de ayuda del Ministerio de Educación y Ciencia (PROFIT) y la Unión Europea.

Herramientas:

- **FOAF** (sitio: <http://xmlns.com/foaf/0.1/>), es una manera de describirse a uno mismo nombre, dirección de email, y la gente de quienes es amigo usando XML y RDF.

FIGURA 2.4 FOAF-a-Matic



[Fuente: Elaboración propia, del sitio <http://xmlns.com/foaf/0.1/>]

- **PET**, la versión de FOAF para descripción de animales (sitio <http://semtext.org/pets/pet-a-matic.htm>).
- **Esperanto Proyecto EU** (sitio <http://www.esperanto.net/>), construcción de herramientas escalables y automáticas para la conversión del contenido de la WWW en contenido semántico.
- **SEKT** Proyecto EU Semantically Enables Knowledge Technologies (sitio <http://www.sekt-project.com/>) Tecnologías de la Web Semántica aplicada a la gestión de conocimiento.

- **SWWS** Proyecto EU Semantic Web enabled Web Services (sitio <http://swws.semanticweb.org/>), tecnologías de los Servicios Web combinada con la Web Semántica.
- **DIP** Proyecto EU Data, Information, and Process Integration with Semantic Web Services (sitio <http://dip.semanticweb.org/>) Tecnologías de los Servicios Web combinada con la Web Semántica, y aplicada al EAI y EDI.

Buscadores y navegadores:

- **Haystack** (sitio <http://haystack.lcs.mit.edu/>), es un navegador que permite al usuario personalizar sus intereses, gustos, también importa archivos RDF/OWL, navegando por los recursos de la Web Semántica que ofrecen.
- **Semantic Web Search** (sitio <http://www.semanticwebsearch.com/>), buscador semántico.
- **Swoogle: Semantic Web Search** (sitio <http://swoogle.umbc.edu/>), buscador semántico.

FIGURA 2.5 Buscador Swoogle



[Fuente: Elaboración propia, del sitio <http://swoogle.umbc.edu/>]

- **Kartoo** (<http://www.kartoo.com/>), es un Topic Map francés, constituye una muestra de las propuestas para la visualización de la web semántica, o red semántica, aunque este es un metabuscador, representa una forma inteligente de búsqueda inteligente.

FIGURA 2.6 Buscador Kartoo



[Fuente: Elaboración propia, del sitio <http://www.kartoo.com/>]

2.3 ONTOLOGÍAS.

Iniciemos aclarando que la Web Semántica tiende a emplear diferentes tecnologías, las cuales buscan distintas formas de representación de la base de conocimientos, necesaria para el desempeño de la web, asimismo, el segundo ingrediente importante esta en, como elaborar esta base de conocimientos. Entonces decimos que el adquirir una determinada tecnología, las cuales explicaremos mas adelante, no se mezclan con la construcción de las ontologías en sí, un claro ejemplo de esto, se tiene al nombrar en que campos puede intervenir una ontología: ingeniería del conocimiento, gestión del conocimiento, procesamiento del lenguaje natural, etc.

También cabe resaltar, que las ontologías son una visión conceptual, que formaliza los antiguos conceptos (metadatos) asimilados para manejar las bases de conocimientos para una web. Por este mismo hecho, cumple ciertas normas, y además es posible su representación a través de diversos lenguajes.

Ahora, habiéndose explicado el contexto en el se presentan las ontologías, definamos el concepto:

Ontología es una jerarquía de conceptos con atributos y relaciones, que define una terminología consensuada para definir redes de conceptuales de unidades de información interrelacionadas. Una ontología proporciona un vocabulario de clases y relaciones para describir un dominio, poniendo el acento en la compartición del conocimiento y el consenso de la representación de este. [Fuente: "La Web Semántica", Castells Pablo].

Toda ontología modela, mediante conceptos y relaciones, lo que conocemos sobre un dominio o un área de conocimiento.

Las ontologías se representan mediante:

- Clases, definiciones de conceptos del dominio,
- Propiedades y atributos de las clases, o de los conceptos,
- Relaciones entre clases y restricciones sobre atributos y propiedades de las clases.

2.3.1 ESTRUCTURA DE UNA ONTOLOGÍA.

Básicamente los grados que se presentan son:

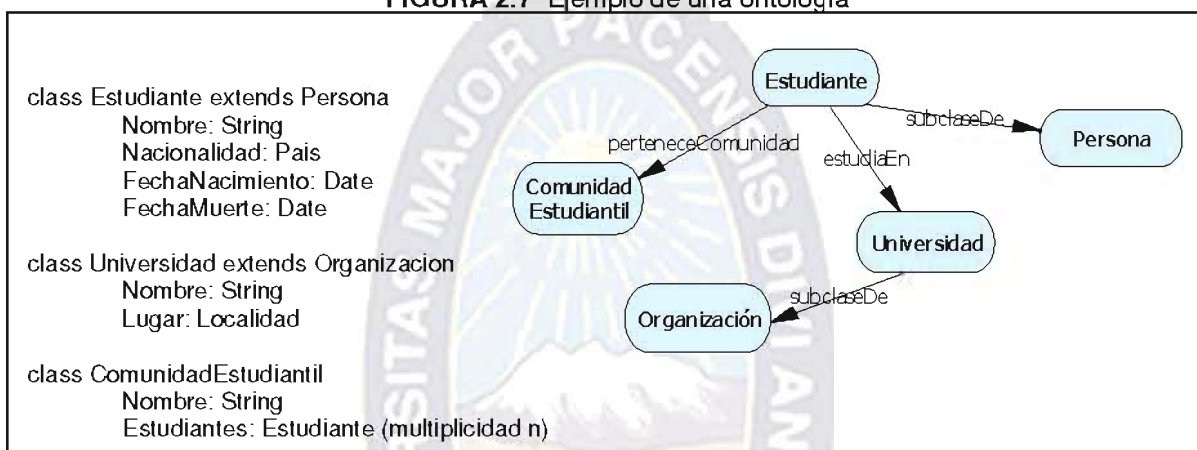
- Vocabularios,
- Taxonomías simples, se refiere a la jerarquización de términos,

- Sistemas relacionales, la definición de nuevas relaciones, y
- Teorías axiomáticas y reglas de inferencia, nos referimos a enunciados lógicos adecuados a una relación.

Estas se someten a restricciones: estáticas (estructura, tipos de valores, etc.), dinámicas (valores temporales, dependencias de contexto), suaves (cuando no se someten estrictamente).

Ejemplo:

FIGURA 2.7 Ejemplo de una ontología



[Fuente: "La Web Semántica" Castells, Pablo, Datos Propios]

2.3.2 TIPOS DE ONTOLOGÍAS.

De acuerdo al tipo de dominio que abarque, se distinguen cuatro tipos:

- **Alto Nivel**, ya que su dominio es amplio.
- **De Dominio o Genéricas**, verticales, orientadas a un dominio específico. Ejemplo: física.
- **De Tarea**, horizontales, orientadas a una tarea en particular. Ejemplo: Servicio Web.
- **De Aplicación**, el dominio lo tiene una aplicación en específico.

Como ejemplos de ontologías, se hicieron algunos intentos para definir una ontología para un dominio en específico [Fuente: sitio <http://www.ontology.org/>], pero con buena aceptación podríamos nombrar a Dublín Core [Fuente: sitio oficial <http://www.dcmi.org/>].

Las ontologías pueden ser representadas de forma esquemática, tal como es un mapa conceptual de una página web, al igual que en algunas ocasiones se representa algún mapa navegacional.

Para muchos esta es la forma más adecuada de comprensión y entendimiento de una base de conocimientos (redes semánticas), tal y como se representa una red cualquiera.

2.4 TECNOLOGÍAS PARA LA WEB SEMÁNTICA.

2.4.1 RDF (Resource Description Framework, marco de descripción de recursos).

Es un lenguaje para representar información sobre recursos de la Web (metadatos), cuyo objetivo es “especificar semántica para los datos basados en XML, de una manera interoperable y estandarizada” [Fuente: W3C <http://www.w3.org>].

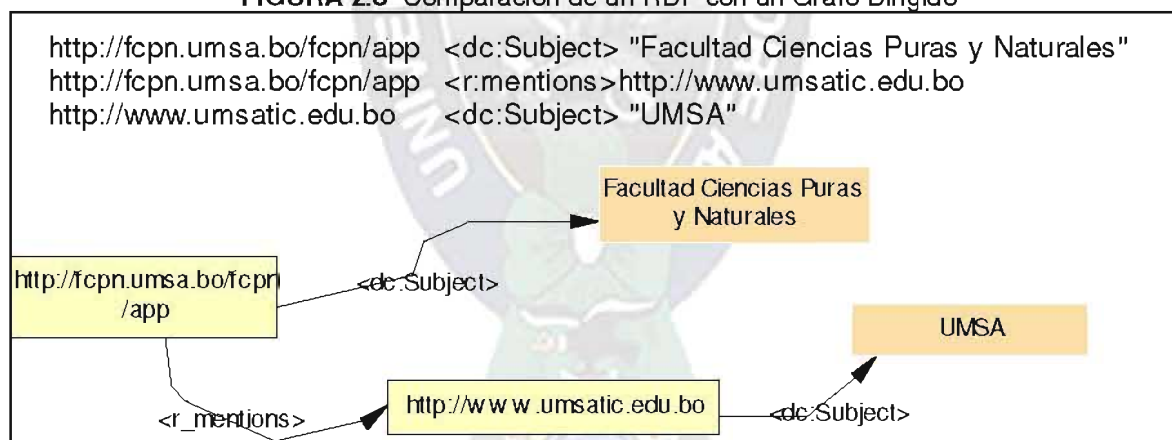
Un RDF puede ser representado de forma gráfica (tripleas sujeto, propiedad y objeto) y en XML (RDF/XML).

RDF define un modelo, compáramos esto con un grafo dirigido:

Nodos Origen =URIs o nodos anónimos
Arcos =URIs
Nodos destino (URIs, literales o nodos anónimos)

Ejemplo:

FIGURA 2.8 Comparación de un RDF con un Grafo Dirigido



[Fuente: "Tecnologías XML y Web Semántica", Depto. de Informática-Univ. Oviedo, Datos Propios]

El modelo RDF se puede representar en sintaxis XML, a esto se denomina serialización.

Paralelamente a las tecnologías mencionadas, las ontologías cubren de forma general como se va a representar el conocimiento, la información, como ya habíamos visto en un principio.

2.5 LENGUAJES DE REPRESENTACIÓN DE ONTOLOGÍAS.-

2.5.1 DAM+OIL.

DAM+OIL es un lenguaje bastante similar a OWL, pero que tuvo menos éxito. Es un predecesor de OWL.

2.5.2 RDFS (RDF Schema, esquema RDF).

RDFS es un lenguaje que se define en función de sentencias RDF. Además, permite especificar las entidades a las que puede aplicarse los atributos del vocabulario. Ejemplo, en el dominio de una biblioteca se podría usar RDFS para definir un vocabulario concreto (autorDe, tieneCarnetDeSocio, estaSancionado, etc.) y especificar condiciones tales como que esta estacionado, solo puede aplicarse a socios de biblioteca. [Fuente: Artículo "El futuro de la Web" Abián, Miguel Ángel].

2.5.3 OWL (Web Ontology Language).

Cabe resaltar que es una propuesta lanzada por la W3C, la cual tendrá importante relevancia en el presente trabajo.

OWL es un lenguaje desarrollado por W3C, tuvo una prevista adopción inminente por proyectos DARPA. Nació casi alrededor de junio de 2003. Se puede decir que es una formalización que ha sido adoptada por gran parte de los proyectos de Web Semántica. Es una extensión de RDFS y emplea el modelo de tripletas de RDF. En cierto modo, es un RDFS mejorado, que mantiene una buena relación entre eficacia computacional y poder expresivo.

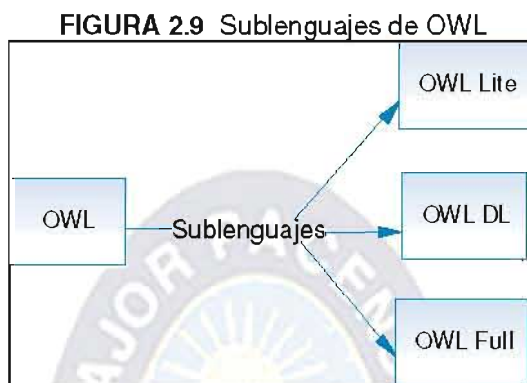
OWL tiene una sintaxis abstracta (independiente de cualquier representación computacional), una basada en XML y otra basada en RDF.

Entre las características destacaremos:

- Definición de clases mediante restricciones sobre propiedades.
- Definición de clases mediante operaciones booleanas sobre otras clases.
- Relaciones entre clases, por ejemplo: inclusión, disyunción, equivalencia.
- Propiedades de las relaciones, por ejemplo: simétrica, transitiva.
- Cardinalidad.
- Igualdad.
- Clases enumeradas.

2.5.3.1 SUBLENGUAJES DE OWL.

En la siguiente figura observamos los sublenguajes del OWL, de acuerdo a la W3C.



[Fuente: Elaboración Propia]

A continuación describiremos las atribuciones que presentan cada una de ellas:

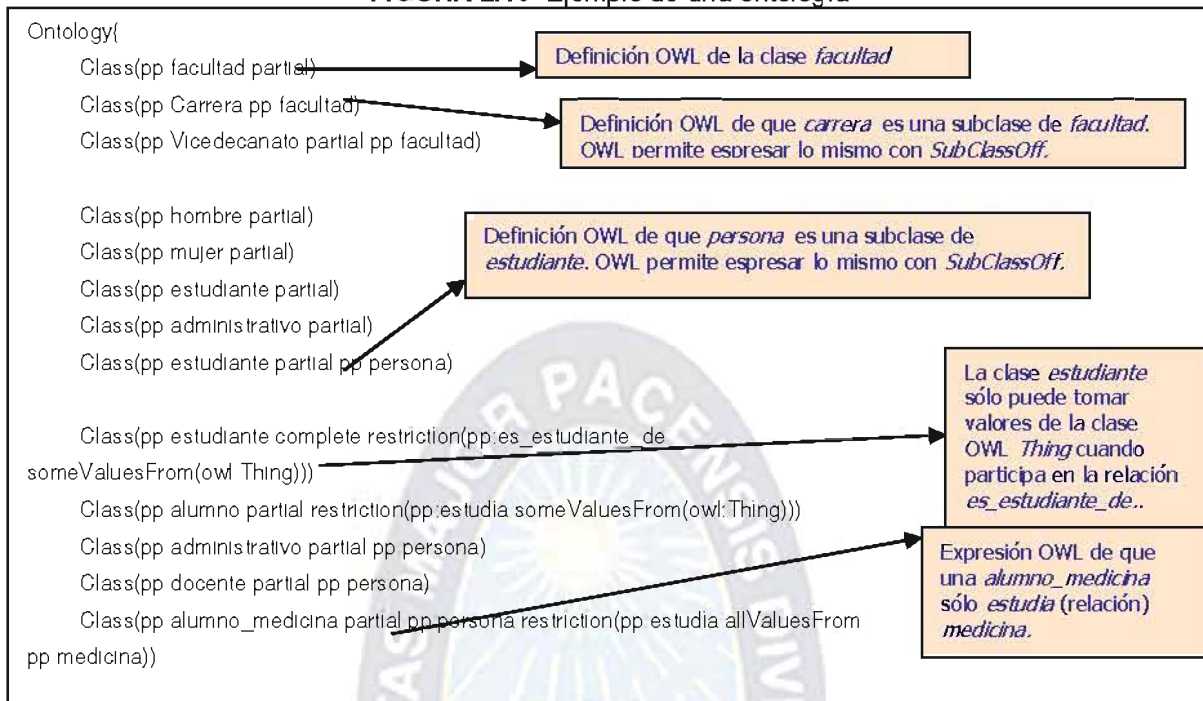
- **OWL Lite**
 - Jerarquía de clasificación.
 - Restricciones simples (por ej. Cardinalidad sólo 0 ó 1).
 - Facilita compatibilidad con otros modelos / paradigmas.
 - Facilita desarrollo de herramientas de autor.

- **OWL DL**
 - Máxima expresividad, mantenimiento, completitud y computabilidad.
 - Limitaciones las clases no son instancias ni tipos, los tipos no son instancias ni clases.
 - RDF es diferente a OWL DL.

- **OWL Full**
 - Libertad sintáctica de RDFS sin garantías computacionales.
 - OWL Full es una extensión de RDF. [Fuente: artículo "OWL" Castells, Pablo y Saiz, Francisco, del sitio <http://www.ii.uam.es/%7Ecastells/docencia/semanticweb/apuntes/6-owl.pdf>]

Como todo mapa conceptual, la visión en pseudo-código nos muestra como se puede traducir una ontología. Veamos un ejemplo:

FIGURA 2.10 Ejemplo de una ontología



[Fuente: <http://www.cs.man.ac.uk/~horrocks/ISWC2003/Tutorial>, Datos Propios]

2.5.3.2 SINOPSIS DEL LENGUAJE OWL

Esta sección brindará un índice rápido de todas las características del lenguaje para OWL Lite, OWL DL y Full.

Es bueno aclarar, que existen términos cabalmente introducidos en OWL, así como owl:Class. También encontraremos prefijos de rdf: o rdfs: que son usados cuando los términos giran alrededor de RDF o RDF Schema, así como el termino rdfs:subPropertyOf.

