

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN - TACNA

ESCUELA DE POSGRADO

**Maestría en Ingeniería de Sistemas e Informática - Administración de
Tecnologías de Información**

**DESARROLLO DE UN SISTEMA DE INFORMACIÓN ONTOLÓGICO
COMO SOPORTE A LA ADMINISTRACIÓN ACADÉMICA
DE LA FACULTAD DE CIENCIAS**

TESIS

PRESENTADA POR:

ING. EDGAR AURELIO TAYA ACOSTA

Para optar el Grado Académico de:

**MAESTRO EN CIENCIAS (*MAGÍSTER SCIENTIAE*)
CON MENCIÓN EN INGENIERÍA DE SISTEMAS E INFORMÁTICA -
ADMINISTRACIÓN DE TECNOLOGÍAS DE INFORMACIÓN**

TACNA - PERÚ

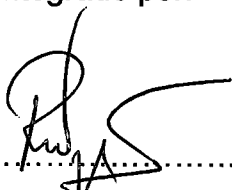
2011

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN – TACNA
ESCUELA DE POSGRADO

**MAESTRÍA EN INGENIERÍA DE SISTEMAS E INFORMÁTICA –
ADMINISTRACIÓN DE TECNOLOGÍAS DE INFORMACIÓN.**

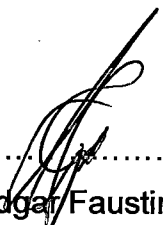
**DESARROLLO DE UN SISTEMA DE INFORMACIÓN ONTOLÓGICO
COMO SOPORTE A LA ADMINISTRACIÓN ACADÉMICA DE LA
FACULTAD DE CIENCIAS**

Tesis sustentada y aprobada por unanimidad el 11 de Marzo del 2011;
estando el jurado calificador integrado por:

PRESIDENTE : 
Mgr. Rolando Wilman Vásquez Jaico

SECRETARIO : 
Mgr. Claudio David Caipa Mamani

MIEMBRO : 
Dra. Liliana del Carmen Lanchipa Bergamini

ASESOR : 
Mgr. Edgar Faustino Taya Osorio

AGRADECIMIENTO

A mi madre Gladys por su sublime amor y ternura.

A mi gran maestro, amigo y padre Edgar.

A mis amigos de verdad por su desinteresado apoyo.

A la Escuela de Posgrado y sus profesores por darme un espacio de
formación.

DEDICATORIA

A Dios todopoderoso, ya que en sus manos está mi espíritu y vida.

A mis hermosos hijos Mao y Tian por ser mi fortaleza y vida, gracias por darme un espacio de su sagrado tiempo para poder concretar mis estudios.

A mi bella esposa Leyla a quien amo y amaré por siempre, estoy muy agradecido por su comprensión y admirable don de desprendimiento.

A Gian Marco Flores Sánchez aquel angelito que está en el cielo esperando por nosotros.

A mi querida alma mater cuyos pasillos recorrí desde niño, luego como estudiante, trabajador y ahora docente.

CONTENIDO

	Pág.
PÁGINA DEL JURADO	ii
AGRADECIMIENTO	iii
DEDICATORIA	iv
CONTENIDO	v
ÍNDICE DE FIGURAS	xi
ÍNDICE DE TABLAS	xiv
ÍNDICE DE CUADROS	xv
ÍNDICE DE ANEXOS	xvi
RESUMEN	xvii
ABSTRACT	xviii
INTRODUCCIÓN	1
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	4
1.1. DESCRIPCIÓN DEL PROBLEMA	4
1.2. FORMULACIÓN DEL PROBLEMA	5
1.2.1. Problema principal	6
1.2.2. Problemas específicos	6

1.3. FORMULACIÓN DE OBJETIVOS	7
1.3.1. Objetivo general	7
1.3.2. Objetivos específicos	7
1.4. IMPORTANCIA Y JUSTIFICACIÓN DE LA INVESTIGACIÓN	8
1.4.1. Importancia	8
1.4.2. Justificación	8
1.5. FORMULACIÓN DE LA HIPÓTESIS	9
1.5.1. Hipótesis general	9
1.5.2. Hipótesis específicas	9
CAPÍTULO II : MARCO TEÓRICO	10
2.1. ANTECEDENTES DEL ESTUDIO	10
2.1.1. Proyecto GALEN (2006). Unión Europea. Cooperación GALEN-CORBAMed	10
2.1.2. Framework : Adoptando reglas y ontologías en los servicios de búsqueda web. Hun, H & Du, X. (2005)	11
2.1.3. Proyecto SAW. Lozano (2006)	12
2.1.4. Vocabulario en XML para la gestión catastral de inmuebles urbanos. Mendoza Da Mata (2004)	14

2.1.5. Ontologías para servicios web semánticos de información de tráfico: Descripción y herramientas de explotación. Samper (2005)	15
2.2. BASES TEÓRICAS	16
2.2.1. ¿Qué es una ontología?	20
2.2.2. Ventajas del uso de ontologías	25
2.2.3. ¿Cuáles son los principales componentes de una ontología?	28
2.2.4. Modelo	29
2.2.5. Principio para el diseño de ontologías	36
2.2.6. Metodologías y métodos para el desarrollo de ontologías	37
2.2.7. Proceso de desarrollo de ontologías	37
2.2.8. Desarrollo de ontologías métodos y metodologías	43
2.2.9. Método Cyc	44
2.2.10. Método de Uschold y King	47
2.2.11. Metodología de Grüninger y Fox	48
2.2.12. Metodología KACTUS	50
2.2.13. Método basado en SENSUS	51
2.2.14. Metodología Methontology	52
2.2.15. Metodología On-To-Knowledge	56

2.2.16. Comparando métodos y metodologías de desarrollo de ontologías	62
2.2.17. Lenguajes para el desarrollo de ontologías	67
2.2.18. El lenguaje OWL	86
2.2.19. Sublenguajes de OWL	89
2.2.20. Lenguajes de consultas o de recuperación de una ontología	92
2.2.21. SPARQL: necesidad de un lenguaje de consulta	95
2.2.22. ¿Qué es SPARQL?	95
2.2.23. SPARQL: sintaxis y cláusulas	97
2.2.24. Ejemplos de consultas en SPARQL	102
2.2.25. SPARQL/Update (SPARUL)	104
2.2.26. Herramientas de ontologías	105
2.2.27. Protégé	107
2.2.28. Aplicaciones de las ontologías	112
2.2.29. Construyendo una ontología para la web	113
2.2.30. Los sistemas y sus componentes	116
2.2.31. Los datos y la información	118
2.2.32. Los sistemas de información	119
2.2.33. Características de los sistemas de información modernos	121

2.2.34. Seis tipos principales de sistemas	121
2.3. TÉRMINOS BÁSICOS	124
CAPÍTULO III : MARCO METODOLÓGICO	128
3.1 TIPO Y DISEÑO DE LA INVESTIGACIÓN	128
3.1.1 Tipo de investigación	128
3.1.2 Tratamiento y análisis de la información	129
3.1.3 Diseño de la investigación	129
3.1.4 Diseño experimental o no experimental	129
3.2 POBLACIÓN Y MUESTRA	130
3.2.1 Población	130
3.2.2 Muestra	130
3.3 OPERACIONALIZACIÓN DE VARIABLES	130
3.3.1 Variables de estudio	130
3.3.2 Operacionalización de variables	131
3.3.3 Clasificación de las variables	131
3.4 TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN	132
3.4.1 Técnica de recolección de datos	132
3.4.1.1 Métodos interactivos	132
3.4.1.2 Métodos no intrusivos	132

3.4.2	Instrumento de recolección de datos	132
3.4.3	Construcción de la encuesta	133
CAPÍTULO IV : RESULTADOS Y DISCUSIÓN		135
4.1.	PRESENTACIÓN, ANÁLISIS E INTERPRETACIÓN DE LOS DATOS	135
4.1.1.	Resultados de la encuesta	135
4.1.2.	Contrastación de hipótesis	152
	CONCLUSIONES	158
	RECOMENDACIONES	160
	REFERENCIAS BIBLIOGRÁFICAS	162
	ANEXOS	169

ÍNDICE DE FIGURAS

	Pág.
Figura 1 : Una parte de la estructura jerárquica de la ontología en el dominio del “cáncer”	25
Figura 2 : Representación estructural de una ontología	30
Figura 3 : Propiedades de una ontología	31
Figura 4 : Propiedad Simétrica	32
Figura 5 : Propiedad Transitiva	33
Figura 6 : Propiedad Funcional	33
Figura 7 : Propiedad Reflexiva	34
Figura 8 : Propiedad Equivalente	35
Figura 9 : Proceso de desarrollo de la ontología	39
Figura 10 : Proceso propuesto por el método Cyc.	44
Figura 11 : Proceso de desarrollo y ciclo de vida de Methontology	54
Figura 12 : Tareas de las actividades de conceptualización de acuerdo a METHONTOLOGY	56
Figura 13 : Proceso de On-To-Knowledge	57
Figura 14 : Lenguaje de marcas de Ontologías	67
Figura 15 : Ejemplo de los dos tipos de sintaxis de RDF	80

Figura 16	: Acceso a Datos	95
Figura 17	: Ejemplo de comparación entre SQL y SPARQL	96
Figura 18	: Ejemplo de uso de BASE y PREFIX	97
Figura 19	: Ejemplo de consulta en SPARQL	102
Figura 20	: Ejemplo de consulta en SPARQL usando FILTER	103
Figura 21	: Ejemplo de consulta en SPARQL usando FILTER	104
Figura 22	: Ejemplo de consulta usando FILTER	105
Figura 23	: Editor de clases de Protégé	109
Figura 24	: Propiedades de tipo objeto de la Ontología Pizza	110
Figura 25	: Individuos de la clase Country	111
Figura 26	: Una porción de la ontología para el tema de artes	115
Figura 27	: Modelo general de un sistema	118
Figura 28	: Componentes de los sistema de información	120
Figura 29	: Resumen eficiencia sin software	139
Figura 30	: Resumen usabilidad sin software	143
Figura 31	: Resumen eficiencia con software	147
Figura 32	: Resumen usabilidad con software	151
Figura 33	: Modelo de clases de la ontología "Onto_GestiónAcadémica" en Protégé	186
Figura 34	: Objetos de propiedades de la ontología "Onto_GestiónAcadémica"	187

Figura 35	: Definición de los tipos de datos de la ontología "Onto_GestiónAcadémica"	188
Figura 36	: Instancia de Profesor con sus propiedades y tipos de datos de la ontología "Onto_GestiónAcadémica"	189
Figura 37	: Visualización de la ontología "Onto_GestiónAcadémica" utilizando Jambalaya de Protégé	190
Figura 38	: Segmento de código en OWL de la ontología "Onto_GestiónAcadémica" generado por Protégé.	191
Figura 39	: Registros de la tabla statements de la base de datos onto_gestionacademica creada con RAP en MySQL	192
Figura 40	: Interfaz del prototipo del sistema de información ontológico	193
Figura 41	: Búsqueda por clases de notas	194
Figura 42	: Código fuente en PHP y SPARQL de la búsqueda por clases.	195

ÍNDICE DE TABLAS

	Pág.
Tabla 1 : El resumen de las estrategias de construcción	65
Tabla 2 : Soporte tecnológico	65
Tabla 3 : Resumen del proceso de desarrollo de ontologías	66
Tabla 4 : Resumen de las características de lenguajes para la construcción de ontologías	83
Tabla 5 : Sintaxis de OWL-Protégé	111
Tabla 6 : Atributos de la Información	119

ÍNDICE DE CUADROS

	Pág.
Cuadro 1 : Modelo RDF	78
Cuadro 2 :Resumen de las características de los seis tipos de sistemas de información	123
Cuadro 3 : Eficiencia sin software, ponderación de Likert	136
Cuadro 4 : Resumen eficiencia sin software	138
Cuadro 5 : Usabilidad sin software, ponderación de Likert	140
Cuadro 6 : Resumen usabilidad sin software	143
Cuadro 7 : Eficiencia con software, ponderación de Likert	144
Cuadro 8 : Resumen eficiencia con software	147
Cuadro 9 : Usabilidad con software, ponderación de Likert.	148
Cuadro 10 : Resumen usabilidad con software	151
Cuadro 11 : Resumen eficiencia en dos grupos	155
Cuadro 12 : Resumen usabilidad en dos grupos	156
Cuadro 13 : Puntajes y promedio de los expertos	175
Cuadro 14 : Resumen y calculo de X^2	176

ÍNDICE DE ANEXOS

	Pág.
ANEXO A : ENCUESTA	169
ANEXO B : VALIDACIÓN DEL INSTRUMENTO	175
ANEXO C : ORGANIZACIÓN DE LA FACULTAD DE CIENCIAS	178
ANEXO D : DESARROLLO DE APLICACIÓN	185

RESUMEN

Este trabajo de tesis presenta una minuciosa revisión de la teoría relacionada con el uso de las Ontologías en el desarrollo de vocabularios como soporte semántico en la construcción de sistemas homogéneos y estándares que podrán ser revisados y reutilizados en repositorios de ontologías.

Como caso práctico se implementó un prototipo de un sistema de información para la administración académica de la facultad de Ciencias, ya que actualmente no se tiene una política de estandarización de sistemas en nuestra Universidad y cada unidad, Facultad u oficina compra o desarrolla sus propios sistemas sin tomar en cuenta la interoperabilidad y la reutilización de vocabularios estándares que permitan un mejor integración de los sistemas y enriquecimiento de sus elementos conceptuales para un posterior refinamiento y rediseño.

Esta propuesta que presentamos como tesis se construirá utilizando Protégé como herramienta de diseño de la Ontología y generación de nuestro código en OWL y la plataforma de diseño RAP (RDF API for PHP) para el desarrollo de la aplicación.

ABSTRACT

This thesis presents a thorough review of the theory related to the use of ontologies in the development of vocabulary and semantic support in the construction of homogeneous systems and standards may be revised and reused in ontology repositories.

As a case study was implemented in a prototype information system for academic administration of the Faculty of Sciences, and currently do not have a policy of standardization in our university system and each unit, School or office purchase or develop their own systems without taking into account the interoperability and reuse of standard vocabularies to enable better integration of systems and enhancement of its conceptual elements for further refinement and redesign.

This presented as a thesis proposal will be built using Protégé as a tool for ontology design and generation of our code in OWL and design platform RAP (RDF API for PHP) for application development.

INTRODUCCIÓN

El Desarrollo de Sistemas de Información (SI), normalmente se hace en diferentes contextos, con distintos puntos de vista y suposiciones acerca del dominio de estudio. Esto provoca problemas de comunicación por falta de entendimiento compartido. Por otra parte, el diseñador de los SI necesita usar representaciones, tan generales como sea posible, para responder a los atributos de calidad del software y aumentar al máximo la posibilidad de reusabilidad. Pero, al mismo tiempo, las representaciones deben corresponderse, lo más estrechamente posible, con las cosas y los procesos que ellos representan. Es así como, los aspectos relacionados con la gestión de datos, información y conocimiento son cuestiones efectivamente ontológicas.