➤ Sinopsis del OWL Lite

La lista de constructores del lenguaje OWL Lite se explica en la Tabla 2.2, dada a continuación:

TABLA 2.2 Características OWL Lite

<p>Características RDF Schema:</p> <p><i>Class (Thing, Nothing)</i> <i>rdfs:subClassOf</i> <i>rdf:Property</i> <i>rdfs:subPropertyOf</i> <i>rdfs:domain</i> <i>rdfs:range</i> <i>Individual</i></p>	<p>(Des)Igualdad:</p> <p><i>equivalentClass</i> <i>equivalentProperty</i> <i>sameAs</i> <i>differentFrom</i> <i>AllDifferent</i> <i>distinctMembers</i></p>	<p>Características de Propiedad:</p> <p><i>ObjectProperty</i> <i>DatatypeProperty</i> <i>inverseOf</i> <i>TransitiveProperty</i> <i>SymmetricProperty</i> <i>FunctionalProperty</i> <i>InverseFunctionalProperty</i></p>
--	---	---

Restricciones de Propiedad: <i>Restriction</i> <i>onProperty</i> <i>allValuesFrom</i> <i>someValuesFrom</i>	Restricciones de Cardinalidad: <i>minCardinality</i> (only 0 or 1) <i>maxCardinality</i> (only 0 or 1) <i>cardinality</i> (only 0 or 1)	Información de Cabecera: <i>Ontology</i> <i>imports</i>
Intersección de Clase: <i>intersectionOf</i>	Versionamiento: <i>versionInfo</i> <i>priorVersion</i> <i>backwardCompatibleWith</i> <i>incompatibleWith</i> <i>DeprecatedClass</i> <i>DeprecatedProperty</i>	Propiedades de Notas: <i>rdfs:label</i> <i>rdfs:comment</i> <i>rdfs:seeAlso</i> <i>rdfs:isDefinedBy</i> <i>AnnotationProperty</i> <i>OntologyProperty</i>
Tipos de Datos <i>xsd datatypes</i>		

[Fuente: "W3C <http://www.w3c.org/> (World Wide Web Consortium)]

➤ Sinopsis OWL DL y Full

La lista de constructores de abajo, Tabla 2.3, representa el agregado a la lista de arriba Tabla 2.2, para los lenguajes OWL DL y OWL Full.

TABLA 2.3 Agregación al OWL Lite para OWL DL y Full

Axiomas de Clase: <i>oneOf</i> , <i>dataRange</i> <i>disjointWith</i> <i>equivalentClass</i> (applied to class expressions) <i>rdfs:subClassOf</i> (applied to class expressions)	Combinaciones Booleanas de Expresiones de Clase: <i>unionOf</i> <i>complementOf</i> <i>intersectionOf</i>
Cardinalidad Arbitraria: <i>minCardinality</i> <i>maxCardinality</i> <i>cardinality</i>	Información Filler: <i>hasValue</i>

[Fuente: "W3C <http://www.w3c.org/> (World Wide Web Consortium)]

2.7 CONCEPTOS DE VISUALIZACIÓN.

2.7.1 VISUALIZACIÓN.

El concepto de visualización, según el diccionario de la Real Academia de la Lengua, hace referencia a la acción y efecto tanto de formar en la mente una imagen visual de un concepto abstracto, como de representar mediante imágenes ópticas fenómenos de otro carácter. Aplicado al contexto de la Visualización [Fuente: "Metodologías para el desarrollo de interfaces visuales de recuperación de información: análisis y comparación", Herrero Víctor Herrero y Yusef Hassan Grupo SC[Imago] de Información (VI), el término describiría tanto el hecho de generar una representación visual de un conjunto complejo de datos, como el fenómeno de percepción y comprensión de dicha representación por el usuario final.

La Visualización de la Información es un proceso de interiorización del conocimiento mediante la percepción de información, construcción mental que va más allá de la simple percepción sensorial [Fuente: *"Information visualisation, what is it all about?"*. *Inf@Vis! Revista digital de InfoVis.net*, Dürsteler, J.C. (2002)].

2.7.2. VIRI – INTERFACES VISUALES PARA LA RECUPERACIÓN DE LA INFORMACIÓN.

"Visual Interfaces for Information Retrieval", de acuerdo con Eick [Fuente: *"Visualizing on-line activity"* Eick, S.G. (2001). *Communication of the ACM*, 44(8), 45-52], es "el área de investigación enfocada a la creación de interfaces visualmente ricas para ayudar al usuario a comprender y navegar a través de espacios de información complejos".

En base a esta definición podemos distinguir claramente dos funciones de cualquier VIRI:

- Ayudar al usuario a comprender, proporcionándole una 'gran imagen general' de todo el conjunto documental.
- Ayudar al usuario a navegar, posibilitándole localizar y recuperar aquellos documentos que satisfagan sus necesidades de información a través de la exploración visual y la interacción con la interfaz.

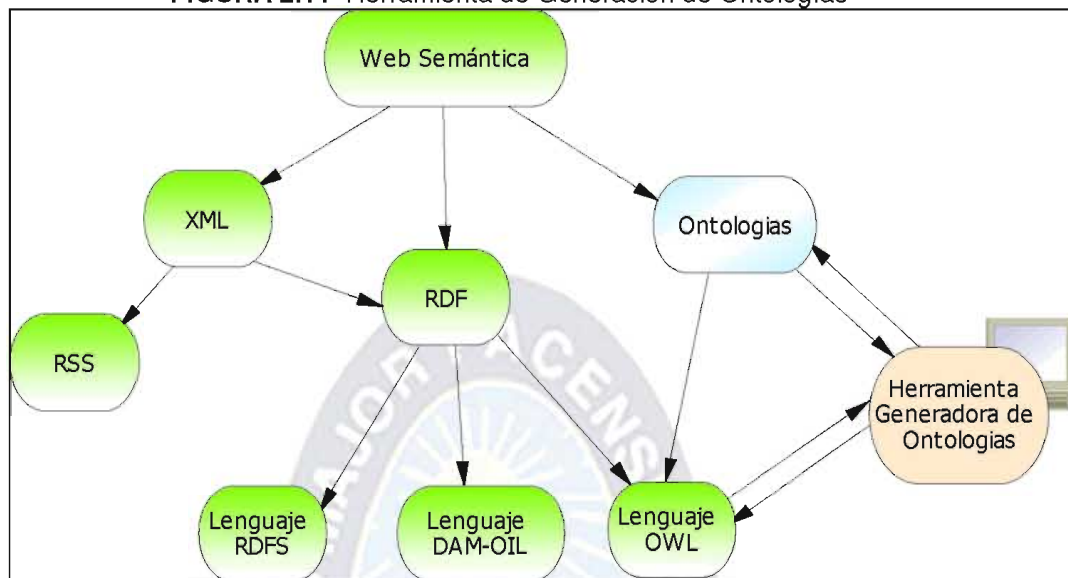
2.8 ¿COMO PARTICIPA LA HERRAMIENTA EN LA ETAPA DE DESARROLLO?

A grandes rasgos ya describimos de qué se trata una ontología. Ahora, teniendo en cuenta las implicaciones que conlleva su aplicación, es que, buscaremos la contribución a que esta tarea este lejos de complicarse.

La herramienta de desarrollo, aplicada a la representación de esquemas conceptuales, tales como son las ontologías, que conllevan el uso de un lenguaje, tal como OWL, principalmente se preocupa de la representación visual, atractiva, interactiva, para un manejo hábil dentro de sus esquemas y de los archivos en sí, los cuales se implementarán definitivamente en la página Web creada.

En una visión más amplia de lo que se pretende lograr, tenemos un esquema de cómo participará la herramienta en el desarrollo:

FIGURA 2.11 Herramienta de Generación de Ontologías



[Fuente: Elaboración propia]

Observamos como interactúa la herramienta entre la ontología y el lenguaje a utilizar para su representación, en este caso el OWL (Lenguaje de Ontologías Web).

Acercas del lenguaje **OWL**, podemos decir sigue en proceso de investigación, pero se considera un lenguaje completo, en comparación con los otros tantos lenguajes.

Como se había comentado una ontología no solo es utilizada en una Web Semántica, es así, que también puede trabajar independientemente del lenguaje OWL, con las mismas restricciones que presenta otros tipos de representaciones del conocimiento.

3 MARCO PRÁCTICO

A partir de la ejecución del método científico en la investigación, obtenemos, como la implementación de ontologías puede compararse con otras formas de representación del conocimiento, inmersas en áreas específicas del manejo de la información en comparación por ejemplo con los metadatos o por tesauros.

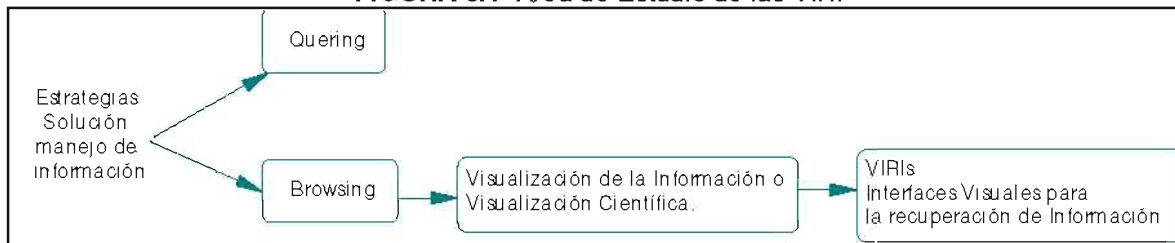
También, gracias al método científico es que tomamos nuestra perspectiva del problema, que es adquirir un manejo del problema desde el campo de trabajo llamada visualización de datos. Este último, los últimos años ha sido estudiado de forma práctica, pero esta abierta a nuevas propuestas de investigación.

Por otro lado presentamos aspectos adicionales que complementan la representación gráfica desde un enfoque de ayuda cognitiva.

3.1 DESCRIPCIÓN DE MÉTODOS.

La investigación en técnicas de recuperación de información ha abordado la problemática como el acceso, búsqueda, localización y recuperación de la información de grandes volúmenes de datos, dividiéndola para su estudio en dos grandes estrategias: el querying (interrogación) y el browsing (exploración), ver en Anexo1, una división más amplia. Dentro del browsing, que es el área que nos interesa, encontramos la Visualización de la Información y dentro de esta están los VIRIs (Interfaces para la recuperación de la Información). Ver Figura 3.1.

FIGURA 3.1 Área de Estudio de las VIRI



[Fuente: Elaboración propia]

Es así, que para el objetivo de realizar una VIRI o Interfase de Recuperación de la Información, existen muchos autores que ofrecen métodos bastante aplicables a las necesidades del objetivo de la Interfase.

Nuestra elección fue el método propuesto por el autor *Ben Fry* en su libro *“Visualizing Data”*, en una visión global de realización de todo el software, y tomamos otras técnicas ya en la fase de representación de los datos, que explicaremos en el capítulo 4.

Con tal aclaración, pasemos a describir brevemente en que consiste la metodología de Siete Niveles para la Visualización de Datos propuesta por Ben Fry [Fuente: *“Visualizing Data”* Fry, Ben, Capítulo 1].

3.1.1 SIETE NIVELES DE LA VISUALIZACIÓN DE DATOS.

Describimos a continuación de forma breve cada nivel:

➤ **ADQUISICIÓN**

Obtener los datos, desde un archivo en un disco un recurso sobre una red.

En nuestro caso la obtención de datos, puntualmente, el ingreso de datos se realiza con un proceso interactivo, pero la adquisición de estos es a través de formularios insertos en ventanas o frames (desde una terminología Java).

➤ **CLASIFICACIÓN.**

Proveer alguna estructura para darle significado a los datos, y orden dentro de sus categorías.

En nuestro caso, en la adquisición de datos notamos claramente que la clasificación de los datos esta hecha en el momento del ingreso, es un proceso paralelo.

Esto a causa de que cada pequeño formulario es distinto de acuerdo ha:

- Los elementos ya existentes en el almacén de conocimientos temporal. Este es representado en realidad por una estructura de tipo lista enlazada, y una estructura propia de algunos lenguajes, como es el Hashtable (tabla Hash).

- El tipo de elemento que se quiere introducir: clases, propiedades, instancias, y cada dato que implica cada una de ellas. Aquí también se mencionan las validaciones, en el caso de datos de tipo String (cadena).
- Las restricciones dadas por el mismo lenguaje OWL, estudiadas en el capítulo 2.

Terminado este ingreso de datos, no existe alguna modificación posterior por parte del programa que implique una reclasificación de los datos, puesto que el objetivo mismo de nuestro programa es el de reflejar en el documento final exactamente lo que se ingreso vía Interfase de Visualización.

➤ **FILTRADO.**

Se trata de remover los datos que no son de interés.

Se propone, posteriormente, una filtración de datos visuales, a causa de atacar el problema de saturación de datos en pantalla.

➤ **MINERÍA O EXPLOTACIÓN.**

Es aplicar métodos desde estadísticos o minería de datos como una forma de discernir principales o colocar los datos en un contexto matemático.

En nuestro caso, esta labor al igual que las ya mencionadas se las realiza en la misma introducción de datos (edición). Podemos referirnos puntualmente de la validación de acuerdo a los elementos ya ingresados, por ejemplo, que los nombres sean únicos para todas las clases.

➤ **REPRESENTACIÓN Y REFINACIÓN.**

Representar quiere decir escoger el modelo visual básico así como un grafo, lista o árbol.

Refinación es improvisar la representación gráfica para hacerla mas limpia y visualmente mas entendible. Es evidente que existen varias técnicas para la graficación de la información (Ver “Lista de Listado de técnicas de visualización en orden alfabético” en Anexos), así que debemos hacer un análisis, para entender cual de estas técnicas obedece a las exigencias de nuestro objetivo final, el de realizar un programa para la edición gráfica de las ontologías.

En el siguiente subtítulo 3.1.2, se precisa la forma como fue escogida la técnica de visualización para esta parte.

➤ **INTERACCIÓN.**

Adicionar métodos para manipulación de datos o controlar que las características sean visibles.

Esta etapa demuestra el proceso en sí del control que realiza el usuario con la gráfica y sus datos. Se refiere exactamente a entrada y salida de datos.

Esta etapa, vamos a remarcarla en el capítulo 4, ya que es un aspecto que muchos otros trabajos dejan de lado y no cubren las expectativas reales, en pro de mejorar la calidad del software.

3.1.2 ELECCIÓN DE LA TÉCNICA DE VISUALIZACIÓN A UTILIZAR.

Del conjunto de técnicas de visualización, el autor Lin, podemos extraer la siguiente clasificación entre lo que llama metáforas visuales: jerárquicas, de redes, de dispersión y mapas [Fuente: "Map displays for information retrieval. *Journal of the American Society for Information Science*" Lin, X. (1997), 48(1), 40-54].

i) JERÁRQUICAS:

Este tipo de representación visual, donde los elementos se presentan en diferentes niveles, ramas o agrupaciones, que descienden de un nodo raíz, es la más común cuando la propia naturaleza del conjunto de datos a visualizar es jerárquica, como por ejemplo en la visualización de estructuras complejas de directorios y ficheros de sistemas informáticos.

ii) REPRESENTACIONES DE REDES:

Las representaciones de redes son aquellas donde los diferentes elementos documentos, términos son presentados en forma de nodos o vértices, mientras que la estructura semántica se encuentra definida por los enlaces o arcos que conectan dichos nodos.

iii) REPRESENTACIONES DE DISPERSIÓN:

Una forma alternativa de representar visualmente estructuras de datos multidimensionales es en forma de nubes de puntos o nubes de dispersión. Estos puntos que visualmente no tienen por qué tener dicha forma, ya que podrían ser presentados como iconos o como rótulos de texto - estarían distribuidos en un espacio visual bidimensional o tridimensional, y distanciados unos de otros en función de las disimilaridades o distancias originales especificadas en la matriz de similaridad.

iv) MAPAS:

Las representaciones visuales basadas en mapas se fundamentan en la idea de utilizar la metáfora de mapa geográfico para la visualización de espacios de información. Por lo general, el objetivo de la utilización de Metodologías para el desarrollo de Interfaces Visuales de Recuperación de cualquier tipo de metáfora visual en el diseño de interfaces es hacer visible para el usuario la estructura y relaciones en un conjunto determinado de datos. Por tanto, la idea de utilizar estas metáforas para la visualización de espacios de información complejos y abstractos parece tener bastante sentido, ya que brindan una visión diferente del conjunto que en la mayoría de los casos enriquecerá la imagen mental previa que el usuario tenga de él.

Es muy acertado incluir nuestro programa a la metáfora de ii) Representaciones de Redes. En otros libros encontramos que pueda pertenecer simplemente a una clasificación de graficación de grafos, pero es un hecho, que los grafos son la base para la representación del conocimiento en un ámbito general.

Ahora, teniendo en cuenta lo anterior, apuntemos que existen técnicas para hacer una refinación en los datos, tales como: clustering, escalamiento de red, técnicas de reducción de la dimensión, etc.

3.1.2.1 ESTUDIO DE LOS MÉTODOS.

Todas las técnicas deben ser usadas en el marco de los requerimientos, es así, que si dos técnicas se pueden emplear para la realización, cosa que no es muy usual, notablemente una nos dará mejor resultado que otra. Tal vez al hacer este trabajo de reducción, es que se observa una notoria diferencia entre estas técnicas, ya que algunas tienden a perder sus datos originales en proporción mayor a otras, claro esta dependiendo también de los escenarios en los cuales se manejan.

Estas técnicas (Ver Tabla 3.1), tienen en común el objetivo de la reducción de datos, esto a raíz de que los datos que se van a visualizar están en cantidades grandes, es decir que deben localizarse en distintos puntos de nuestra pantalla, entonces puede haber una saturación en grado muy alto, y por tanto va en contra de los objetivos mismos de una representación de conocimiento en general.