Tradicionalmente, la ontología se define como "la ciencia de lo que es, de los tipos y las estructuras de objetos, propiedades, eventos, procesos y relaciones en cada área de la realidad" (Gruber, 1993).

En la presente tesis presentamos una propuesta de desarrollo de un prototipo de sistema de información ontológico para la administración académica de la Facultad de Ciencias, específicamente un prototipo que

nos permitirá tener una aproximación de cómo se podría concretar la utilidad de las ontologías en un escenario cercano a nosotros que es la Universidad Nacional Jorge Basadre Grohmann de Tacna, en especial la Facultad de Ciencias.

En el primer capítulo del trabajo se presenta la problemática formulada a través de un diagnóstico, pronóstico y control al pronóstico para luego formular debidamente el problema, luego los objetivos que perseguimos con esta tesis, finalmente la hipótesis que validamos en el presente estudio.

En el segundo capítulo del trabajo se presenta una revisión de cinco trabajos antecedentes a éste, en donde se analizan rápidamente sus conclusiones, luego una muy trabajada y detallada revisión bibliográfica de algunos libros y artículos de buen nivel donde enfocan el tema de las Ontologías e Ingeniería Ontológica que nos permiten configurar un rico marco teórico que dará el debido soporte documental a nuestra tesis y finalmente un glosario de términos para familiarizarse con la terminología del presente trabajo.

En el tercer capítulo se muestra la metodología de investigación usada para formalizar la propuesta de tesis y darle el enfoque científico que amerita.

En el cuarto capítulo se muestran los resultados y discusión de los datos obtenidos con nuestro instrumento validado, a través de gráficos y tablas. También visualizamos la contrastación de nuestras hipótesis.

Finalmente se emiten las conclusiones de la tesis al igual que algunas recomendaciones para trabajos futuros.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

1.1. DESCRIPCIÓN DEL PROBLEMA

Actualmente el diseño de los pocos sistemas de información de la Universidad Nacional Jorge Basadre Grohmann de Tacna han seguido distintas concepciones y definiciones, esto ha generado una heterogeneidad que imposibilita la compatibilidad de dichos sistemas y hace que sean poco funcionales e intuitivos.

Esto a su vez evita tener acceso a información importante para el soporte de la administración académica, además los usuarios tienen bajos niveles de satisfacción con respecto a la modalidad intuitiva de la navegación en estos sistemas sobre la información que se les proporciona y los procesos académicos que les interesa.

Si esto continúa así se puede convertir en un inconveniente para la administración académica y en consecuencia debilitar mas la gestión universitaria ya que no tendremos un puente informativo

claro y bien definido donde se pueda encontrar toda la información que se necesita conocer del proceso académico de nuestra Facultad porque tanto los profesores, alumnos, administrativos y padres de familia no tendrán ninguna posibilidad de acceder a ésta.

De esta manera el diseño de las ontologías permite que, cada vez la descripción explícita de un determinado dominio se convierta en un vocabulario común y un entendimiento compartido entre personas, agentes de software y de esta manera se pueda reutilizar estos dominios (por ejemplo en otras Facultades y porque no en otras Universidades).

1.2. FORMULACIÓN DEL PROBLEMA

En tal sentido diseñaremos una ontología que nos permita desarrollar un prototipo de sistema de información ontológico para dar soporte a la administración académica y además publicar la ontología para poder ser reutilizada y compartida además de mejorada por otros desarrolladores.

1.2.1. Problema principal

Por lo tanto nos plantearemos la siguiente interrogante:

¿Cuál es el nivel de dependencia entre un sistema de información basado en ontologías y el soporte a la administración académica de la facultad de ciencias de la UNJBG?

1.2.2. Problemas específicos

p1: ¿Será posible determinar el nivel de dependencia de la ontología del dominio de aplicación con el grado de eficiencia en la administración académica de la facultad de ciencias?

p2: ¿Será posible medir el nivel de dependencia entre el producto de software basado en ontologías utilizando RAP (RDF API for PHP) y el grado de usabilidad en el sistema de administración académica de la facultad de ciencias?

1.3. FORMULACIÓN DE OBJETIVOS

1.3.1. Objetivo general

Determinar el nivel de dependencia entre el sistema de información basado en ontologías y el soporte a la administración académica de la Facultad de Ciencias.

1.3.2. Objetivos específicos

OE1: Determinar el nivel de dependencia de la ontología del dominio de aplicación con el grado de eficiencia la administración académica de la Facultad de Ciencias.

OE2: Determinar el nivel de dependencia entre la calidad de producto del sistema de información basado en ontologías utilizando RAP (RDP API for PHP) y el grado de usabilidad del sistema de administración académica de la Facultad de Ciencias.

1.4. IMPORTANCIA Y JUSTIFICACIÓN DE LA INVESTIGACIÓN

1.4.1. Importancia

La estandarización de concepciones en el desarrollo de sistemas de Información es crucial para asegurar la flexibilidad y adaptabilidad de estos y es un tema de investigación que recién entra en vigencia y que necesita muchos estudios para poder consolidarse como una herramienta semántica muy potente.

1.4.2. Justificación

Es necesario abordar este tema puesto que la Universidad Nacional Jorge Basadre Grohmann y específicamente la Facultad de Ciencias esta enrumándose al proceso de acreditación por lo cual necesita que sus procesos sean automatizados por completo, lo que permitirá una mejor administración de sus procedimientos. (Resoluciones, plan de estudios y autoevaluación)

1.5. FORMULACIÓN DE LA HIPÓTESIS

1.5.1. Hipótesis general

El sistema de información basado en Ontologías no tendrá independencia con el soporte a la administración académica de la facultad de Ciencias.

1.5.2. Hipótesis específicas

H1: La estandarización de la ontología del dominio de aplicación no tendrá independencia con el grado de eficiencia de la administración académica de la facultad de ciencias.

H2: El producto del sistema de información basado en ontologías utilizando RAP (RDP API for PHP) no tendrá independencia con el grado de usabilidad del sistema de administración académica de la facultad de Ciencias.

CAPÍTULO II

MARCO TEÓRICO

2.1. ANTECEDENTES DEL ESTUDIO

2.1.1. Proyecto GALEN. Unión Europea. Cooperación GALEN-CORBAMed. (2006)

Objetivo: GALEN, Arquitectura General para Enciclopedias y Nomenclaturas en Medicina, es un proyecto desarrollado y financiado por la Unión Europea que tiene como objetivo el desarrollo de herramientas y métodos para una construcción y mantenimiento de clasificaciones de procedimientos quirúrgicos basados en ontologías para mejorar la búsqueda de Información de términos que sean sinónimos.

Conclusiones:

- Garantiza descripciones clínicas detalladas basadas en un modelo semántico terminológico.

- Los conceptos complejos se almacenan con una representación fija.
- Contiene herramientas lingüísticas que hacen posible el desarrollo de sistemas multilingües.
- Facilita el intercambio de datos clínicos entre sistemas con diferentes estructuras clasificadoras.
- Aumenta y expande los esquemas de codificación y clasificación ya existentes.

2.1.2. Framework : Adoptando reglas y ontologías en los servicios de búsqueda Web. Hun, H. & Du, X.(2005)

Objetivo: Este trabajo tuvo como finalidad proponer un Framework de búsqueda Web usando Ontologías y Reglas. El Framework usa Ontologías para tratar con dos problemas principales como son los sinónimos y la polisemia, por lo tanto la búsqueda tiende a ampliarse debido a la polisemia y a encogerse debido a los sinónimos.

Conclusiones:

- Proporcionan un vocabulario común y la definición de reglas para usar independientemente los servicios desarrollados.
- Logra acuerdos entre compañías, organizaciones que comparten servicios.
- Facilitan conversaciones entre agentes para coleccionar, procesar, fusionar e intercambiar información.
- Mejoran la búsqueda precisa a través de la búsqueda contextual usando conceptos de definiciones y relaciones entre ellos en lugar de palabras claves de relevancia estadísticas.