Es así, que todas estas técnicas tienen la labor de reducción de datos multidimensionales, es decir que tratan de implementar bajo un esquema 2D (2 dimensiones) y 3D (3 dimensiones), los cuales sean comprensibles por el ojo humano, pero existe una

especialidad acerca de los datos que se preservan una vez aplicado el método (Ver Tabla 3.1), y por defecto los datos que se pierden, a esto se le llama fidelidad.

TABLA 3.1 Comparación de Métodos

<i>Técnica</i>	<i>Elementos que preserva de la estructura</i>
MDS	Distancia entre elementos
Clustering	Relaciones de agrupación
Pathfinder	Relaciones locales mas fuertes
SOM	Relaciones de vecindad

[Fuente: "Metodologías para el desarrollo de interfaces visuales de recuperación de información: análisis y comparación", Herrero, V., Hassan Y. Grupo SCImago]

3.1.2.2 CONCLUSION DEL ESTUDIO.

En el caso de nuestro programa, claramente notamos que, una técnica de reducción de datos no va a solucionar el problema de saturación del esquema que buscamos representar.

Sin embargo, presentamos la solución por medio de división del esquema, en varios esquemas visuales, por medio de filtrado de datos. Dicha solución se explica con mas detenimiento en el capítulo 4.

Nuestra visión de los esquemas a representar, evidentemente nos llevan a concluir que estamos tratando con una representación de redes conceptuales, y consecuentemente de grafos dirigidos, ya que encontramos nodos, representados por los diferentes componentes, como son las clases, instancias, y por otro lado observamos los arcos como los componentes de tipo Propiedad, los cuales enlazan a los componentes ya mencionados. Entonces para el efecto, y dados los principales requerimientos, hemos seleccionado una herramienta que cumple, de manera eficiente, y que nos servirá para el éxito de nuestro programa. Esta herramienta es la librería ZVTM, la cual es desglosada mas adelante.

Cabe resaltar también el apoyo de la librería ZVTM, en java y que cubre de manera eficiente el apoyo de las características que deseamos en nuestro programa.

3.2 MÉTODO CIENTÍFICO.

La utilización del método científico pone en relieve la investigación del presente trabajo de investigación, enfocando para ello los siguientes pasos importantes que involucran el desarrollo de la herramienta:

- Problema
- Identificación de variables

- Hipótesis
- Comprobación de hipótesis
- Resultados
- Conclusiones

En cuanto a los métodos involucrados para el desarrollo, mencionaremos:

3.2.1 MÉTODO ANALÓGICO-SINTÁCTICO.

Para la lectura sintáctica del lenguaje y por tanto hacer posible la inferencia en la ontología que este representa, es imprescindible realizar un análisis jerárquico, propio de la interpretación de redes conceptuales, y así obtener el equivalente solicitado.

3.2.2 MÉTODO INDUCTIVO-DEDUCTIVO.

Se utiliza este método, para la diferenciación de las representaciones gráficas hacia el lenguaje OWL, cada una individual en su representación, hace que esta labor deba inmiscuirse en los detalles necesarios, inclusive incorporando algunos, para que tenga la validez que el lenguaje exige, y de esta manera ayudar a la transformación de dicha representación a su equivalente en lenguaje OWL.

3.2.3 MÉTODO ABSTRACCIÓN-CONCRECIÓN.

En el proceso inverso al mencionado anteriormente, a partir de un código elaborado en lenguaje OWL, se realiza una generalización de las sentencias, dejando de lado algunos detalles que solo le conciernen al lenguaje. Así, se engloba las divisiones entre conceptos, haciendo posible su representación conceptual gráfica de forma más general.

3.3 DESCRIPCIÓN DE COMPORTAMIENTO.

Para la realización del prototipo se hizo una selección detallada de las herramientas a emplear.

3.3.1 DESCRIPCIÓN DE TECNOLOGÍAS.

Describiremos dos tecnologías base en la construcción del prototipo: Jena e Isaviz, pero resaltaremos tareas de mejora en la última citada.

3.3.1.1 JENA.

Jena es software de Licencia Open Source, habilitada en su sitio oficial [<http://jena.sourceforge.net/>].

Jena es un API para Java, es decir, un entorno para el desarrollo de aplicaciones en el lenguaje de programación Java para la web semántica.

Framework desarrollado por HB Labs para manipular metadatos desde una aplicación Java.

En la actualidad existen dos versiones:

- Jena 1:
Principalmente soporte para RDF.
Capacidades de razonamiento limitadas.
- Jena 2:
Incluye además una API para el manejo de ontologías.
Soporta el lenguaje OWL.

En nuestro caso por las capacidades de inferencia explicadas mas adelante, utilizamos Jena en su versión 2.

CARACTERÍSTICAS DE JENA.

Jena permite gestionar todo tipo de ontologías (añadir hechos, borrarlos y editarlos), almacenarlas y realizar consultas contra ellas. Soporta RDF, DAML y OWL y es independiente del lenguaje. Los recursos no están ligados estáticamente a una clase java particular.

Los principales problemas que presenta son:

- El recurso no cambió, pero la clase Java para modelarlo es otra.
- No podemos modificar la clase dinámicamente.
- Jena considera que la abstracción Java del recurso es sólo una vista del mismo.

Incluye varios componentes:

- ARP: un parser de RDF
- API RDF
- API de Ontologías para OWL, DAML y RDF Schema.

En particular, tiene soportes para los tres sublenguajes del OWL, los cuales son OWL Lite, OWL DL y OWL Full.

- Subsistema de razonamiento
- Soporte para persistencia
- RDQL: Lenguaje de consultas de RDF

- Validador de OWL.

Existe la posibilidad de realizar una validación básica de OWL. Esta validación sólo comprueba la sintaxis, no infiere ni razona. Para validaciones más complejas existe Jena 2, que ofrece soporte para inferencias y detecta la violación de las restricciones definidas en el schema por las instancias.

- Inferencias con Jena.

El módulo de Jena, Inference Support, realiza la tarea de inferir, por eso es llamado razonador (Reasoner).

Jena ofrece mecanismos para añadir nuevos razonadores e incluye un conjunto básico de éstos:

- OWL Reasoner
- DAML Reasoner
- RDF Rule Reasoner
- Generic Rule Reasoner

Para hacer inferencias debemos crear un modelo inferido a partir de un razonador y a partir de ahí, todas las consultas que se le hagan al modelo inferido devolverán información inferida.

3.3.1.2 ZVTM.

Creado en un proyecto encabezado por Emmanuel Pietriga, con licencia GNU. Surge a raíz, de que si bien el Swing de Java cumple los requerimientos de portabilidad, eficiente, de uso general; su utilización requiere una programación de bajo nivel, es decir bastante minucioso.

Es así, que surge ZVTM, como una Herramienta de Interfase de Usuario Zoomable, especializado en el área de edición en contextos visuales, apoyado en objetos Glyph, los cuales son bastante comunes en el área de Graficación de grafos o redes.

Otra característica bastante importante: Movilidad y Zoom. Ambas son cruciales en nuestro editor, y el ZVTM lo realiza mediante una técnica de cámaras, que dada una capa visual o espacio de trabajo, puede hacer una vista especial controlada por el usuario, o por el programa.

Cabe resaltar, que si bien esta técnica, bastante poderosa de las cámaras, que hacen posible una movilidad o zoom del programa, no nos da la solución completa para el objetivo de funcionalidad especial en la representación visual, nos proporciona una parte importante.

Recordando un poco, la cantidad de datos a mostrar, es nuestra principal enemiga, así que nuestra propuesta de solución, se explicará en el capítulo 4.

3.3.1.3 PROTEGE-OWL 3.2.1.

El proyecto Protege inicialmente fue abordada por la Universidad de Standford, ahora es un proyecto donde trabajan entes universitarios, gubernamentales, y otras empresas. Escrito en Java, es el preferido en cuanto a herramientas de edición de ontologías, y por lo mismo tiene muchos plugins. No obstante, el plugin Protege-OWL, es el que realmente abarca la representación mediante el lenguaje OWL, tal cual es el objetivo del presente trabajo, pero, la modalidad de este plugin es de introducción de datos por medio de los paneles de edición, llamados widgets. Es así, que no cuenta con el respaldo visual, necesario, pero lo importante es que presenta un software adaptable a otras implementaciones.

3.3.2 INTERACCIÓN ENTRE COMPONENTES.

Como habíamos descrito anteriormente, la Visualización de datos es tomado con la metodología de los siete niveles, es así que encontramos las siguientes tareas puntuales:

- Elección de una base de interfaz potente, capaz de hacer la tarea de mediadora entre las herramientas escogidas.
- Adecuación de funciones de graficación a través de la tecnología GraphViz y las funciones a necesitar.
- Implantación de las funciones de Jena, en cuanto a razonamiento Semántico, inferencia.
- Implementación del módulo de ayuda con la participación de las tecnologías ya planteadas.

Es así, que describimos una visión general de los componentes de la herramienta.

FIGURA 3.2 Diagrama de Componentes



[Fuente: Elaboración Propia]

4 PROCESO DE INVESTIGACIÓN.

Para la formalización en un nivel más particular de la herramienta visual de edición de ontologías, la cual toma se ha denominado Visod (Visual Ontology Diagram, Diagramación de Ontologías Visual), la explicación refleja el proceso de elaboración de una ontología.

4.1 CONSTRUCCIÓN DE UNA ONTOLOGIA POR PASOS.

Para observar el proceso de la modelación de nuestra ontología, tomaremos una serie de pasos sugeridos en el texto “Guía Para Crear Tu Primera Ontología” [Fuente: “Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología” Noy, Natalya y McGuinness, Deborah traducido por: Erick Antezana, Pag. 10].

Para esta explicación, con fines netamente explicativos elaboraremos una pequeña ontología como ejemplo, tomado de la sección “Institucional” del espacio dedicado a la Facultad de Ciencias Puras y Naturales de la página web de la Universidad Mayor de San Andrés desarrollada por UmsaTic, la dirección url es <http://fcpn.umsa.bo/fcpn/app>. Ver Figura 4.1.

FIGURA 4.1 Página Web de la Facultad de Ciencias Puras y Naturales



[Fuente: Elaboración propia, extraído de <http://fcpn.umsa.bo/fcpn/app>]

PASO 1: DOMINIO.

El dominio o alcance, similar al dominio que tomamos para el diseño de una base de datos, debe ayudar a concentrar la atención en un contexto, el cual, si bien puede ser modificado, es crucial para el objetivo que perseguimos.

Las siguientes preguntas, nos ayudan a definir nuestro dominio:

- ¿Cuál es el dominio que la ontología cubriría?
- ¿Para qué usaremos la ontología?
- ¿Para qué tipos de preguntas la información en la ontología debería proveer respuestas?
- ¿Quién usará y mantendrá la ontología?

En nuestro ejemplo; realizado el estudio, obtenemos las siguientes respuestas:

1. ¿Cuál es el dominio que la ontología cubrirá?

R.- La sección Institucional de la página web de la Facultad de Ciencias Puras y Naturales.

Nos limitamos a describir los conceptos relacionados a esta sección, no así a los enlaces con los cuales se encadena.

Dentro de estos conceptos, solo tocamos los referentes a la Facultad, y relaciones claras con otros conceptos como Carrera o Universidad.

2. ¿Para qué usaremos la ontología?

R.- Para la información de la organización institucional dentro de la Facultad de Ciencias Puras y Naturales. Esta información es dirigida a los estudiantes en primera instancia, y al público en general.

Es improbable que la información tenga carácter referencial, en cuanto a datos de ubicación o nombres de los miembros de la organización.

3. ¿Para qué tipos de preguntas la información en la ontología debería proveer respuestas?

R.- Las preguntas son las siguientes:

- ¿Cómo se estructura la organización de tipo institucional en la Facultad de Ciencias Puras y Naturales?
- ¿Cómo se relacionan los componentes de dicha organización?
- ¿Cuáles son las dependencias entre unidades?
- ¿Cuáles son las autoridades posesionadas actualmente?
- ¿Quién es la máxima autoridad de la Facultad de Ciencias Puras y Naturales?

4. ¿Quien usará y mantendrá la ontología?

R.- Los administradores, y técnicos informáticos encargados del mantenimiento de la página web de la Facultad de Ciencias Puras y Naturales de la Universidad Mayor de San Andrés.

PASO 2: CONSIDERAR LA REUTILIZACION DE ONTOLOGIAS.

La instrucción de este paso es realizar una reutilización de otras ontologías que tengan una relación con la que estamos desarrollando, o que se busque la conexión con tecnologías que hacen uso de aquellas ontologías a reutilizar.

Vale la pena mencionar, que existen ontologías agrupadas en grandes bibliotecas, tal es el caso de Ontolingua (<http://www.ksl.stanford.edu/software/ontolingua/>) o la biblioteca de ontologías DAML (<http://www.daml.org/ontologies/>). También hay un cierto número de ontologías comerciales públicamente disponibles (ejemplo, UNSPSC (<http://www.unspsc.org>), RosettaNet (<http://www.rosettanet.org>), DMOZ (<http://www.dmoz.org>)).

En nuestro ejemplo: No consideramos la reutilización de otra ontología, ya que, no existen antecedentes con alguna relación en el tema propuesto.

PASO 3: TERMINOS IMPORTANTES PARA LA ONTOLOGIA.

Como ya tenemos, entre otras cosas, el objetivo de nuestra ontología, entonces ya sabemos el “**Que**” queremos decir. Así pues, enumeramos los términos referentes a los temas que vamos a tocar, de forma general, sin distinguir entre que término puede ser clase o propiedad.

En nuestro ejemplo: nos avocamos a los términos que se utilizan solo en la sección Institucional de la página web, como ser: facultad, estructura orgánica, etc. También se incluyen algunos términos, que ya están implícitos, como por ejemplo el término Universidad, que si bien se nombra algún concepto o definición, sabemos que la facultad pertenece a la universidad.

A continuación se tiene una lista de los términos seleccionados en una primera inspección.

TABLA 4.1 Términos Importantes Ejemplo de Ontología.

<i>Posibles clases</i>	<i>Posibles propiedades</i>
Institucionalidad	unidad, dependencia, coordinación, supervisa, académico, científico.
Universidad	académico, pagina web
Facultad	Carrera, estudiante, docente, administrativo, auxiliar de docencia, Dirección, Teléfono.
Estructura orgánica	Decano, Vicedecano, Área desconcentrada, Carrera, Honorable consejo facultativo, Centro, Jefatura, secretaria, Kardex, académico, Area desconcentrada.
Autoridad	Persona, Reglamento, Reglamento
Carrera	Física, Matemática, informática, biología, estadística, química, estudiante

[Fuente: Elaboración propia]

PASO 4: CLASES.

Se pueden mencionar los enfoques propuestos por Uschold y Gruninger en 1996. Estos son:

- **Desarrollo top-down**, comienza con la definición de los conceptos más generales en el dominio la subsecuente especialización de los conceptos.
- **Desarrollo bottom-up** comienza con la definición de las clases mas específicas, las hojas de la jerarquía, con el subsecuente agrupamiento de esas clases en conceptos más generales.
- **Desarrollo combinado** es el resultado de una combinación de los enfoques top-down y bottom-up: primero definimos los conceptos más sobresalientes y luego los generalizamos y especializamos apropiadamente.

En nuestro ejemplo: usamos una jerarquía top-down, de arriba hacia abajo. Es decir, buscamos en la lista de términos del Paso 3, los términos más generales, los términos que por si solos representan un concepto sin depender de la existencia de otro, haciendo a un lado por un momento sus propiedades o características individuales. A continuación vemos los conceptos más generales:

- Organización.
- Facultad
- Carrera

- Persona
- Función
- Autoridad

Posteriormente las propiedades van siendo adjuntadas con el respectivo concepto general.

JERARQUIA DE CLASES.

Al realizar este trabajo es muy importante hacernos la siguiente pregunta:

“Si una clase A es una superclase de la clase B, entonces cada instancia de B lo es también de A”. [Fuente: *Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología*, Natalya F. Noy and Deborah L. McGuinness, traducido del inglés por: Erick Antezana, Pag. 10].

En nuestro ejemplo: observemos que, un funcionario es una persona, de forma inequívoca, viendo por el lado inverso, una persona puede ser o no, un funcionario, es decir, no es segura esta relación inversa.

Tratando la información observamos el siguiente cuadro.

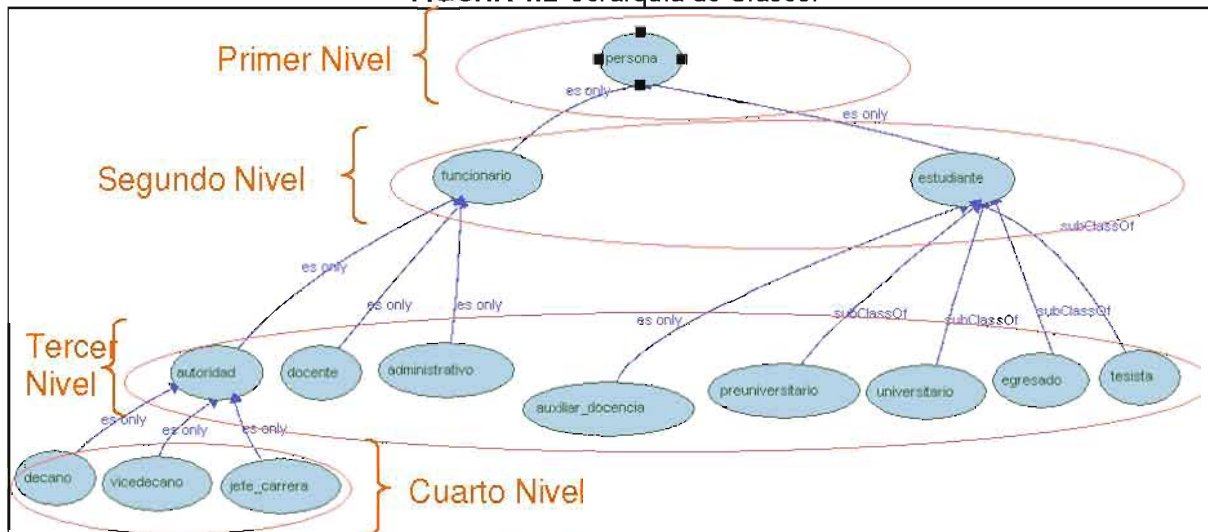
TABLA 4.2 Jerarquía de Clases, Ejemplo de Ontología

Persona	Funcionario	Autoridad	Decano
			Vicedecano
		Docente	
		Administrativo	
	Estudiante	Grado	Preuniversitario
			Universitario
			Egresado
			Testista
			Auxiliar Docencia

[Fuente: *Elaboración propia*]

Esta deducción de jerarquías, se denotan mediante el esquema siguiente, el cual, como se observa es básicamente la estructura de clases que se conoce. La relación básica que se utiliza, es la de **subClaseDe (subClass Of)**, la misma que representa las jerarquías de paternidad.

FIGURA 4.2 Jerarquía de Clases.



[Fuente: Elaboración propia]

PASO 5: DEFINIR LAS PROPIEDADES DE LAS CLASES.

Los conceptos formados ya en clases jerárquicas, necesitan ser más detallados, y relacionarse entre sí. Esa tarea es posible con los términos que aún no hemos utilizado.

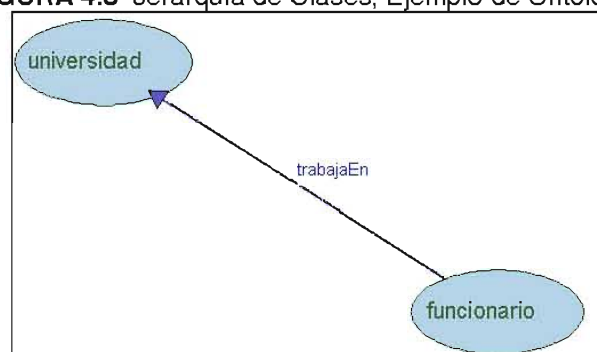
Por ejemplo, una autoridad tiene un **nombre (nombre, apellido)**, que en nuestro caso es importante mencionar. También podemos mencionar las relaciones como la de dependencia en las Unidades, es decir, que una unidad tiene otra unidad de la cual es dependiente, como por ejemplo, la unidad de Vicedecanatura **es dependiente** de la unidad de Decanatura.

Es así que podemos mencionar los tipos de propiedades [Fuente: *Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología, Natalya F. Noy and Deborah L. McGuinness, traducido del inglés por: Erick Antezana, Pag. 10*]:

- Propiedades “intrínsecas” tales como **es dependiente**;
- Propiedades “extrínsecas” tales como el **nombre** de una autoridad;
- partes, si el objeto es estructurado; pueden ser “partes” físicas y abstractas.
- relaciones con otros individuos; estas son las relaciones entre miembros individuales de una clase y otros ítems.

En nuestro ejemplo; observemos que, el concepto funcionario puede ser tomado como un funcionario de cualquier institución, puede ser pública, privada, talvez de la alcaldía, en fin, es ambiguo, pero si tomamos el concepto universidad, y lo relacionamos con la propiedad **trabajaEn**, entonces, ya podemos especificar la idea, que es, **funcionario trabajaEn universidad**, y entonces sabremos que se trata de cierto tipo de funcionarios. Al respecto ver siguiente figura.

FIGURA 4.3 Jerarquía de Clases, Ejemplo de Ontología.



[Fuente: Elaboración propia]

PASO 6: DEFINIR FACETAS DE LOS SLOTS (PROPIEDADES).

Cuando nos referimos a slots o propiedades, inevitablemente existen características enlazadas con ellos, tal es el caso de: cardinalidad, tipo de dato. Estos refieren a restricciones del tipo de valores se aceptaran, en todo caso es una funcionalidad posible con el OWL.

Entre las características mas importantes están:

- Cardinalidad del slot.

La cardinalidad de un slot define cuantos valores puede tener dicho slot. Algunos sistemas, solamente distinguen entre cardinalidad simple (admitiendo a lo sumo un valor) y cardinalidad múltiple (admitiendo cualquier cantidad de valores).

- Tipo de valor de los slots.

Una faceta tipo de valor describe qué tipos de valores pueden llenar el slot. Aquí está una lista de los tipos de valores más comunes:

- String es el tipo de valor más simple el cual es usado por slots tales como nombre: el valor es una simple cadena de caracteres. En nuestro caso el nombre de una autoridad es de tipo String.

- Number (algunas veces los tipos de valores Float e Integer son usados por ser más específicos) describe slots con valores numéricos.

- Los slots del tipo Boolean son simples banderas si/no.

Puede ser representado como un valor de un slot Boolean: si el valor es "true" ("si") el vino es espumante y si el valor es "false" ("no") el vino no es espumante.

- Los slots del tipo Enumerated especifican una lista específica de valores admitidos para el slot.

• Los slots del tipo Instance (Instancia) admiten la definición de relaciones entre individuos. Los slots con tipo de valor Instance deben también definir una lista de clases admitidas de las cuales las instancias pueden provenir.

- Dominio y rango de un slot

Las clases admitidas para los slots de tipo Instance son a menudo llamadas rango de un slot.

En nuestro ejemplo: las definiciones de las facetas de los datos referentes a estudiantes, se observan en la siguiente tabla:

TABLA 4.3 Facetas de los Slots (propiedades) de la Clase Estudiante.

Slot	Facetas	
	Tipo	Dominio
Nombres	String	A-Za-z
Apellidos	String	A-Za-z
Año de Ingreso	Number	>1950
Número de Carnet de Identidad	Number	>1

[Fuente: Elaboración propia]

PASO 7: CREAR INSTANCIAS.

El último paso consiste en crear instancias individuales de clases en la jerarquía. La definición de una instancia individual de una clase requiere (1) elegir una clase, (2) crear una instancia individual de la clase y (3) rellenar los valores del slot.

En nuestro ejemplo: las instancias que corresponden son las autoridades actuales, que se distinguen por su nombre, de acuerdo al siguiente cuadro:

TABLA 4.4 Ejemplo Instancias

Clase	Instancia
Decano	Nombre: Lic. M.Sc. Franz Cuevas Quiroz
ViceDecano	Nombre: Lic. Luis Morales Escobar

[Fuente: Elaboración propia]

4.2 HERRAMIENTAS QUE INTERVIENEN EN LA ELABORACIÓN DEL SOFTWARE.

Tuvimos una mirada breve acerca de los pasos que se deben cumplir al elaborar una Ontología, básica, con recomendaciones en un nivel bastante práctico, pero aclaramos que se pueden tomar otras metodologías, ya que incluso se observan estudios al respecto. Esto simplemente para expresar que es un campo abierto a la investigación.

En la figura 3.2 observamos un cuadro de componentes, destacándose los módulos siguientes: Soporte para proyectos e Interfase de Usuario, Almacenamiento de Datos Sintácticos y Validación de Datos, Graficador, Analizador Semántico.

Es así que, explicaremos brevemente como se ha trabajado para la obtención de estos módulos, poniéndose énfasis en la Graficación.

4.2.1 SOPORTE PARA PROYECTOS E INTERFACE DE USUARIO.

El objetivo de realizar un diseño es un proceso largo y moroso, es así que la forma de llevar este proceso se determina como un proyecto.

Es así que precisamos la gestión de este proyecto de diseño de ontología con el soporte de **Protege 3.3**. Es una librería y un proyecto de investigación de carácter libre, que colabora con las implementaciones fáciles de interfase con el usuario a través de menús, y las funciones adjuntas se separan en pestañas.

Por otro lado estas librerías tienen la capacidad de insertar diferentes plugins, es decir que es un software adaptable a las necesidades del diseñador.

Resaltamos que se trabajó adaptando dicho soporte en algunos aspectos, tales como la inclusión de clases para colaborar con el diseño ergonómico de su interfaz de usuario por medio de la presentación de Estilos de Interfase de Usuario como se llama en el entorno Java Look & Feel.

Utilizamos este soporte estándar base, dándonos como resultado que cada proyecto conlleve archivos de extensión “.pprj”.

4.2.1.1 PANELES DE EDICIÓN.

En esta parte debemos resaltar, que los paneles de edición, propuestos por **Protege-OWL 3.2.1**, son enteramente reutilizables, y eficientes, ya que no existe ningún de problema respecto su interacción con las clases validadoras.

Es así que estas clases fueron adaptadas a los requerimientos que la inserción de objetos visuales de forma dinámica, lo requería. Vemos los paneles de edición, en las figuras mostradas en el punto 4.3.

4.2.2 ALMACENAMIENTO Y VALIDACIÓN DE DATOS SINTÁCTICOS.

Contamos con una estructura que básicamente almacene los datos de cada objeto de la ontología, y que por tanto interactúe dinámicamente con las acciones de edición de dichos objetos, especificados en sentencias, es decir de cada elemento forma parte de la estructura misma de la ontología.

Posteriormente estos datos almacenados son la base para la posterior interacción con el clasificador, elaborador de la ontología. Este, debe tener un soporte para los componentes que vamos a manejar, como son: clases, propiedades, instancias, expuestos en el capítulo 2.

Es así, que como primera medida tomamos la estructura de el paquete *Protege-OWL 3.2.1*. Este software posee un soporte extraordinario, y bastante eficiente. Es de carácter libre.

4.2.2.1 Validación de Datos en la edición de los Componentes.

Es imprescindible aclarar que existen equipos de investigación encargados de realizar la labor de herramientas que permiten manipular los elementos de lenguajes como es el OWL.

En este caso *JENA*, es la librería en Java, apta para realizar esta tarea, y de carácter libre.

De lo ya expuesto, vemos que nuestro almacén de datos temporal, soportado por librerías de *Protege-OWL 3.2.1*, son los que interactúan directamente con las clases de *JENA*.

Esta combinación de librerías, hacen dinámica la validación, clasificación, pero aunque la labor es amplia, aquí no concluye el trabajo de *JENA*.

4.2.3 GRAFICADOR.

La graficación o representación de datos visuales, es un área que ha adquirido gran trascendencia en los últimos años. Prueba clara son las implementaciones de tecnologías como los GIS (Sistema de Información Geo-referencial).

El principal objetivo de dichos esquemas es facilitar al usuario la comprensión de datos que son de carácter abstracto, tal como se explicó en el capítulo 2.

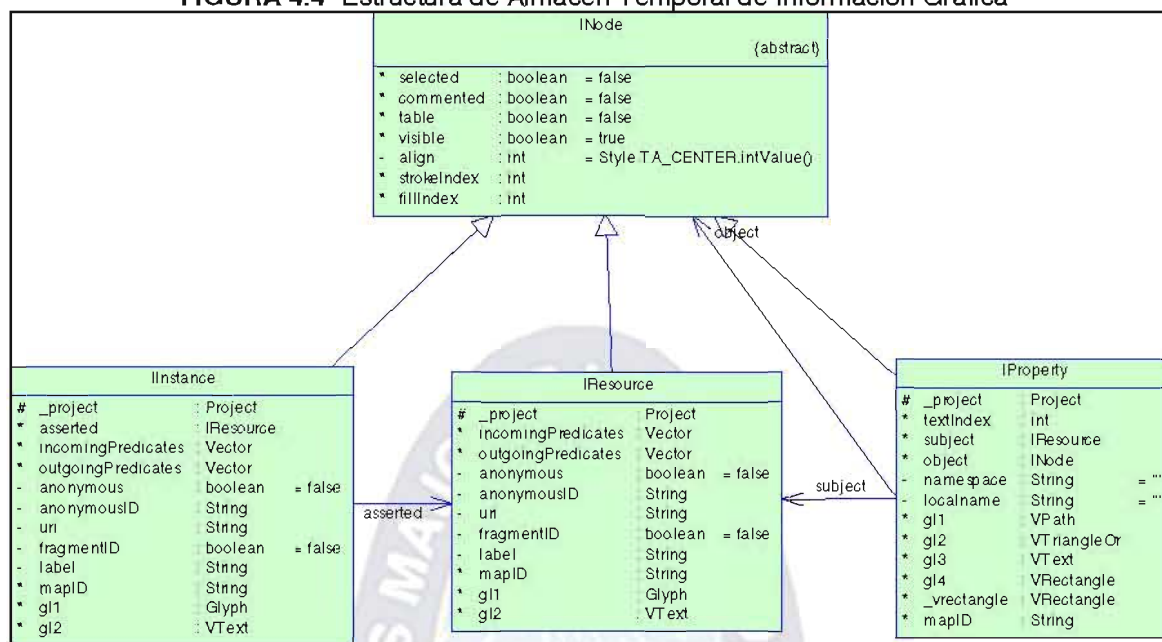
Es así que, siguiendo los pasos para la construcción de una VIRI estudiada en el capítulo 3, nos encontramos con los siguientes componentes:

4.2.3.1 ALMACÉN DE INFORMACIÓN GRÁFICA.

A pesar de la robustez de *Protege*, notemos, que sobre esta estructura debemos manejar datos enteramente de tipo sintáctico, y no es recomendable, combinarlos con los datos de tipo gráfico.

Es así, que optamos por el segundo almacén de la estructura de los gráficos tomamos la jerarquía de clases siguiente:

FIGURA 4.4 Estructura de Almacén Temporal de Información Gráfica



[Fuente: Elaboración propia]

En esta estructura destacamos los atributos correspondientes a la clasificación básica para una ontología, como son:

- Clases (representado por la clase IResource).
- Propiedades (representado por la clase Iproperty).
- Instancias (representado por la clase IInstance).

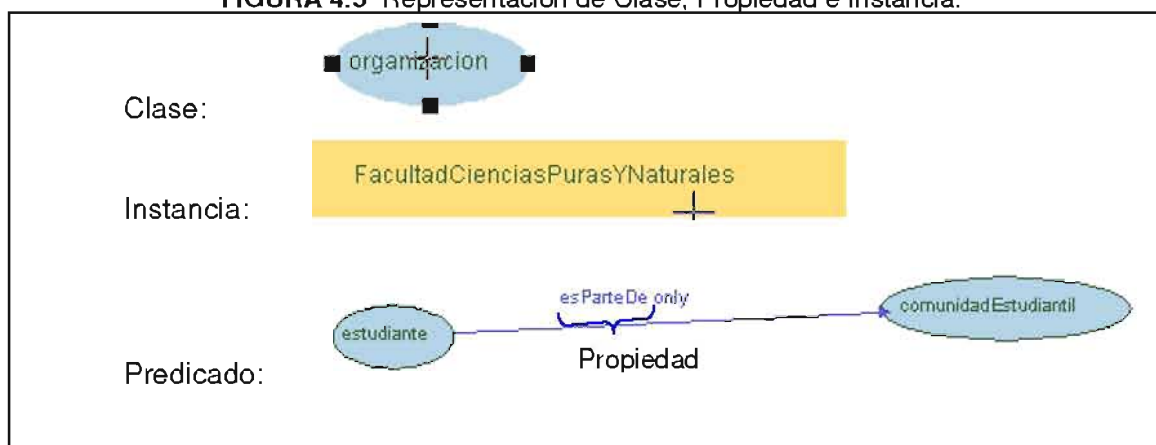
Las clases IInstance, Iresource, Iproperty, contienen campos que almacenan datos visuales, como Vrectangle, Vtext, Glyph. Todos orientados a las clases de la librería ZVTM, explicada a continuación.

4.2.3.2 REPRESENTACIÓN DE LOS COMPONENTES CLASE, PROPIEDAD, INSTANCIA Y PREDICADO.

Requerimos la representación gráfica de cada uno de estos componentes.

Mediante los Glyphs, los cuales son clases ya definidas de la *librería ZVTM*, le asignamos a cada uno de nuestros componentes una representación visual.

FIGURA 4.5 Representación de Clase, Propiedad e Instancia.



[Fuente: Elaboración propia]

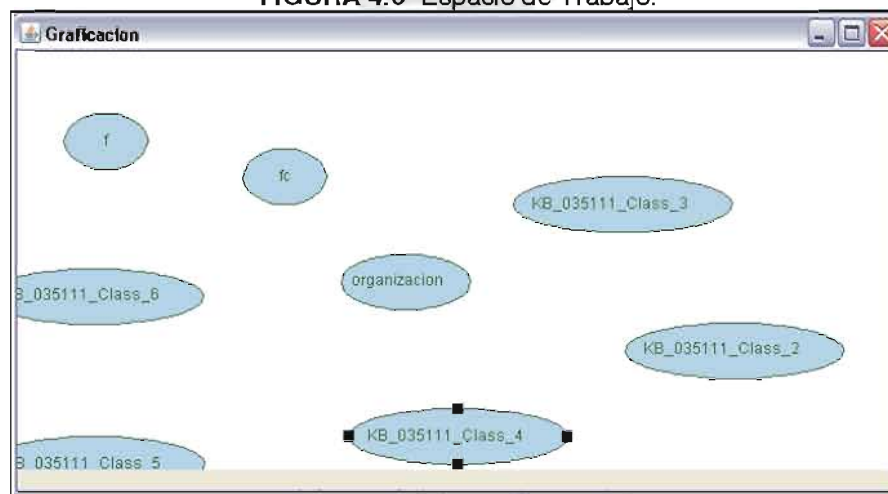
4.2.3.3 ESPACIO DE TRABAJO.

Requerimos un Panel Gráfico, espacio en el cual se van colocando los elementos de tipo gráficos, los cuales se van a visualizar, estos de hecho son los componentes para los cuales tenemos la estructura temporal explicada anteriormente y forman el esquema lógico de la ontología.