2.1.3. PROYECTO SAW. Lozano (2006)

Objetivo: Cuenta con diversas herramientas integradas que cooperan entre si y trabajan a distintos niveles para conseguir una accesibilidad real, por un lado ofrecen una herramienta para el diseñador de páginas web, EDITSAW, un editor que permite realizar anotaciones semánticas enfocadas a mejorar la accesibilidad de los contenidos de una página web. En

cuanto al usuario con discapacidad visual se ofrece NAVISAW, un navegador multimodal que incorpora síntesis y reconocimiento de voz e interacción a través de MOUSESAW, un gestor de correo accesible.

Conclusiones:

- Identifica los atributos de los elementos que componen las páginas web para hacerlas accesibles y representarlos en una ontología denominado ONTOSAW. El conocimiento que se representa en ontologías trata de recoger de manera perfecta y precisa los elementos de un dominio y está disponible para ser utilizable por varias aplicaciones en múltiples propósitos y en los últimos años su uso está muy extendido debido a la iniciativa de la web semántica.

2.1.4. Vocabulario en XML para la gestión catastral de inmuebles urbanos. Mendonça Da Mata. (2004)

Objetivo: Su objetivo principal fue el diseño e Implementación de un vocabulario en XML para la gestión catastral de inmuebles.

Conclusiones:

- Se logró una estandarización de la información y así la gestión catastral pueda realizar el intercambio de información entre las entidades involucradas y también se pueda crear y mantener actualizada una base de datos central con la información catastral nacional. Al adoptar este vocabulario como estándar en las entidades municipales e institucionales involucradas con la gestión catastral, se puede tener grandes beneficios en cuanto al mejor y rápido procesamiento de información, emisión de estadísticas, reportes más exactos y con información precisa y real.

2.1.5. Ontologías para servicios web semánticos de información de tráfico: Descripción y herramientas de explotación. Samper, J. (2005)

Objetivo: Facilitar información de tráfico al usuario, y que a su vez permita la gestión de la información, su tratamiento e intercambio, de manera eficaz entre los diferentes elementos que componen la arquitectura de servicios de información de tráfico, como los usuarios, aplicaciones, y proveedores de la información.

Conclusiones:

- Construcción de una infraestructura ontológica cuyo dominio queda enmarcado en la información sobre tráfico vial.
- Se ha expuesto un marco de trabajo para la conversión de portales Web convencionales de información en Servicios Web Semánticos

- Extensión a las metodologías de construcción de ontologías, previamente existentes, para plantear el proceso a seguir en la obtención de un modelo semántico formal a partir de un modelo de Entidad-Relación (ER).

2.2. BASES TEÓRICAS

Las ontologías son ahora ampliamente usadas en la Ingeniería del Conocimiento, Inteligencia Artificial y Ciencia de la Computación; en aplicaciones relacionadas a la gestión del conocimiento, procesamiento del lenguaje natural, comercio electrónico, inteligencia en la integración de la información, recuperación de información, integración de bases de datos, bioinformática, educación; y en campos emergentes nuevos como la Web Semántica.

En 1991, DARPA con el esfuerzo de compartir conocimiento visualizó una nueva forma de construir sistemas inteligentes. Ellos propusieron lo siguiente:

Construir sistemas basados en conocimiento hoy usualmente conlleva construir bases nuevas de conocimiento desde cero. Ello podría ser más bien hecho mediante el ensamblado de componentes reusables. Los desarrolladores de sistema entonces sólo necesitarían preocuparse por crear los conocimientos especializados y razonadores nuevos para la tarea específica de sus sistemas. Este sistema nuevo inter-operaría con otros existentes, usándolos para realizar una parte de su razonamiento. De este modo, el conocimiento declarativo, resuelve problemas técnicos y todos los servicios de razonamiento serán compartidos entre sistemas (Neches et al., 1999).

El conocimiento declarativo es modelado por medio de ontologías mientras el método de solución de problemas especifica mecanismos genéricos de razonamiento. Ambos tipos de componentes pueden ser vistos como entidades complementarias que pueden usarse para configurar nuevos sistemas basados en conocimientos de componentes reusables existentes. Nicola Guarino señaló los beneficios principales de usar ontologías en el tiempo del diseño:

Facilita al desarrollador practicar un nivel "más alto" de reutilización que es usualmente lo fundamental en la ingeniería del software (o sea la reutilización del conocimiento en lugar de la reutilización del software). Además, faculta al desarrollador a reusar y compartir conocimiento de dominio de aplicación usando un vocabulario común a través de plataformas heterogéneas del software. También faculta al desarrollador a concentrarse en la estructura del dominio y la tarea a la mano y le protege de ser molestado por los detalles de la implementación (Gómez, Fernández & Corcho, 2004).

Desde entonces el progreso considerable ha sido hecho para desarrollar las bases conceptuales para construir tecnología que permite reusar y compartir componentes de conocimiento.

Las ontologías y Métodos de Resolución de Problemas (PSMs) han sido creados para compartir y reusar conocimiento y razonar comportamiento a través de los dominios y las tareas.

El surgimiento de la Web Semántica ha marcado una nueva etapa en la evolución de ontologías (y PSMS). La Web Semántica es

una extensión de la Web actual en la que la información recibe significado bien definido, facilitando a las computadoras y las personas a trabajar en cooperación. Ésta se puede lograr mediante el uso compartido de componentes de conocimiento, y así ontologías y PSMs se han convertido en instrumentos clave en el desarrollo de la Web Semántica. Las ontologías representan el conocimiento del dominio estático y PSMs se utilizarán dentro de Servicios Web Semántica para modelar procesos de razonamiento y tratar con el conocimiento del dominio.

Ahora, presentaremos los conceptos fundamentales acerca de las ontologías, y al final de este capítulo, podremos responder preguntas tales como: ¿Qué es una ontología?, ¿Cuáles son los principales tipos de ontologías?, ¿Cuáles son sus principales componentes?, ¿Qué criterio de diseño deberíamos seguir para construir una ontología?, ¿Cuáles son las principales diferencias entre ontologías y base de datos?, ¿Cuáles son las principales diferencias entre modelar un dominio con ontologías y con técnicas de modelado de ingeniería de software? y finalmente ¿Cómo son las ontologías y las bases de conocimiento relacionados?

2.2.1 ¿Qué es una ontología?

FILOS. Parte de la metafísica, que trata del ser en general y de sus propiedades trascendentales.

El término ontología en informática hace referencia al intento de formular un exhaustivo y riguroso esquema conceptual dentro de un dominio dado, con la finalidad de facilitar la comunicación y la comparación de la información entre diferentes sistemas.

La ontología es una especificación formal y explícita de una conceptualización compartida.

- El término "conceptualización" implica que toda ontología desarrolla un modelo abstracto del dominio o fenómeno del mundo que representa. Dicho modelo abstracto se basa esencialmente en el empleo de conceptos, atributos, valores y relaciones.
- Con "especificación explícita" se quiere expresar que una ontología supone la descripción y representación de un

dominio concreto mediante conceptos, atributos, valores, relaciones, funciones, etc., definidas explícitamente. Las máquinas no pueden dar nada por supuesto o por obvio, y todo conocimiento -por básico que parezca, mientras sea necesario- debe ser explícitamente representado.