Para llenar este requerimiento, tomamos la herramienta de la *librería ZVTM*, con sus Paneles de Vista (representado por la clase *View*).

Básicamente, estos paneles de vista son espacios virtuales, los cuales tienen la característica de ser aparentemente infinitos. Es decir que podremos ampliar nuestro esquema hasta píxeles más grandes que la pantalla, claro esta, con otras herramientas que nos ayudaran a realizar la vista de este panel por partes.

FIGURA 4.6 Espacio de Trabajo.



[Fuente: Elaboración propia]

4.2.3.4 VISTA GLOBAL Y PARTICULAR DEL ESQUEMA DE LA ONTOLOGÍA.

Entramos a las características visuales, recomendadas por autores de técnicas de visualización estudiadas en el capítulo 3.

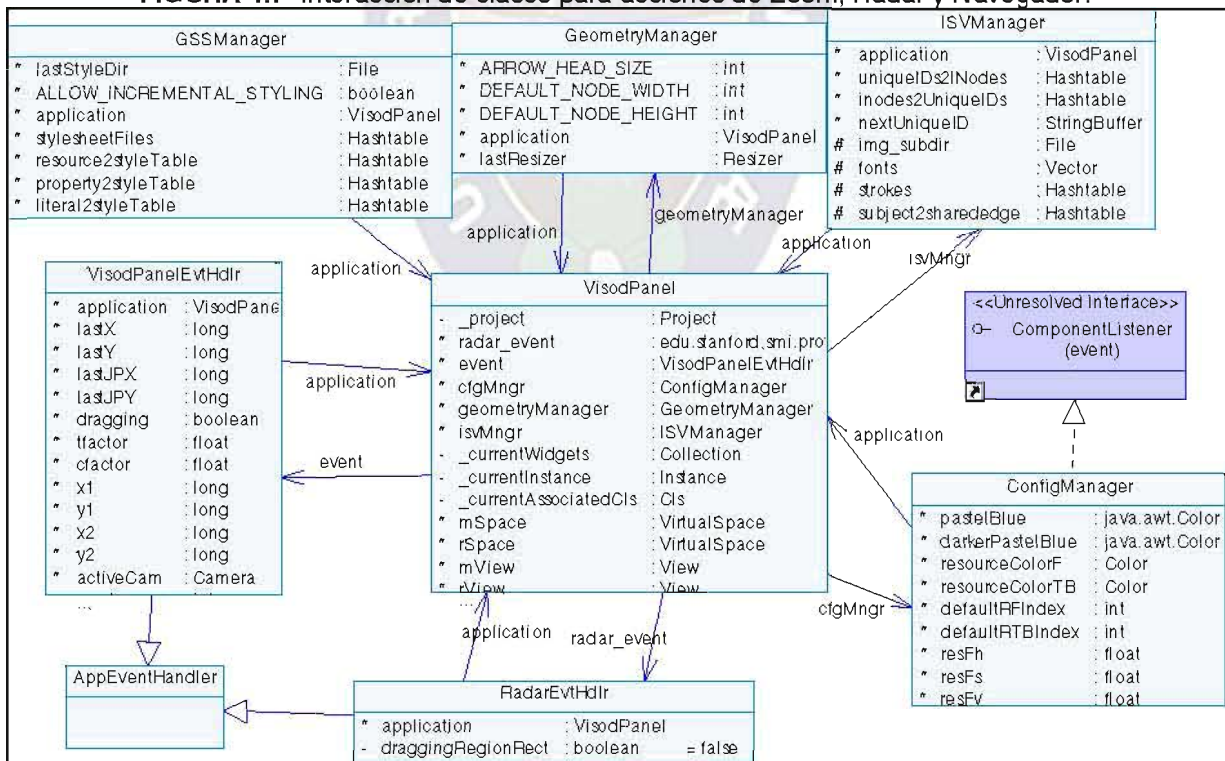
Básicamente, se presentan tres acciones que complementan la utilidad de la herramienta:

- Zoom.
- Radar.
- Navegador.

De estas tres tareas, se encarga la implementación de la técnica de cámaras que emplea la librería **ZVTM**, la cual nos servirá para manejar el espacio de trabajo desde varios puntos visuales (cámaras), es decir como si tuviéramos distintas cámaras que nos muestran diferentes vistas de un mismo espacio visual.

Las tres acciones, son manipuladas por clases del tipo control de eventos. Para las funciones de Zoom y navegador, la clase `VisodPanel`, se encarga de controlar dichos eventos. Para el caso del Radar, observamos que la clase `RadarEvtHdlr` como la que controla sus eventos. A continuación tenemos una parte de la interacción entre objetos.

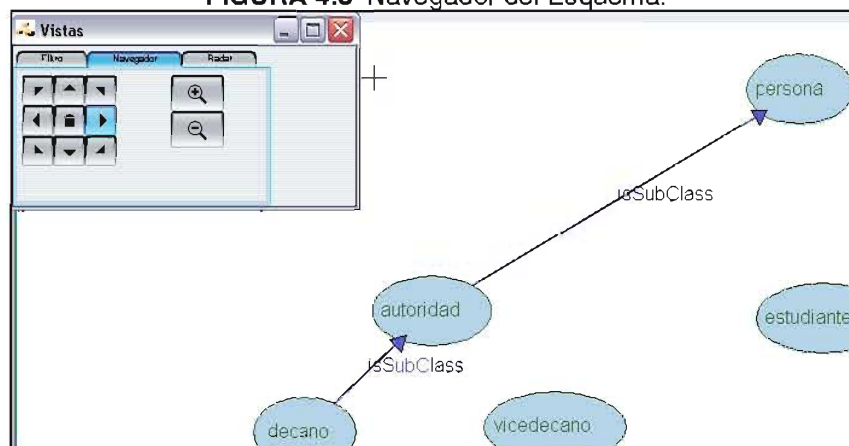
FIGURA 4.7 Interacción de clases para acciones de Zoom, Radar y Navegador.



[Fuente: Elaboración propia]

En cuanto a las funciones, El Zoom, simplemente es el acercamiento o alejamiento de la vista, ver Figura. En cuanto al navegador, simplemente es una palanca que recorre la vista principal arriba, abajo, derecha e izquierda, ver figura.

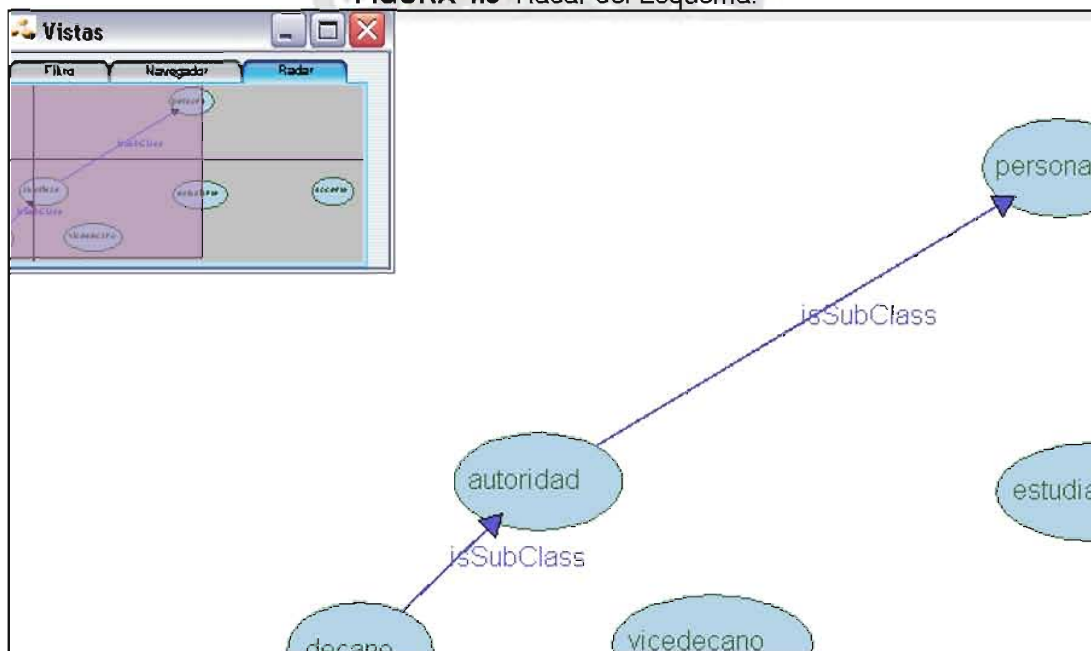
FIGURA 4.8 Navegador del Esquema.



[Fuente: Elaboración propia]

El Radar, es una inspección de toda el área utilizada desde una perspectiva tan alejada como para abarcar toda el área, y dando la posibilidad por medio de eventos del ZVTM, de ubicar la vista principal de acuerdo al eje de coordenadas móvil, ver figura.

FIGURA 4.9 Radar del Esquema.



[Fuente: Elaboración propia]

4.2.3.5 FILTRO DE DATOS.

Nuestro Filtro Visod es un modulo pequeño que hace la reconversión del atributo visible del Almacén de Datos de Carácter Lógico.

Para esto simplemente se deben seguir pasos fundamentales:

- No se trata de hacer menos comprensible el esquema.
- Debemos asignar visibilidad a algunos elementos, y a otros objetos debemos quitarles esa propiedad de visibilidad.

El algoritmo se muestra en la siguiente figura.

FIGURA 4.10 Algoritmo para filtro "visodFilter".

```

public static int CLASS_VIEW = 1;
public static int PROPERTIES_VIEW = 2,
/**
 * visodFilter Realiza el filtrado de objetos en el espacio
 * de trabajo
 * @type el tipo de objeto que deseamos ocultar
 */
void visodFilter (int type, Hashtable clases){
    Enumeration i, j, h;
    Vector v;
    // Vista de clases
    if(type == CLASS_VIEW){
        i = resourcesByURI.elements();
        while(i.hasMoreElements()){
            Object o = (IResource)i.nextElement();
            IResource _resource = (IResource)o;
            if((clases != null) && !clases.containsKey(_resource.getIdentity())){
                resourcesByURI.remove(_resource.getIdentity());
                _resource.setVisible(true);
                resourcesByURI.put(_resource.getIdentity(),_resource);
            }
        }
        h = propertiesByURI.elements();
        while(h.hasMoreElements()){
            Object o = h.nextElement();
            if(o instanceof IProperty){
                IProperty _property = (IProperty)o;
                if(!_property!=null){
                    propertiesByURI.remove(_property.getIdent());
                    _property.setVisible(false);
                    propertiesByURI.put(_property.getIdent(),_property);
                }
            }
        }
        //instancias
        j = instancesByURI.elements();
        while(j.hasMoreElements()){
            Object o = j.nextElement();
            if(o instanceof IProperty){
                IInstance _instance = (IInstance)o;
                instancesByURI.remove(_instance.getIdentity());
                _instance.setVisible(true);
                instancesByURI.put(_instance.getIdentity(),_instance);
            }
        }
    } else if (type == PROPERTIES_VIEW) {
        i = propertiesByURI.elements();
        while(i.hasMoreElements()){
            Object o = i.nextElement();
            if(o instanceof IProperty){
                IProperty _property = (IProperty)o,

```

```

        propertiesByURI remove(_property getIdent()),
        _property setVisible(true);
        propertiesByURI put(_property getIdent(),_property);
    }
}
i = resourcesByURI elements();
while(i.hasMoreElements()){
    Object o = i.nextElement();
    if(o instanceof IResource){
        IResource _resource = (IResource)o;
        if((clases !=null) && !clases.containsKey(_resource.getIdentity())){
            resourcesByURI remove(_resource.getIdentity()),
            _resource.setVisible(false),
            resourcesByURI put(_resource.getIdentity(),_resource);
        }
    }
}
//instancias
j = instancesByURI elements();
while(j.hasMoreElements()){
    Object o = j.nextElement();
    if(o instanceof IInstance){
        IInstance _instance = (IInstance)o;
        instancesByURI remove(_instance.getIdentity());
        _instance.setVisible(false);
        instancesByURI.put(_instance.getIdentity(),_instance);
    }
}
}
}
}

```

[Fuente: Elaboración propia]

Como explicación básica para el filtro, observamos que se realiza una alteración a la propiedad de visibilidad en los objetos almacenados temporalmente en las estructuras de objetos, representadas por arrays, los mismos que contienen agrupados por componentes las instancias de las clases estudiadas en la figura 4.7, que representan los objetos visuales en sí.

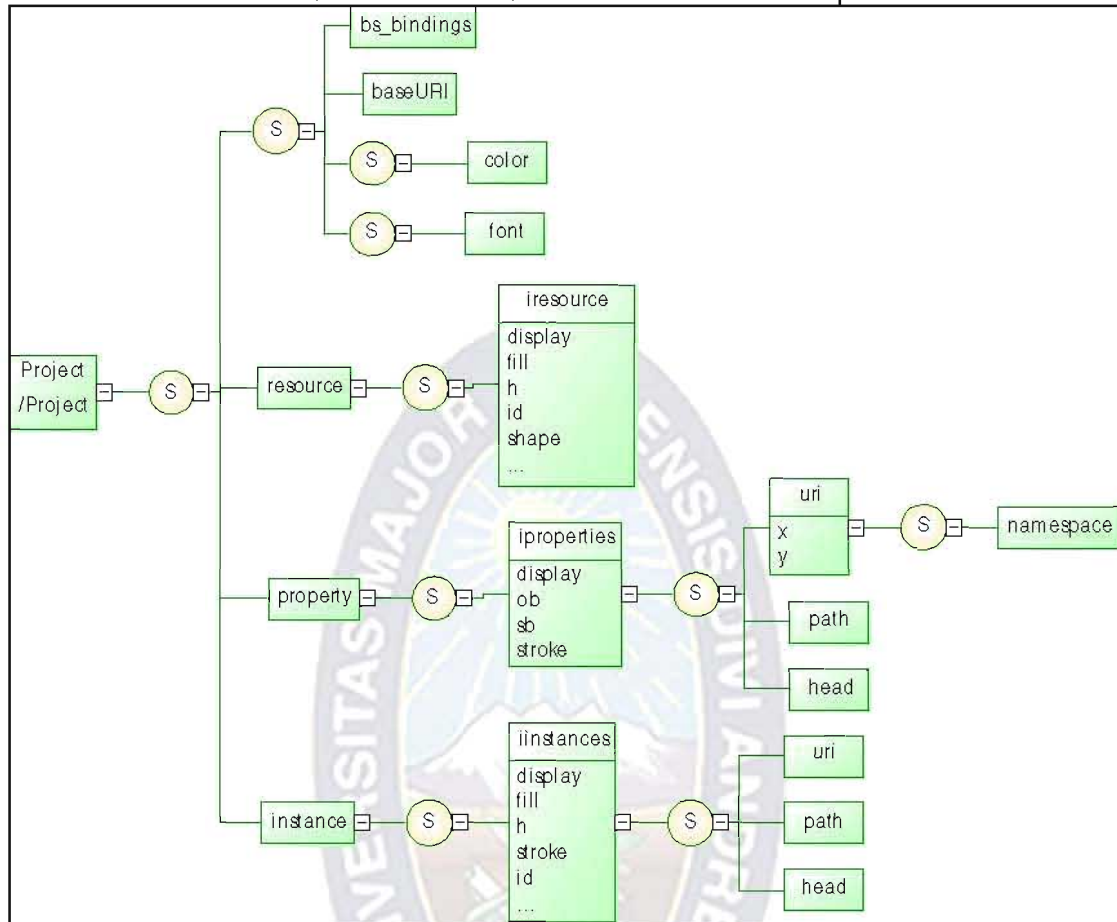
Estos arrays, observados en el algoritmo son tres: resourcesByURI, propertiesByURI e instancesByURI.

4.2.3.6 ALMACENAMIENTO PERMANENTE DE DATOS GRAFICOS.

Como habíamos explicado, los datos referentes a los gráficos se almacenan de forma temporal en una estructura de clases ya mencionada en la figura 4.5. Pero una vez concluida la edición por cualquier motivo, debemos guardar estos datos para proseguir con la edición en otro momento.

Para esta labor, se ha implementado el almacenamiento mediante archivos XML, el cual se ha planificado de tal modo que siga el siguiente árbol de representación:

FIGURA 4.11 Árbol estructura Almacén Permanente de Esquema Grafico.



[Fuente: Elaboración propia]

Notemos que tiene una estructura que hace una correspondencia con las clases observadas en la Figura 4.4. Esto es, porque se almacenan dichos datos en primera instancia almacenados de manera temporal.

Gracias a este archivo es que interactuamos dinámicamente con los atributos específicos para cada componente.

Para el manejo de dicho archivo XML, fue necesaria la interacción de clases, las cuales se describen en los anexos. Ahora es importante destacar que importamos las librerías de código abierto en Java:

```
import org.w3c.dom.Document;
import org.xml.sax.SAXException;
```

En la clase XMLManager, encontramos la funcionalidad de creación y recuperación de dichos archivos en formato XML, a través de interacción directa con ISVManager y las clases de la estructura de objetos ya estudiada: Iresource, Iproperty e IInstance.

4.2.3.7 ALMACENAMIENTO PERMANENTE DE GRAFICO.

Por otro lado, el esquema en formato de imagen, es una acción implementada mediante la conjunción de acciones de las siguientes tecnologías:

- Librería de Java “*imageio*” del paquete “*javax*”.
- Librería **ZVTM**, con su método de exportación de imagen a un Buffer de Imagen empleado en Java.

Esto nos da como resultado el almacenamiento del esquema gráfico en formato PNG.

4.2.4 ANALIZADOR SEMÁNTICO.

El analizador semántico es el razonador de ontologías que permitirá hacer el almacenamiento de los datos independientes y posteriormente convertirlos a formato del lenguaje OWL.

Nuestros objetos visuales en realidad son un puente de comunicación con este razonador. También, observamos que este componente es el encargado de la devolución de un producto, que en nuestro caso es un archivo con formato OWL.

Este razonador también incorpora un validados OWL, que se comunica con la interfaz de gestión de proyectos para la interacción con el usuario.

Todo este trabajo es llevado gracias a las librerías de **JENA** y la estructura de almacenamiento temporal propuesta por **Protege-OWL 3.2.1**.

Esta labor de transposición de datos esta incluida en las librerías de **Protege3.3**. **Jena**, también contiene un soporte de inferencias útil para otro tipo de aplicaciones.

4.3 PRESENTACIÓN DE LA SIMULACIÓN.

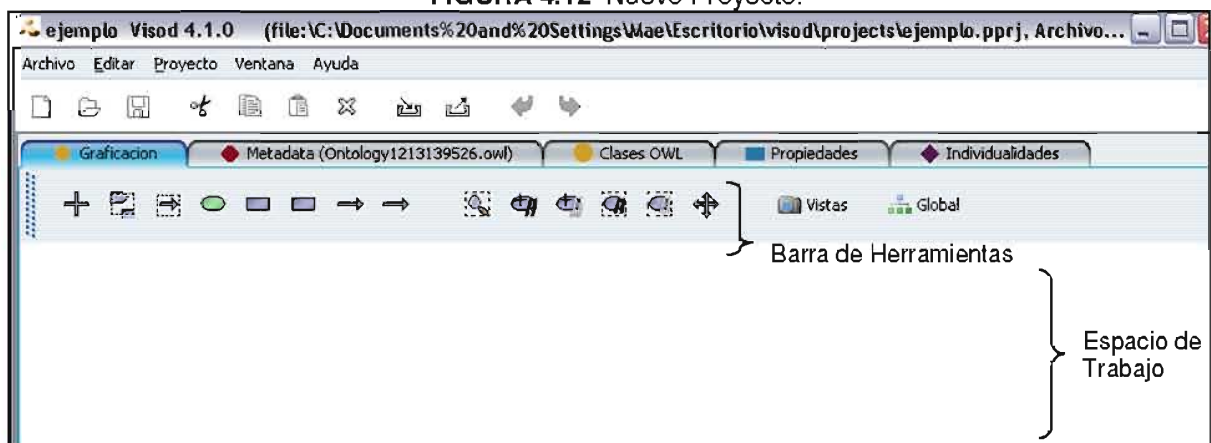
4.3.1 PRIMERA SIMULACIÓN.

Siguiendo el ejemplo utilizado para explicar los pasos de diseño de una ontología observado en el punto 4.1, ahora observaremos los pormenores del diseño con la herramienta.

PROYECTO:

Primero creamos un nuevo proyecto, lo guardamos. Continuamos observando el espacio de trabajo y colocando las clases con la herramienta “Crear Recurso”. Ver siguiente figura:

FIGURA 4.12 Nuevo Proyecto.



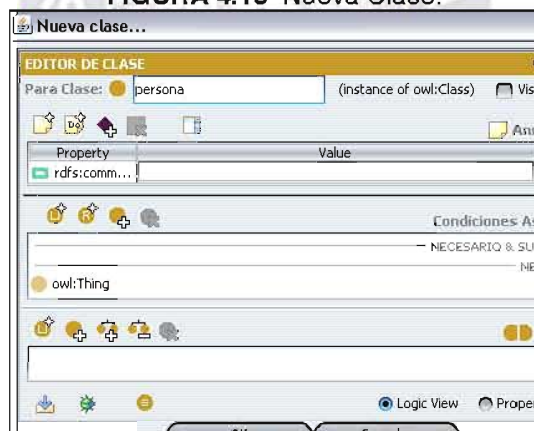
[Fuente: Elaboración Propia]

CLASES:

Nuestras clases se colocaran por partes, la referida a la clase padre Persona y la clase Organización.

Al clickear en el espacio de trabajo, aparece el cuadro de dialogo:

FIGURA 4.13 Nueva Clase.



[Fuente: Elaboración Propia]

Verifiquemos que si repetimos un nombre ya asignado, o con caracteres inválidos, el indicador de error marcara el nombre con color rojo. Ejemplo, ver siguiente figura:

FIGURA 4.14 Validación en la Clase.



[Fuente: Elaboración Propia]

PROPIEDADES:

Los Objetos Propiedades, también pueden crearse gráficamente, y a través del panel de edición.

La relación entre clases, o predicado, se dibuja con la herramienta Crear Propiedad:

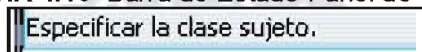
FIGURA 4.15 Crear Propiedad



[Fuente: Elaboración Propia]

Para cualquier herramienta escogida también observamos indicadores en la barra de estado del espacio de trabajo. A continuación el indicador para el proceso de relacionamiento.

FIGURA 4.16 Barra de Estado Panel de Trabajo



[Fuente: Elaboración Propia]

Al escoger la clase sujeto de nuestro predicado, aparecerá el siguiente cuadro de dialogo:

FIGURA 4.17 Nuevo Predicado



[Fuente: Elaboración Propia]

Básicamente son pequeñas herramientas para crear el predicado. El indicador verde nos da la señal de un predicado correcto.

FIGURA 4.18 Insertar Clase



[Fuente: Elaboración Propia]

También podemos escoger el objeto del predicado mediante la opción "Insertar Clase" y clickeando la clase en el dibujo. Ver siguiente figura.

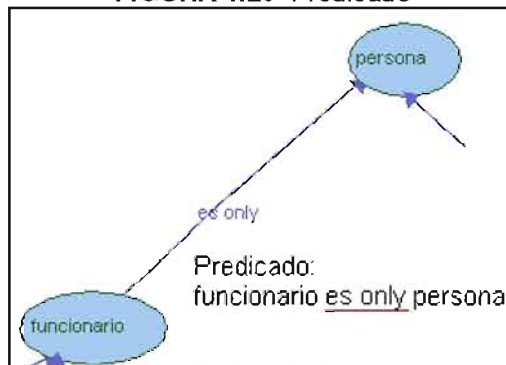
FIGURA 4.19 Barra de Herramientas



[Fuente: Elaboración Propia]

Presionando Ok, marcado con verde, ya tenemos nuestro predicado.

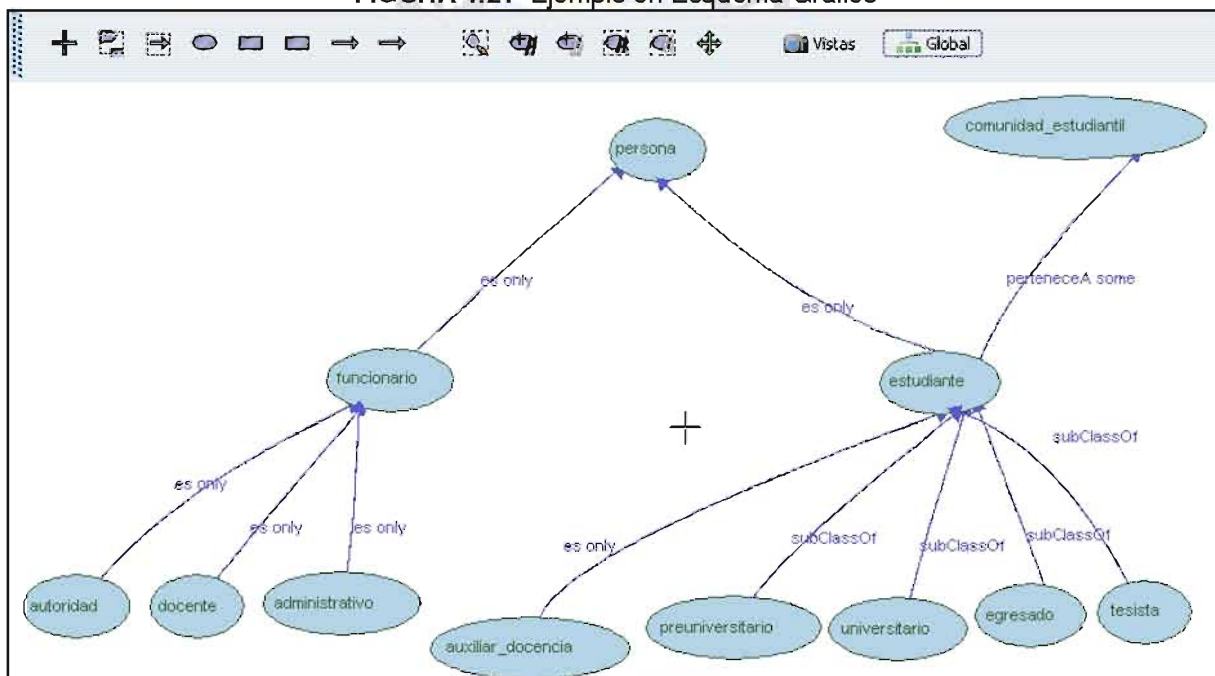
FIGURA 4.20 Predicado



[Fuente: Elaboración Propia]

Es así que llegamos al siguiente esquema:

FIGURA 4.21 Ejemplo en Esquema Grafico



[Fuente: Elaboración Propia]

Una vez guardado, observamos el archivo ejemplo2.owl, con el siguiente resultado:

FIGURA 4.22 Ejemplo en formato OWL

```
<?xml version="1.0"?>
<rdf RDF
  xmlns rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns owl="http://www.w3.org/2002/07/owl#"
  xmlns xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns="http://www.owl-ontologies.com/Ontology1213141787.owl#"
  xmlns rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.owl-ontologies.com/Ontology1213141787.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="persona"/>
  <owl:Class rdf:ID="funcionario">
    <rdfs:subClassOf rdf:resource="#persona"/>
    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  </owl:Class>
```

```

<owl Class rdf ID="auxiliar_docencia">
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty>
        <owl ObjectProperty rdf ID="es"/>
      </owl onProperty>
      <owl allValuesFrom>
        <owl Class rdf ID="estudiante"/>
      </owl allValuesFrom>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl Class rdf ID="autoridad">
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty rdf resource="#es"/>
      <owl allValuesFrom rdf resource="#funcionario"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl Class rdf ID="egresado">
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty rdf resource="#es"/>
      <owl allValuesFrom>
        <owl Class rdf about="#estudiante"/>
      </owl allValuesFrom>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl Class rdf ID="testista">
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom>
        <owl Class rdf about="#estudiante"/>
      </owl allValuesFrom>
      <owl onProperty rdf resource="#es"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl Class rdf ID="universitario">
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty rdf resource="#es"/>
      <owl allValuesFrom>
        <owl Class rdf about="#estudiante"/>
      </owl allValuesFrom>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl Class rdf ID="preuniversitario">
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom>
        <owl Class rdf about="#estudiante"/>
      </owl allValuesFrom>
      <owl onProperty rdf resource="#es"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl Class rdf about="#estudiante">
  <rdfs subClassOf>
    <owl Restriction>
      <owl someValuesFrom>

```

```

    <owl Class rdf ID="comunidad_estudiantil"/>
  </owl someValuesFrom>
  <owl onProperty>
    <owl ObjectProperty rdf ID="perteneceA"/>
  </owl onProperty>
  </owl Restriction>
</rdfs subClassOf>
<rdfs subClassOf>
  <owl Restriction>
    <owl allValuesFrom rdf resource="#persona"/>
    <owl onProperty rdf resource="#es"/>
  </owl Restriction>
</rdfs subClassOf>
<rdfs subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl Class rdf ID="administrativo">
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom rdf resource="#funcionario"/>
      <owl onProperty rdf resource="#es"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl Class rdf ID="docente">
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom rdf resource="#funcionario"/>
      <owl onProperty rdf resource="#es"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl ObjectProperty rdf ID="subClaseDe"/>
<owl ObjectProperty rdf ID="subordinadoDe"/>
<owl ObjectProperty rdf ID="subOrganizacionDe"/>
<owl ObjectProperty rdf ID="esAlumnoDe"/>
<owl ObjectProperty rdf ID="depedienteDe"/>
<owl ObjectProperty rdf ID="parteDe"/>
</rdf RDF>
<!-- Created -->

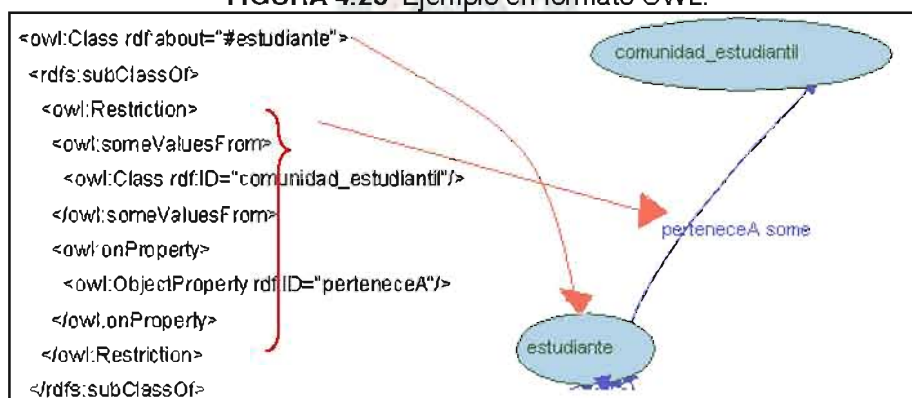
```

[Fuente: Elaboración Propia]

La versión de la ontología en formato OWL completa la observamos en Anexos.

Observando detenidamente tenemos la correlación con los componentes gráficos. Por ejemplo, tenemos:

FIGURA 4.23 Ejemplo en formato OWL.



[Fuente: Elaboración Propia]

Verificamos que efectivamente se trata de un archivo con el formato OWL.

4.3.2 SEGUNDA SIMULACIÓN.

En el siguiente cuadro se muestra como se ingresa los datos de los objetos que conforman la ontología bastante simple, se trata de una clase escritor, con su única subclase Galeano.

FIGURA 4.24 Barra de Herramientas.



[Fuente: Elaboración Propia]

En la primera vista, se observa el tab de graficación, llamado graficación, y la barra de herramientas, donde exceptuando los tres botones finales, los demás son para manejar los objetos dentro del espacio de trabajo, ya sea para crearlos o moverlos.

Los pasos relevantes para este ejemplo se denotan con la introducción de instancias.

Ahora mostremos como creamos cada instancia en nuestro esquema. Empezamos igual que en una anteriores etapas, escogiendo la herramienta crear Instancias de la Barra de Herramientas.

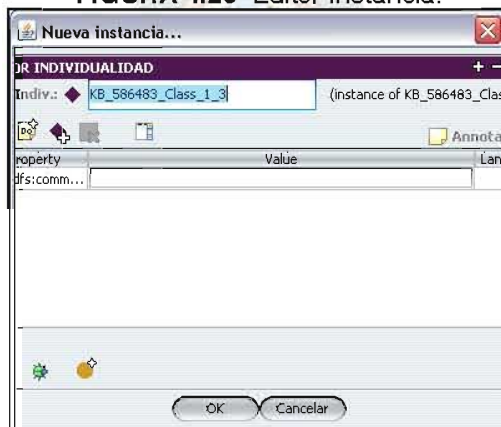
FIGURA 4.25 Crear Instancia.



[Fuente: Elaboración Propia]

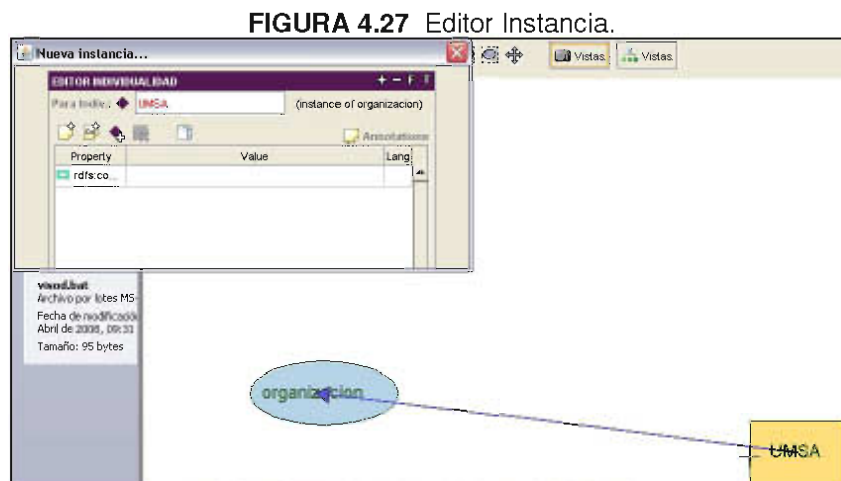
Si tenemos seleccionado una clase, solo clickeamos en el espacio de trabajo obtenemos la siguiente pantalla, si no se nos pedirá que se lo haga.

FIGURA 4.26 Editor Instancia.



[Fuente: Elaboración Propia]

Como se observa en la siguiente figura nos indica si es que ya existe una instancia con el mismo nombre.



[Fuente: Elaboración Propia]

Cuando ya tenemos una parte del esquema, se realiza el procedimiento de guardar, con un cuadro de diálogo, este proceso puede tardar dependiendo el tamaño del esquema. Revisando nuestro ejemplo observamos que al guardar obtenemos cuatro archivos los cuales están correctamente guardados.