- El término "formal" alude al hecho de que cualquier representación (concepto, atributo, valor, etc.) ha de ser expresado en una ontología mediante un formalismo siempre idéntico, de manera que pueda ser reutilizado y leído por cualquier máquina independientemente del lugar o de la plataforma o idioma del sistema que lo emplee.
- Quizá el término más restrictivo e importante de los que figuran en la definición sea el de "compartida". En efecto, una ontología lo es cuando dicha conceptualización y su representación formal y explícita ha sido favorablemente acogida por todos los usuarios de la misma. Ello permite por una parte distinguir claramente las ontologías de las bases de datos (en las que también puede hablarse de conceptualización y especificación formal y explícita mediante conceptos, atributos y valores, pero en la que el

creador no tiene que lograr el consenso de nadie). Sin embargo, al mismo tiempo plantea una gran dificultad, pues en la práctica es imposible o casi imposible conseguir el consenso de todos los involucrados en un dominio específico (piénsese, por ejemplo, en una ontología sobre sanidad: ¿quién pone de acuerdo a médicos, pacientes, profesores, alumnos, gestores, etc., en la terminología, valores y relaciones en el dominio de la medicina?). De ahí que, en la práctica, se considere imposible desarrollar una ontología de carácter genérico o global, y sin embargo, se desarrollen ontologías en ámbitos mucho más restringidos, porque alcanzar aquí el consenso es factible (un banco para desarrollar servicios online para sus clientes, una empresa que desea una ontología sobre el conocimiento generado internamente por ella, etc.). Cuanto más genérico es el ámbito, en mayor medida el proceso hasta alcanzar el consenso se produce mediante una guerra de estándares inicial (varios organismos lanzan sus ontologías, esperando que cada una de ellas alcance el consenso de los demás), de las que surgen con el tiempo 2 ó 3 estándares de facto

por área o sector, normalizándose finalmente hasta alcanzar una variante con el consenso de todos (Contreras & Martínez, 2006).

Una ontología es usualmente definida como una jerarquía de conceptos correspondiendo a la estructura jerárquica de datos dentro de un cierto dominio. Esta estructura incluye las interrelaciones entre los conceptos, lo cual puede implicar relaciones directas, acíclicas, transitivas y reflexivas (Nigro, Gonzales & Xodo, 2008).

Es una especificación explícita de una conceptualización compartida.

Teniendo como base las definiciones anteriores se definió a la ontología como una base de conocimiento donde se definen explícitamente conceptos y las relaciones existentes entre cada uno de ellos de un determinado dominio del mundo real.

Por otro lado, las ontologías tratan de describir o proponer las categorías o relaciones básicas del ser y su existencia para definir las entidades y de qué tipo son. Por ejemplo las entidades o conceptos las comprenden: los sujetos, personas, ideas, cosas, etc.; esta ha de concebirse con un entendimiento común de un dominio que puede comunicarse entre humanos y sistemas computacionales.

El concepto de Ontología no solo es utilizado en la Web Semántica; si no se extiende a más campos de la informática como la inteligencia artificial o la ingeniería de software, pero su base es la misma en todos los campos.

Consideremos el ejemplo de una ontología en el dominio del "cáncer". Hay varios tipos de cáncer: de mama, de pulmón, de estómago, etc. Podemos decir que el de estómago es un tipo de cáncer, esto es, el "cáncer del estómago" es un subconcepto del concepto "cáncer". También podemos decir que ese "cáncer" es un subconcepto de "tumor", desde que éste es un tumor maligno (Nigro, Gonzales & Xodo, 2008).

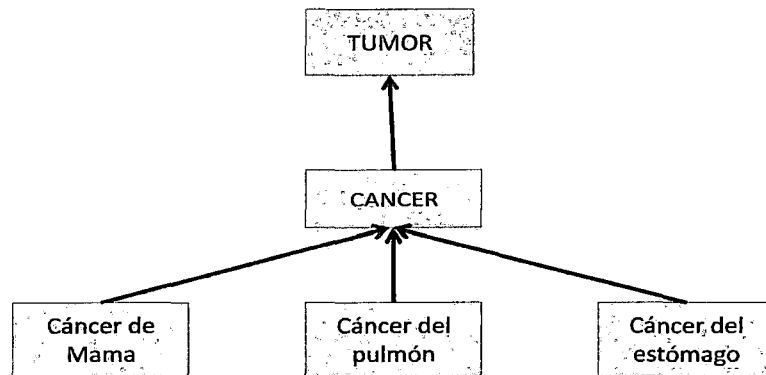


Figura 1: Una parte de la estructura jerárquica de la ontología en el dominio del “cáncer”.

Fuente : Nigro, Gonzales & Xodo (2008).

Por supuesto, las relaciones entre los conceptos en la jerarquía ontológica de un dominio dado están basadas en sus relaciones semánticas, las cuales son específicas para ese dominio.

2.2.2 Ventajas del uso de ontologías

Ahora que ya fue visto que es una ontología, es importante que se destaque las ventajas de su uso. A continuación se presenta una lista con las principales

ventajas de la utilización de ontologías en las Ciencias de la Computación.

Es posible hacer el relacionamiento del lenguaje de la ontología sin que con eso altere su conceptualización, o sea, una misma conceptualización puede ser expresada en varios idiomas.

Puede ser posible extender el uso de una ontología genérica de forma que se adecue a un dominio específico. Por ejemplo, si alguien necesita de una ontología sobre bicicletas para construir una aplicación y solo encuentra una ontología sobre el dominio genérico de vehículos, puede utilizar esa ontología extendiéndola para el dominio específico de la aplicación, que en el caso son bicicletas (Bulcão, 2006).

El enfoque declarativo utilizado por las ontologías permite describir un dominio sin algún compromiso con la implementación de un sistema de software, o sea, el

conocimiento modelado es independiente de la implementación.

Otro beneficio obtenido con el uso de las ontologías es la interoperabilidad entre sistemas de software. Sistemas que comparten la ontología librería, por ejemplo, poseen la misma interpretación de la semántica del vocabulario en cuestión, lo que les permite la integración transparente de servicios, como servicios de almacenamiento, clasificación, búsqueda y correlación de informaciones distribuidas en la Web.

Dado que la interoperabilidad requiere concordancia en cuanto a conceptos, las ontologías son públicamente compartidas como archivos distribuidos en la Web. Cabe pues a los desarrolladores saber la localización de cada ontología necesaria para sus sistemas.

2.2.3 ¿Cuáles son los principales componentes de una ontología?

Las ontologías tienen los siguientes componentes que servirán para representar el conocimiento de algún dominio.

Estos son:

- **Conceptos:** son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- **Relaciones:** representan la interacción y enlace entre los conceptos del dominio. Suelen formar la taxonomía del dominio. Por ejemplo: parte-de, subclase-de, conectado-a, parte-exhaustiva-de, etc.
- **Funciones:** son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de función que considera varios elementos de la ontología. Por ejemplo: categorizar clase, asignar fecha, etc.
- **Instancias:** se utilizan para representar objetos determinados de un concepto.
- **Reglas o Axiomas:** son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la

ontología. Por ejemplo: "Si A y B son de la clase C, entonces A no es subclase de B", "Para todo A que cumpla la condición B1, A es C", etc. Los axiomas, junto con la herencia de conceptos, permiten inferir conocimiento que no esté indicado explícitamente en la taxonomía de conceptos.

2.2.4 Modelo

Las informaciones contenidas en una ontología son modeladas a través de clases, que son los conceptos del dominio, siendo organizadas en jerarquías, definiendo relaciones de generalización (subclases → superclase) y especialización (superclase → subclase).

La figura 2, ilustra un diagrama conteniendo la descripción básica de una ontología y sus elementos básicos.