FIGURA 4.28 Archivos resultantes.

```
C:\Documents and Settings\Mae\Escritorio\visod\projects>dir
El volumen de la unidad C no tiene etiqueta.
El número de serie del volumen es: 3490-5314

Directorio de C:\Documents and Settings\Mae\Escritorio\visod\projects
11/06/2008  04:14 p.m.    <DIR>          .
11/06/2008  04:14 p.m.    <DIR>          ..
10/06/2008  08:08 p.m.             4 511 ejemplo.owl
10/06/2008  08:08 p.m.          186 328 ejemplo.pprj
10/06/2008  08:08 p.m.             8 533 ejemplo.pvod
10/06/2008  08:08 p.m.             26 ejemplo.repository
                4 archivos          199 398 bytes
                2 dirs   12 405 354 496 bytes libres

C:\Documents and Settings\Mae\Escritorio\visod\projects>
```

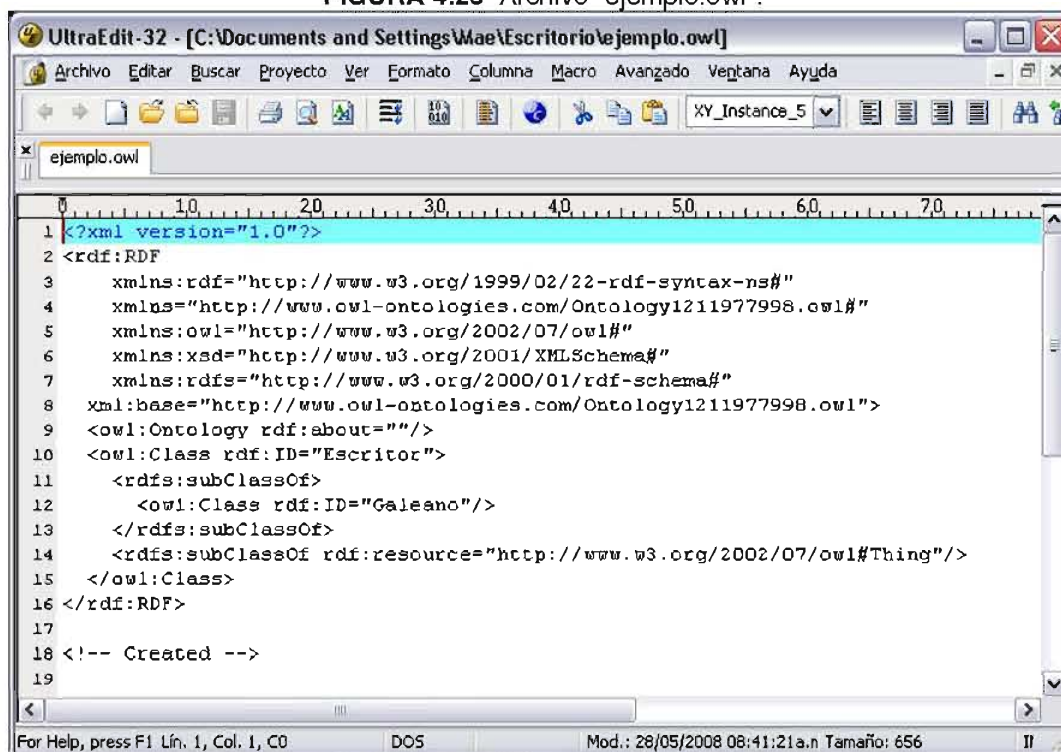
[Fuente: Elaboración Propia]

El archivo ejemplo.pvod, es un XML que corresponde al árbol observado en la figura 4.11, ya explicado. Los archivos ejemplo.pprj y ejemplo.repository corresponden al proyecto.

El resultado de toda la graficación es el archivo ejemplo.owl. Ahora, con cualquier editor de texto, observamos el contenido de dicho archivo en la figura 4.29.

Con detenimiento vemos cada parte de los tags y podemos realizar una correspondencia con el esquema, con exactitud.

FIGURA 4.29 Archivo "ejemplo.owl".



```

1 <?xml version="1.0"?>
2 <rdf:RDF
3   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4   xmlns="http://www.owl-ontologies.com/Ontology1211977998.owl#"
5   xmlns:owl="http://www.w3.org/2002/07/owl#"
6   xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
7   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
8   xml:base="http://www.owl-ontologies.com/Ontology1211977998.owl">
9   <owl:Ontology rdf:about="" />
10  <owl:Class rdf:ID="Escritor">
11    <rdfs:subClassOf>
12      <owl:Class rdf:ID="Galeano"/>
13    </rdfs:subClassOf>
14    <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
15  </owl:Class>
16 </rdf:RDF>
17
18 <!-- Created -->
19

```

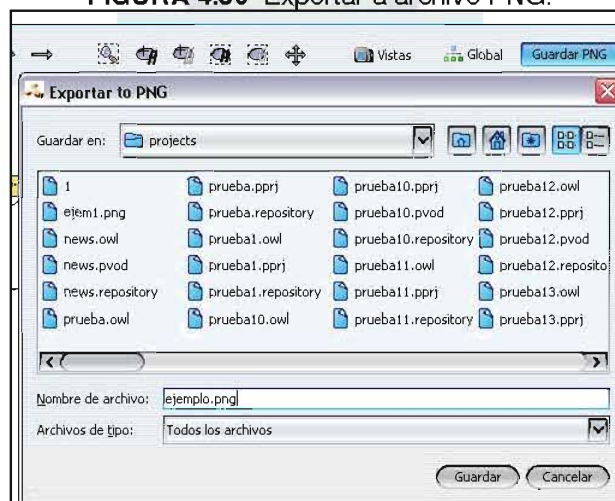
For Help, press F1 Lín. 1, Col. 1, CO DOS Mod.: 28/05/2008 08:41:21a.n Tamaño: 656

[Fuente: Elaboración Propia]

Por último, recurrimos a la funcionalidad de la exportación del esquema gráfico en un archivo gráfico PNG, imagen bitmap. Esta acción se realiza a través del botón "Guardar PNG" de la barra de herramientas, ver Figura 4.30.

El resultado simplemente es un archivo .png, el cual refleja la misma gráfica del esquema elaborado.

FIGURA 4.30 Exportar a archivo PNG.



[Fuente: Elaboración Propia]

5 DISCUSION Y CONCLUSIONES

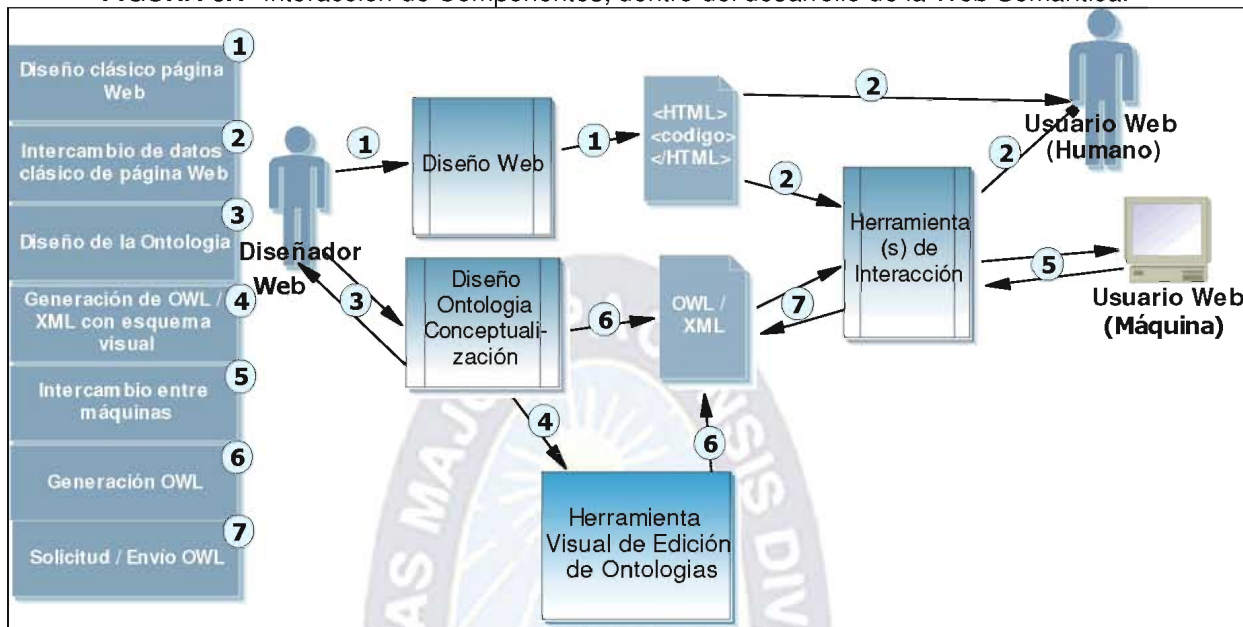
La observación de los resultados en las pruebas de la Herramienta Visual de Edición de Ontologías, llamado Visod, da por entendido los siguientes puntos:

5.1 HERRAMIENTA DE EDICION DE ONTOLOGIAS.

Es un software que sirve para conseguir la edición de una ontología, de forma gráfica, de tal modo que su resultado es claramente una equivalencia entre el esquema diseñado con las funcionalidades dispuestas a un archivo de tipo OWL, el cual se encuentra listo para ser implementado, adjuntado a un proyecto de Web Semántica. Este proceso realizado a través del uso de técnicas de graficación simples, las cuales ayudan al usuario en varias tareas como la dispersión de los datos, el zoom, la desconcentración de los datos.

La herramienta tiene un papel importante dentro de un ciclo de una página web semántica, como se muestra en la figura 5.1. Observemos cual es su posición, interactuando directamente con el diseño, el cual tiene características de enorme responsabilidad dadas las atribuciones que conlleva una web semántica, y que incide directamente en la puesta en producción de dicha página web, ya que los resultados que se observarán más detenidamente más adelante, demuestran archivos netamente OWL los cuales pueden ser llevados directamente al servidor si el caso lo sugiere.

FIGURA 5.1 Interacción de Componentes, dentro del desarrollo de la Web Semántica.



[Fuente: Elaboración propia]

5.2 CONCLUSIONES.

Al incursionar en la tendencia de sitios Web Semánticos, observamos que la utilidad que se extrae de ellos es de gran importancia, puesto que la interoperabilidad a nivel semántico, posibilita la incursión de nuevas tecnologías como un ejemplo están tecnologías como los agentes, buscadores semánticos, etc. Hecho que eleva sustancialmente la labor de encontrar información en un nivel eficiente, incentivando la mayor explotación de la información.

En consecuencia la definición de variables a partir de la identificación del problema llevó a encontrar las variables independiente y dependiente de la herramienta.

La variable independiente VI(X): Herramienta Visual de Edición de Ontologías, permite la esquematización visual de una ontología.

La variable dependiente VD(Y): Generación de ontologías en lenguaje OWL para una Web Semántica, genera la ontología esquematizada en un lenguaje OWL.

Es así que, la implementación de sitios Web Semánticos, se facilita e incrementa con la Herramienta Visual de Ontologías, de tal modo que el diseñador, puede realizar una diagramación del esquema y este sea llevado posteriormente a su equivalente en formato OWL.

Para comprobar la hipótesis planteada, se realizó la prueba de la esquematización visual efectiva de una Página Web perteneciente a la Facultad de Ciencias Puras y Naturales en su sección Institucional, la cual se trasladó al lenguaje OWL, verificándose la equivalencia entre el esquema y el código en el lenguaje OWL.



BIBLIOGRAFIA

Libros:

- "A Framework for the Visualization of Multidimensional and Multivariate Data", Selan Rodrigues dos Santos, The University of Leeds School of Computing.
- "Visualization and Protégé", Margaret-Anne Storey, Robert Lintern, Neil Ernst, David Perrin.
- "OntoSphere: more than a 3D ontology visualization tool", Alessio Bosca, Dario Bonino, Paolo Pellegrino.
- "Visualizing Data", Ben Fry, Ed. O'Reilly.

Artículos en Internet:

- W3C (World Wide Web Consortium) Documentos y recomendaciones del sitio [<http://www.w3c.org/>]
- W3C (World Wide Web Consortium) Características OWL del sitio [<http://www.w3.org/TR/owl-features/>]
- Artículo: "OWL", Castells Pablo y Saiz Francisco, del sitio [<http://www.ii.uam.es/%7Ecastells/docencia/semanticweb/apuntes/6-owl.pdf>]
- Artículo: "Tecnologías XML y Web Semántica", Departamento de Informática-Universidad de Oviedo
- Artículo: "El futuro de la Web", Miguel Ángel Abián
- Artículo: "La Web Semántica", Luís Codina y Cristòfol Rovira.
- Artículo: "La Web Semántica", Pablo Castells
- Proyectos de lenguajes para ontologías en el sitio [<http://www.ontology.org/>].
- Apuntes proyecto Dublín Core del sitio [<http://www.dcmi.org/>].
- Protege-2000, Stanford Medical Informatics, [<http://protege.stanford.edu>]
- Polanco, X., & Zartl, A. (2002). Information visualization. EICSTES Project. Deliverable 1.4. State of the art part c: WP9. Retrieved 12 April, 2005 de [http://eicstes.inist.fr/public/D1.4_Visualization_WP9.pdf]
- Recuperado el Domingo, 13 de Mayo de 2007, de [<http://eprints.rclis.org/archive/00006616/02/paper258.pdf>]

- Dürsteler, J.C. (2002). Information visualisation, what is it all about?. Inf@Vis! La revista digital de InfoVis.net, (100). Recuperado el 22 de Noviembre del 2005 de [<http://www.infovis.net/printMag.php?num=100&lang=2>] recuperado el Domingo, 13 de Mayo de 2007, de [<http://eprints.rclis.org/archive/00006616/02/paper258.pdf>]
- Metodologías para el desarrollo de interfaces visuales de recuperación de información: análisis y comparación Víctor Herrero-Solana y Yusef Hassan Grupo SCImago, Universidad de Granada Recuperado el Domingo, 13 de Mayo de 2007, de [<http://eprints.rclis.org/archive/00006616/02/paper258.pdf>]
- "Visualizing on-line activity" Eick, S.G. (2001). Communication of the ACM, 44(8), 45-52.
- "Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología", Natalya F. Noy y Deborah L. McGuinness, Stanford University, Stanford, CA, 94305, Traducido del inglés por: Erick Antezana. 19 de septiembre 2005.

Tecnologías

- Protege y Protege-OWL 3.2.1, sitio de descarga [<http://protege.stanford.edu/download/download.html>].
- Jena 2, sitio Oficial [<http://jena.sourceforge.net/>].
- ZVTM, descarga de los sitios [<http://zvtm.sourceforge.net/index.html>] y [<http://zvtm.sourceforge.net/doc.html>]

ANEXOS

ANEXO 1: División de las VIRI (Interfaces para la recuperación de la Información).



[Fuente: Elaboración Propia]

ANEXO 2: Listado de técnicas de visualización y sus respectivas referencias en orden alfabético.

Este documento primero describe las características en Lite OWL, seguidamente por una descripción de las características que son adicionadas en OWL DL y OWL Full (OWL DL y OWL Full contienen las mismas características, pero OWL Full es más liberal acerca de cómo estas características pueden ser combinadas).

TABLA Anexo2: Listado de técnicas de visualización y sus respectivas referencias en orden alfabético

WebBook	3D scatterplots
3D trees	Aggregate Manipulator
Andrews curves	Animation
Bifocal lens	Brushing and linking
Cam Trees	Chernoff faces
Circle segments	Cluster analysis
Graph visualizations	Color icons
Cone trees	Data visualization sliders
Details-on-demand	Dimension ordering
Dimension stacking	Document lens
Dynamic Query (DQ)	FilmFinder
Fisheye views	GIS
Grand tour	Heat maps
Hierarchical axis	Hierarchical parallel coordinates
HomeFinder	Human project
Hyperbolic trees	Hyperbox
Hyperslice	Influence Explorer
Infocrystal	Information mural
Interactive mapping	Interpolation
Landscapes	LifeLines
Macromedia Director	Magic lens filter
M and N plot	Mapping categorical data to numerical representation
Movable Filters	Multidimensional scaling methods(MDS)
Multi-line graphs	Natural textures
Netmaps	Networks
Neural network algorithms	Parallel coordinates
Perspective wall	Pixel-oriented methods

Polar charts	Principal component analysis (PCA)
Projection pursuit	Project management tools
Prosecons views	Quad-tree mapping (QTM)
RadViz	Rapid serial visual presentation- RSVP
Recursive pattern	Rotation in the n -dimensional data space
Sampling	Scalar visualization animation model
Scatterplot matrix	SeeSoft
Selective Dynamic Manipulator	Shape coding (autoglyph)
SOM maps	Spatial display of document collections
Spiral	Spotfire
Spreadsheet-like interface for visualization exploration	Star coordinates
Starfield visualization	Star glyph
Stars	Statistical procedures normalization, mean, standard deviation
Stick figure (Exvis) 's	Survey plots
Table lens	TreeBrowser
Treemap	Trees and castles
Value bars	VisDB
Visual Hierarchical Dimension Reduction (VHDR)	Visualization for multidimensional function by projections (VMFP)
Worlds within worlds	Zooming & panning

[Fuente: "A Framework for the Visualization of Multidimensional and Multivariate Data", Selan Rodrigues dos Santos, The University of Leeds School of Computing, Pag. 68-69]

ANEXO 3: Ejemplo de Ontologías en lenguaje OWL del ejemplo completo de la Página Web de la Facultad de Ciencias Puras y Naturales enlace Institucionalidad.

TABLA Anexo3: Ejemplo OWL Página Web F.C.P.N. enlace Institucionalidad.