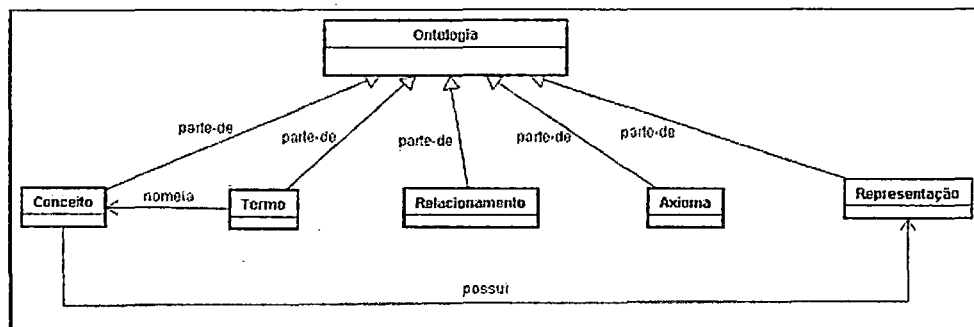


Figura 2 : Representación estructural de una ontología.

Fuente : Timm, (2008)

Sin embargo, solamente las clases no proveen información suficiente para responder a las preguntas de la competencia del ámbito. Una vez definidas las clases, se debe describir la estructura interna de los conceptos. Para eso deben ser definidas también las propiedades (atributos y relaciones), ya que cada clase es caracterizada por un conjunto de atributos y relaciones.

Las propiedades de una clase son heredadas por sus subclases, por lo tanto las propiedades solamente necesitan ser definidas en uno de los niveles de la jerarquía.

La figura 3 muestra la jerarquía de propiedades, donde, las propiedades son especializadas en **Propiedad Objeto** y **Propiedad de Tipo de Datos**.

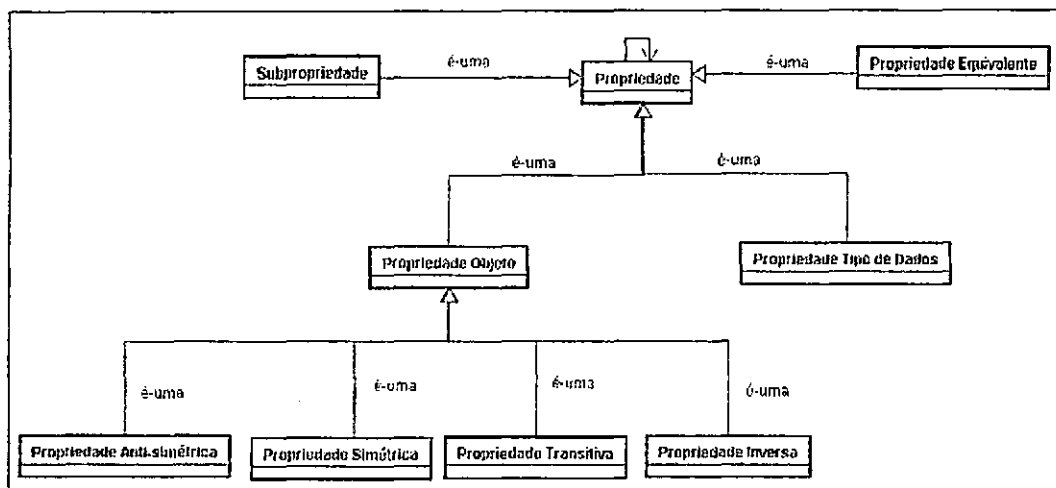


Figura 3: Propiedades de una ontología

Fuente : Timm, (2008)

Una propiedad es declarada como Propiedad Objeto cuando tiene el papel de relacionar una clase con otra clase.

La Propiedad de Tipo de Datos se diferencia de la Propiedad Objeto por utilizar una variable para representar cualquier cosa en el dominio abordado. En este tipo de propiedad, también es necesario definir el dominio al cual

ella pertenece y su valor que, no será una clase, sino un elemento del tipo string, boolean, int, entre otros.

Además, las Propiedades Objeto son especializadas en Simétrica, Transitiva, Funcional e Inversa, que son definidas a continuación.

Propiedad Simétrica: Una propiedad R es simétrica cuando, si x se relaciona con y, y se relaciona con x, así: si para cada $x \in E$; $R(x, y)$ implica $R(y,x)$, conforme muestra la figura 4.

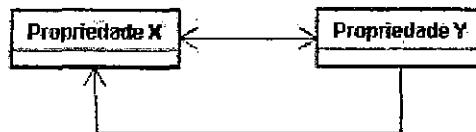


Figura 4: Propiedad Simétrica

Fuente : Timm, (2008)

Propiedad Transitiva: Una propiedad R es transitiva cuando, x está relacionado con y e y está relacionado con z, entonces x se relaciona con z, así; si, para cada $x, y, z \in E$;

existe $R(x, y)$ y $R(y, z)$ implica $R(x, z)$, conforme muestra la figura 5.

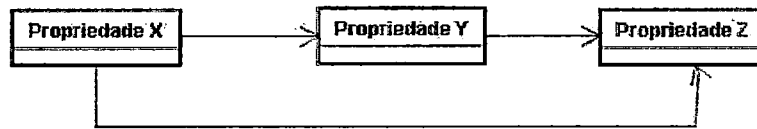


Figura 5: Propiedad Transitiva

Fuente : Timm, (2008)

Propiedad funcional: Una propiedad funcional es aquella en que se tiene un valor único de y para cada instancia de x . Por lo tanto, no tiene más que el único valor para cada individuo. Una propiedad con esta característica tiene cardinalidad mínima 0 y cardinalidad máxima 1; así: si , para cada $x \in W$, existe un único $y \in W$ tal que $R(x, y)$, conforme muestra la figura 6.

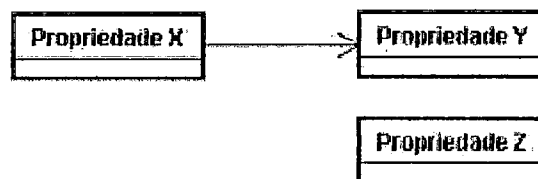


Figura 6: Propiedad Funcional

Fuente : Timm,(2008)

Propiedad Inversa: Las propiedades tienen un sentido, en el caso de la inversa, este sentido puede ser dado de ambos lados, así para cada $P1(x, y)$ existe un inverso $P2(y, x)$ y viceversa.

Las propiedades también pueden poseer subpropiedades, poseer relaciones de equivalencia y ser reflexivas.

Propiedad Reflexiva: Cuando todo elemento de E se relaciona con el mismo, así: si para cada $x \in E$, existe $R(x, x)$ en E , conforme muestra la figura 7.

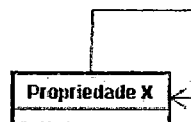


Figura 7: Propiedad Reflexiva.

Fuente : Timm, (2008).

Propiedad Equivalente: Una propiedad es equivalente si posee las propiedades reflexiva, transitiva y simétrica, conforme muestra la figura 8.

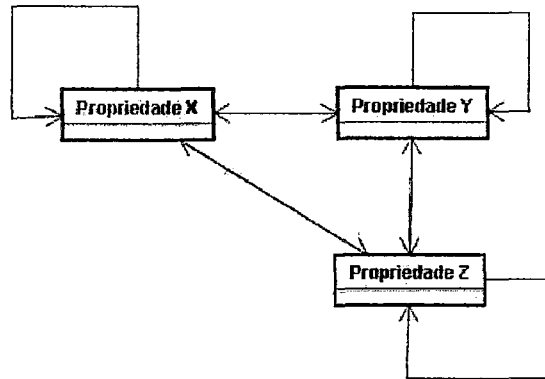


Figura 8: Propiedad Equivalente

Fuente : Timm, (2008).

Existen otras propiedades interesantes: Anti-simétrica, Irreflexiva, Asimétrica y Transitiva.

Cada propiedad es modelada por un conjunto de restricciones que definen los valores que pueden ser asumidos, los cuales describen o limitan el conjunto de valores posibles para una propiedad.

Las restricciones más comunes son la cardinalidad del valor de la propiedad sus límites, strings, números, elementos de un conjunto, etc.

2.2.5. Principio para el diseño de ontologías

Esta sección resume algunos criterios del diseño y un conjunto de principios que han sido comprobados útiles en el desarrollo de ontologías. Según, los principios del diseño de ontología son criterios objetivos para la guía y evaluación del diseño de las mismas. Al momento de diseñar una ontología debemos tener en cuenta 5 cuestiones de finalidad claves:

- **Claridad:** Una ontología debe poder comunicar de manera efectiva el significado de sus términos. Las definiciones serán lo más objetivas posibles y deben explicarse también en lenguaje natural.
- **Coherencia:** Una ontología debe permitir hacer inferencias que sean consistentes con las definiciones.
- **Extensibilidad:** Se debe anticipar nuevos usos para así poder permitir extensiones y especializaciones.
- **Especificidad:** Se debe especificar a nivel de conocimiento, sin que dependa de una codificación particular a nivel de símbolo.

- **Precisión:** Debe hacerse la menor cantidad de suposiciones acerca del mundo modelado.

Algunos otros criterios han resultado ser útiles en el diseño de ontología, tal como la estandarización de nombres, que propone usar la misma convención de nombres para términos relacionados, a fin de facilitar la comprensión de la ontología (Calero, Ruiz & Piattini, 2006).

2.2.6. Metodologías y métodos para el desarrollo de ontologías

Como cualquier pieza de software, se puede aplicar metodologías de desarrollo de software a la construcción de ontologías.

2.2.7. Proceso de desarrollo de ontologías

El proceso de desarrollo de ontología se refiere a las actividades que tienen que ser realizadas cuando se construyen ontologías. Pueden estar clasificadas en las tres categorías presentadas en la figura 9.

Las **actividades de gestión de ontología** incluyen planificación, control y aseguramiento de la calidad. La actividad de planificación identifica las tareas a ser realizadas, su distribución, el tiempo y recursos necesarios para su terminación.

Esta actividad es esencial para las ontologías que se usan almacenadas en librerías o para aquellas que requieran un nivel más alto de abstracción y generalidad. La actividad de control garantiza que las tareas programadas sean completadas de la manera deseada a ser realizadas. Finalmente, la actividad de aseguramiento de la calidad asegura que la calidad de todos y cada una de las salidas del producto (la ontología, informática y la documentación) sea satisfactoria.

Las **actividades orientadas al desarrollo de ontologías** son agrupadas, como se presenta en la figura 9, en las actividades de pre-desarrollo, desarrollo y post-desarrollo. Durante el **pre-desarrollo**, un estudio del ámbito identifica el problema para ser solucionado con la ontología,

las aplicaciones donde la ontología será integrada, etc. También durante el pre-desarrollo, el estudio de factibilidad responde preguntas como: "¿Es posible construir la ontología?"; "¿Es adecuado construir la ontología?"; etc.

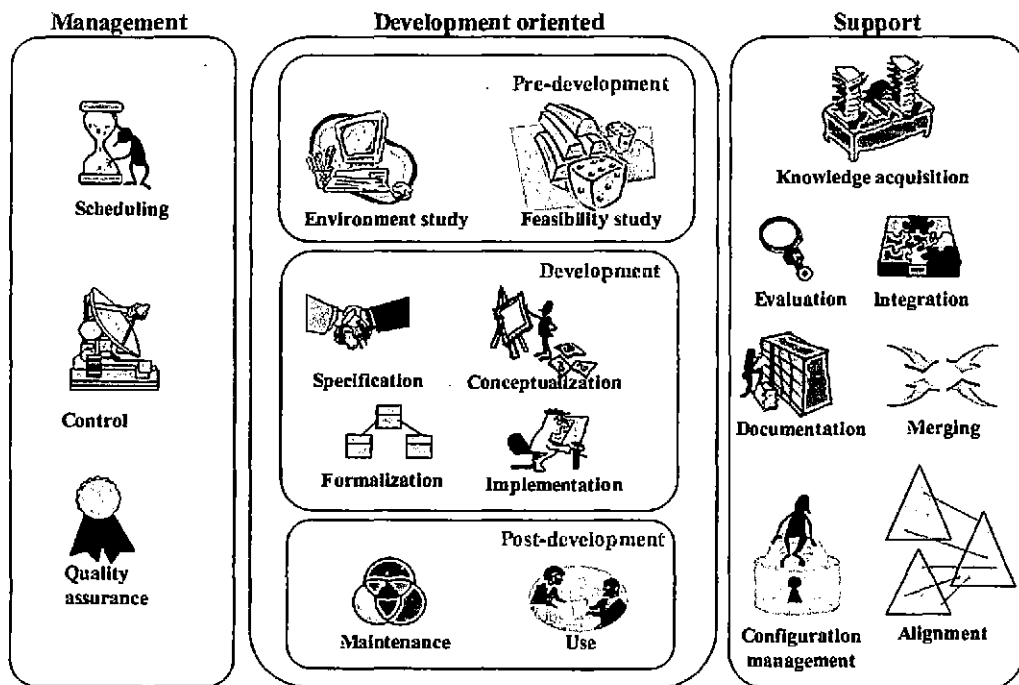


Figura 9: Proceso de desarrollo de la ontología

Fuente : Calero, Ruiz & Piatinni (2006)

Una vez en el **desarrollo**, la actividad de especificación declara por qué la ontología está siendo construida, cuáles son sus usos pretendidos y quiénes son

los usuarios finales. La actividad de conceptualización estructura el conocimiento de dominio como los modelos significativos en el nivel de conocimiento ya sea desde cero o reusando un modelo existente. En este último caso, las actividades relacionadas recortan ramas de las taxonomías existentes, expandiendo la cobertura de ontologías con la adición de conceptos nuevos en los niveles superiores de sus taxonomías, o especializando ramas que requieren más granularidad. Dado que la actividad de conceptualización es independiente del lenguaje de implementación, permite modelar ontologías de acuerdo al criterio mínimo del diseño de inclinación de codificación. La actividad de formalización transforma el modelo conceptual en un modelo formal o semicomputable. La actividad de implementación construye modelos computables en un lenguaje de ontología.

Durante el **post-desarrollo**, la actividad de *mantenimiento* actualiza y corrige la ontología si es necesario. También durante el post-desarrollo, la ontología es **(re)usada** por otras ontologías u otras aplicaciones. La actividad de evolución consta de manejar cambios de

ontología y sus efectos creando y manteniendo variantes diferentes de aquella, tomando en cuenta que puede ser usado en diferentes ontologías y aplicaciones.

Finalmente, las **actividades de soporte de ontologías** incluyen una serie de actividades que pueden ser realizadas durante el desarrollo orientado a actividades, sin las cuales la ontología no podría ser construida. Estas actividades incluyen la *adquisición de conocimiento, evaluación, integración, unión, alineamiento, documentación y gestión de la configuración*. La finalidad de la *actividad de adquisición de conocimiento* es adquirir conocimientos de expertos en un dominio dado o a través de algún tipo de proceso semiautomático, el cual es llamado aprendizaje de ontología. La *actividad de evaluación* hace un juicio técnico de las ontologías, de sus ambientes informáticos asociados, y de la documentación. Este juicio es hecho con relación a un esquema de referencia durante cada etapa y entre etapas del ciclo de vida de la ontología. La *actividad de integración* es requerida al construir una ontología nueva reusando otras ontologías ya disponibles. Otra actividad de soporte es la

unión, que consiste en obtener una ontología nueva a partir de varias otras en el mismo dominio. La ontología resultante puede unificar conceptos, terminologías, definiciones, restricciones, etc., de todas las fuentes de ontologías.