<pre> <?xml version="1.0"?> <rdf RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns="http://www.ow ontologies.com/Ontology1213690448.owl#" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:xsd="http://www.w3.org/2001/XMLSchema#" xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xml:base="http://www.ow ontologies.com/Ontology1213690448.owl"> <owl:Ontology rdf:about=""/> <owl:Class rdf:ID="decanatura"> <rdfs:subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/> <rdfs:subClassOf> <owl:Restriction> <owl:onProperty> <owl:ObjectProperty rdf:ID="es"/> </owl:onProperty> <owl:allValuesFrom> <owl:Class rdf:ID="departamento"/> </owl:allValuesFrom> </owl:Restriction> </rdfs:subClassOf> </owl:Class> <owl:Class rdf:ID="carrera_quimica"/> <owl:Class rdf:ID="egresado"> <rdfs:subClassOf> <owl:Class rdf:ID="estudiante"/> </rdfs:subClassOf> <rdfs:subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/> </owl:Class> <owl:Class rdf:ID="persona"/> <owl:Class rdf:ID="carrera_informatica"/> <owl:Class rdf:about="#departamento"> <rdfs:subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/> </pre>	<pre> <rdfs:subClassOf> <owl:Restriction> <owl:someValuesFrom> <owl:Class rdf:ID="organización"/> </owl:someValuesFrom> <owl:onProperty> <owl:ObjectProperty rdf:ID="esParteDe"/> </owl:onProperty> </owl:Restriction> </rdfs:subClassOf> </owl:Class> <owl:Class rdf:ID="preuniversitario"> <rdfs:subClassOf> <owl:Class rdf:about="#estudiante"/> </rdfs:subClassOf> <rdfs:subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/> </owl:Class> <owl:Class rdf:ID="funcionario"> <rdfs:subClassOf> <owl:Restriction> <owl:allValuesFrom rdf:resource="#persona"/> <owl:onProperty rdf:resource="#es"/> </owl:Restriction> </rdfs:subClassOf> <rdfs:subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/> </owl:Class> <owl:Class rdf:ID="docente"> <rdfs:subClassOf> <owl:Restriction> <owl:allValuesFrom rdf:resource="#funcionario"/> <owl:onProperty rdf:resource="#es"/> </owl:Restriction> </rdfs:subClassOf> <rdfs:subClassOf rdf resource="http://www.w3.org/2002/07/owl#Thing"/> </owl:Class> </pre>
--	--

```

<owl Class rdf ID="grupo_investigacion">
  <rdfs subClassOf
rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom rdf resource="#organización"/>
      <owl onProperty rdf resource="#es"/>
    </owl Restriction>
  </rdfs subClassOf>
</owl Class>
<owl Class rdf ID="tesisista">
  <rdfs subClassOf>
    <owl Class rdf about="#estudiante"/>
  </rdfs subClassOf>
  <rdfs subClassOf
rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl Class rdf ID="programa">
  <rdfs subClassOf
rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty rdf resource="#es"/>
      <owl allValuesFrom rdf resource="#organización"/>
    </owl Restriction>
  </rdfs subClassOf>
</owl Class>
<owl Class rdf ID="comunidad_estudiantil"/>
<owl Class rdf ID="kardex_academico">
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom>
        <owl Class rdf ID="vicedecanatura"/>
      </owl allValuesFrom>
      <owl onProperty>
        <owl ObjectProperty rdf ID="dependeDe"/>
      </owl onProperty>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf
rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty rdf resource="#es"/>
      <owl allValuesFrom rdf resource="#departamento"/>
    </owl Restriction>
  </rdfs subClassOf>
</owl Class>
<owl Class rdf ID="carrera_fisica"/>
<owl Class rdf ID="administrativo">
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom rdf resource="#funcionario"/>
      <owl onProperty rdf resource="#es"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf
rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl Class rdf ID="carrera_biologia"/>
<owl Class rdf ID="jefatura_carrera">
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom rdf resource="#carrera_fisica"/>
      <owl onProperty>
        <owl ObjectProperty rdf ID="aCargoDe"/>
      </owl onProperty>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf>

```

```

<owl Restriction>
  <owl allValuesFrom rdf resource="#carrera_quimica"/>
  <owl onProperty>
    <owl ObjectProperty rdf about="#aCargoDe"/>
  </owl onProperty>
</owl Restriction>
</rdfs subClassOf>
  <rdfs subClassOf
rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom
rdf resource="#carrera_informatica"/>
      <owl onProperty>
        <owl ObjectProperty rdf about="#aCargoDe"/>
      </owl onProperty>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty>
        <owl ObjectProperty rdf about="#dependeDe"/>
      </owl onProperty>
      <owl allValuesFrom>
        <owl Class rdf about="#vicedecanatura"/>
      </owl allValuesFrom>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom>
        <owl Class rdf ID="carrera_estadistica"/>
      </owl allValuesFrom>
      <owl onProperty>
        <owl ObjectProperty rdf about="#aCargoDe"/>
      </owl onProperty>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty rdf resource="#es"/>
      <owl allValuesFrom rdf resource="#departamento"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty>
        <owl ObjectProperty rdf about="#aCargoDe"/>
      </owl onProperty>
      <owl allValuesFrom rdf resource="#carrera_biologia"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty>
        <owl ObjectProperty rdf about="#aCargoDe"/>
      </owl onProperty>
      <owl allValuesFrom>
        <owl Class rdf ID="carrera_matematicas"/>
      </owl allValuesFrom>
    </owl Restriction>
  </rdfs subClassOf>
</owl Class>
<owl Class rdf ID="centro_publicaciones">
  <rdfs subClassOf
rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom rdf resource="#departamento"/>
      <owl onProperty rdf resource="#es"/>
    </owl Restriction>
  </rdfs subClassOf>

```



```

</owl Restriction>
</rdfs subClassOf>
<rdfs subClassOf>
  <owl Restriction>
    <owl onProperty>
      <owl ObjectProperty rdf about="#dependeDe"/>
    </owl onProperty>
    <owl allValuesFrom>
      <owl Class rdf about="#vicedecanatura"/>
    </owl allValuesFrom>
  </owl Restriction>
</rdfs subClassOf>
</owl Class>
<owl Class rdf ID="Instituto">
  <rdfs subClassOf
rdf resource="http //www w3 org/2002/07/owl# Thing"/>
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom rdf resource="#organización"/>
      <owl onProperty rdf resource="#es"/>
    </owl Restriction>
  </rdfs subClassOf>
</owl Class>
<owl Class rdf ID="universidad">
  <rdfs subClassOf
rdf resource="http //www w3 org/2002/07/owl# Thing"/>
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty rdf resource="#es"/>
      <owl allValuesFrom rdf resource="#organización"/>
    </owl Restriction>
  </rdfs subClassOf>
</owl Class>
<owl Class rdf ID="autoridad">
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom rdf resource="#funcionario"/>
      <owl onProperty rdf resource="#es"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf
rdf resource="http //www w3 org/2002/07/owl# Thing"/>
</owl Class>
<owl Class rdf ID="jefe_carrera">
  <rdfs subClassOf>
    <owl Restriction>
      <owl allValuesFrom rdf resource="#autoridad"/>
      <owl onProperty rdf resource="#es"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf
rdf resource="http //www w3 org/2002/07/owl# Thing"/>
</owl Class>
<owl Class rdf ID="decano">
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty rdf resource="#es"/>
      <owl allValuesFrom rdf resource="#autoridad"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf
rdf resource="http //www w3 org/2002/07/owl# Thing"/>
</owl Class>
<owl Class rdf about="#estudiante">
  <rdfs subClassOf>
    <owl Restriction>
      <owl someValuesFrom
rdf resource="#comunidad_estudiantil"/>
      <owl onProperty>
        <owl ObjectProperty rdf ID="perteneceA"/>

```

```

</owl onProperty>
</owl Restriction>
</rdfs subClassOf>
<rdfs subClassOf>
  <owl Restriction>
    <owl allValuesFrom rdf resource="#persona"/>
    <owl onProperty rdf resource="#es"/>
  </owl Restriction>
</rdfs subClassOf>
<rdfs subClassOf
rdf resource="http //www w3 org/2002/07/owl# Thing"/>
<rdfs subClassOf>
  <owl Restriction>
    <owl onProperty>
      <owl ObjectProperty rdf ID="estudiaEn"/>
    </owl onProperty>
    <owl someValuesFrom rdf resource="#universidad"/>
  </owl Restriction>
</rdfs subClassOf>
</owl Class>
<owl Class rdf ID="colegio">
  <rdfs subClassOf
rdf resource="http //www w3 org/2002/07/owl# Thing"/>
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty rdf resource="#es"/>
      <owl allValuesFrom rdf resource="#organización"/>
    </owl Restriction>
  </rdfs subClassOf>
</owl Class>
<owl Class rdf about="#vicedecanatura">
  <rdfs subClassOf
rdf resource="http //www w3 org/2002/07/owl# Thing"/>
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty rdf resource="#es"/>
      <owl allValuesFrom rdf resource="#departamento"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty>
        <owl ObjectProperty rdf about="#dependeDe"/>
      </owl onProperty>
      <owl allValuesFrom rdf resource="#decanatura"/>
    </owl Restriction>
  </rdfs subClassOf>
</owl Class>
<owl Class rdf ID="area_desconcentrada">
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty>
        <owl ObjectProperty rdf about="#dependeDe"/>
      </owl onProperty>
      <owl allValuesFrom rdf resource="#decanatura"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf
rdf resource="http //www w3 org/2002/07/owl# Thing"/>
  <rdfs subClassOf>
    <owl Restriction>
      <owl onProperty rdf resource="#es"/>
      <owl allValuesFrom rdf resource="#departamento"/>
    </owl Restriction>
  </rdfs subClassOf>
  <rdfs subClassOf rdf resource="#vicedecanatura"/>
</owl Class>
<owl Class rdf ID="auxiliar_docencia">
  <rdfs subClassOf>
    <owl Restriction>

```

```

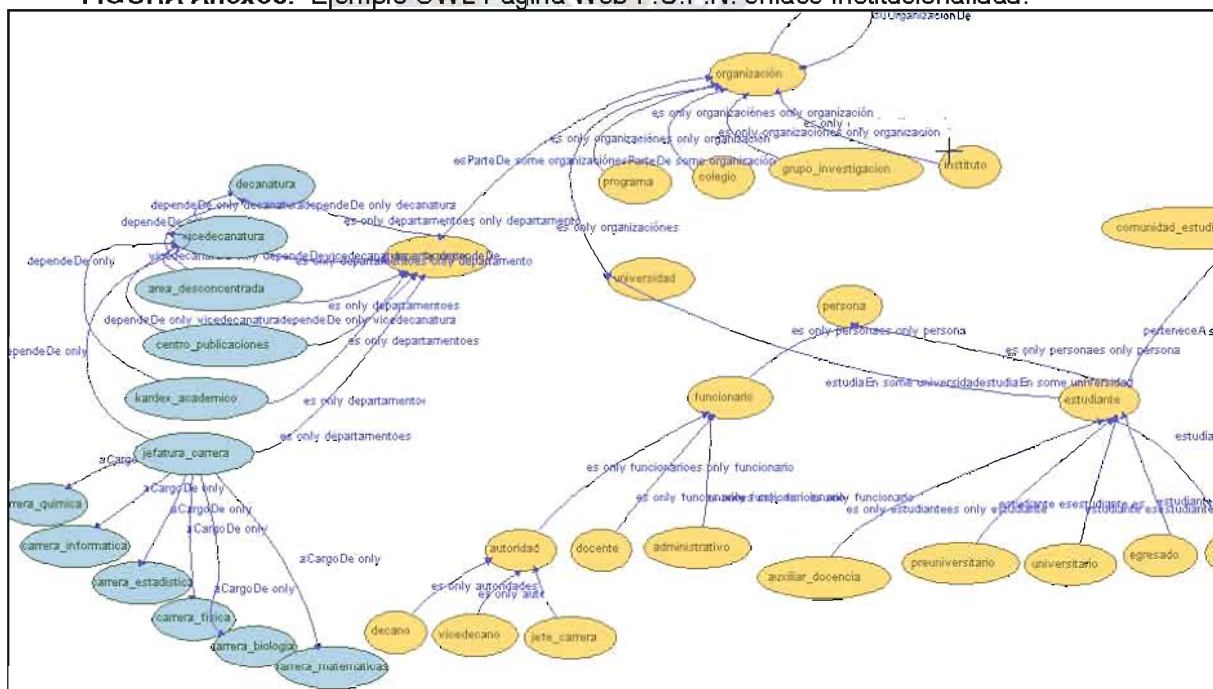
<owl onProperty rdf resource="#es"/>
<owl allValuesFrom rdf resource="#estudiante"/>
</owl Restriction>
</rdfs subClassOf>
<rdfs subClassOf
rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl Class rdf ID="universitario">
<rdfs subClassOf rdf resource="#estudiante"/>
<rdfs subClassOf
rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl Class rdf ID="vicedecano">
<rdfs subClassOf>
<owl Restriction>
<owl onProperty rdf resource="#es"/>
<owl allValuesFrom rdf resource="#autoridad"/>
</owl Restriction>
</rdfs subClassOf>
<rdfs subClassOf
rdf resource="http://www.w3.org/2002/07/owl#Thing"/>
</owl Class>
<owl ObjectProperty rdf about="#aCargoDe">
<owl inverseOf>
<owl ObjectProperty rdf about="#dependeDe"/>
</owl inverseOf>
</owl ObjectProperty>
<owl ObjectProperty rdf about="#dependeDe">
<owl inverseOf rdf resource="#aCargoDe"/>
</owl ObjectProperty>
<owl ObjectProperty rdf ID="suOrganizacionDe"/>
</rdf RDF>

<!-- Created -->

```

[Fuente: Elaboración Propia]

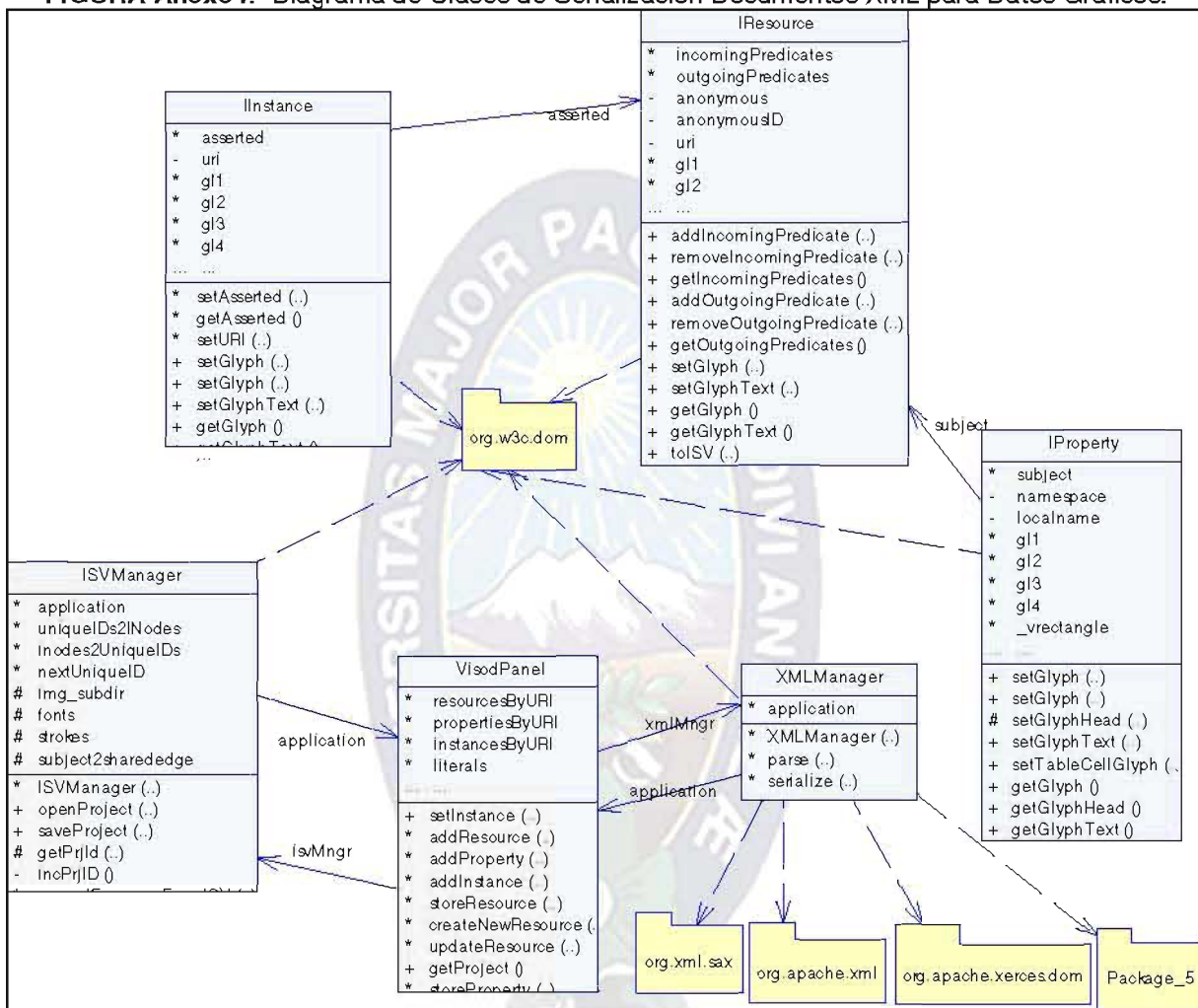
FIGURA Anexo3: Ejemplo OWL Página Web F.C.P.N. enlace Institucionalidad.



[Fuente: Elaboración Propia]

ANEXO 4: Diagrama de Clases que Intervienen en la Serialización de documentos XML para Almacenamiento de Datos Gráficos permanente.

FIGURA Anexo4: Diagrama de Clases de Serialización Documentos XML para Datos Gráficos.



[Fuente: Elaboración Propia]