La unión de dos o más ontologías puede ser efectuada ya sea en el tiempo de ejecución o el tiempo del diseño. La *actividad de alineación* establece diferentes tipos de mapeos (o los eslabones) entre las involucradas. Por lo tanto esta opción conserva las ontologías originales y no las une. La *actividad de documentación* detalla, claramente y exhaustivamente, todos y cada uno de las etapas completadas y productos generados. La *actividad de gestión de la configuración* registra todas las versiones de la documentación y el código de la ontología para controlar los cambios. La actividad de multilingüismo consiste en trazar mapas de ontologías encima de descripciones formales de conocimiento lingüístico. Usualmente no ha sido considerado como una actividad de soporte de ontología, pero se ha vuelto más relevante en el contexto de ontologías enlazadas en red disponibles en la Web Semántica.

Como podemos ver, el proceso de desarrollo de ontología no identifica el orden en la cual las actividades deberían ser ejecutadas. Éste es el rol del ciclo de vida de la ontología, lo cual identifica cuando las actividades deberían ser efectuadas; esto es, identifica el conjunto de etapas a través de las cuales la ontología cambia durante su tiempo de vida, describe qué actividades deben ser realizadas en cada etapa y cómo las etapas están relacionadas (la relación de precedencia, regreso, etc.) (Calero, Ruiz & Piattini, 2006).

2.2.8. Desarrollo de ontologías métodos y metodologías

Tomando como referencia a Gómez, Fernández & Corcho, 2004, ahora hablaremos acerca de los métodos y las metodologías, los avances de los que se ocupan el método Cyc, el método Uschold y King, la metodología Grüninger y Fox, el acercamiento KACTUS, METHONTOLOGY, el método SENSUS, y la metodología del On-To-Knowledge.

Estas metodologías deberían ser adaptadas a las particularidades especiales que tienen las ontologías, pues no existe una metodología estándar. Diferentes proyectos reales de desarrollo de ontologías han dado lugar a diversas aproximaciones, las cuales menciono a continuación:

2.2.9. Método Cyc

Puede considerarse como una ontología porque puede ser utilizado como un sustrato para la construcción de diferentes sistemas inteligentes que puedan comunicarse e interactuar.

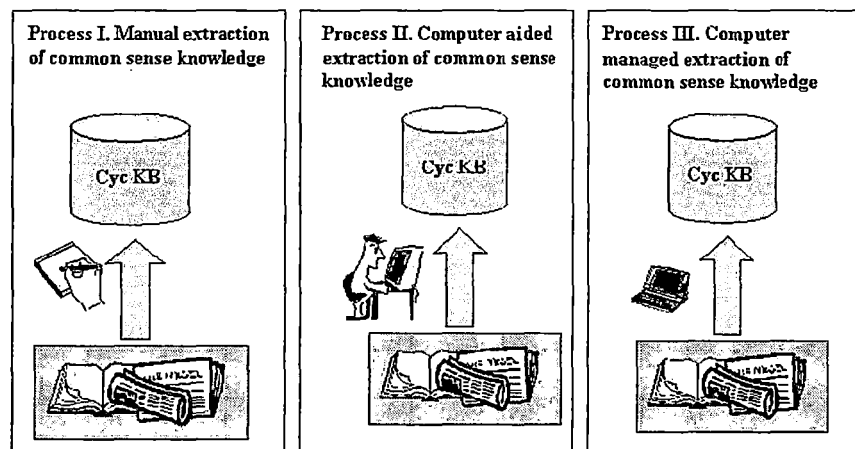


Figura 10: Proceso propuesto por el método Cyc.

Fuente : Gómez, Fernández & Corcho, (2004)

- **Proceso I. Codificación Manual** de artículos y piezas de conocimiento.
- Codificación de los conocimientos necesarios para comprender los libros y periódicos. Esto no significa la codificación del contenido de dichas obras, pero la búsqueda y representación de los conocimientos de sentido común que los escritores de esos artículos asumieron y sus lectores ya poseían.
- Examen de los artículos que son increíbles, por ejemplo, un documento que dice que un avión estaba volando un año sin reabastecerse. El propósito de este examen es el estudio de las razones que hace de algunos artículos increíbles.
- Identificación de preguntas que "nadie" debería ser capaz de responder con tan solo leer el texto. El KB aumentada para poder responder a esas preguntas.
- **Proceso II. Codificación de conocimientos asistido por herramientas** utilizando los conocimientos ya almacenados en la KB Cyc.

- Este segundo proceso se puede realizar cuando las herramientas para el análisis de lenguaje natural y máquinas de aprendizaje pueden usar el suficiente conocimiento y sentido común para la búsqueda de nuevos conocimientos.
- **Proceso III.** Codificación de conocimientos realizada principalmente por uso de herramientas de conocimiento que ya se ha guardado en la KB Cyc.
- Este tercer proceso delega la mayoría del trabajo a las herramientas. Para trabajar con las herramientas Cyc, los usuarios sólo recomendamos al sistema, las fuentes de conocimiento son leídas y explicadas las partes más difíciles del texto.
- Dos actividades se realizan en los tres procesos anteriores:
- **Actividad 1.** Desarrollo de una representación del conocimiento y la ontología de nivel superior conteniendo la mayoría de conceptos abstractos.
- Los términos como atributo o valor son ejemplos de representación del conocimiento, y Thing, Intangibles o Collection, son ejemplos de conceptos abstractos.

- **Actividad 2.** Representación del conocimiento de los diferentes dominios usando primitivas.

2.2.10. Método de Uschold y King

Esta metodología se utilizó para el desarrollo de Enterprise Ontology, desarrollado por la Universidad de Edimburgo junto a IBM, Lloyds y Unilever. Esta metodología consta de cuatro fases:

- ◆ **Identificación del Propósito:** ¿Porqué y para qué se construye?
- ◆ **Construcción de la Ontología:**
 - **Captura de la Ontología:**
 - Identificación de los conceptos clave y relaciones del dominio, centrándose más en conceptos que en nombres específicos.
 - Producción de definiciones no ambiguas para estos conceptos y relaciones.
 - Identificación de los términos para referirse a estos conceptos y relaciones.

- Las recomendaciones son comenzar por un nivel de conceptos intermedio y después generalizar y especializar.
- **Codificación**
Representar el conocimiento adquirido en un lenguaje formal.
- **Integrar Ontologías ya existentes**
Durante los procesos anteriores no podemos plantear que recursos existentes pueden ser aprovechados.
- ◆ **Evaluación:** Juzgar el resultado frente a las especificaciones de las necesidades establecidas y del dominio real.
- ◆ **Documentación:** En paralelo documentar adecuadamente la Ontología desarrollada.

2.2.11. Metodología de Grüninger y Fox

Se utilizó para el desarrollo de la Ontología TOVE (Toronto Virtual Enterprise), Ontología de procesos empresariales. Se divide en cinco fases:

- ❖ **Captura de escenarios.**

- Se parte de escenarios informales que surjan del dominio de aplicación y que no se puedan resolver con ontologías existentes.
- Estos escenarios proveerán soluciones que darán la semántica de los objetos y relaciones, que serán después incluidos en la ontología.
- ❖ Formulación de preguntas de competencias informales.
 - Formulamos cuestiones informales que la ontología debe describir.
 - Las respuestas a esas cuestiones también han de poder ser formuladas con términos de la ontología.
 - Estas cuestiones han de permitir evaluar las necesidades y restricciones que debe cumplir la ontología.
- ❖ Especificar la terminología en un lenguaje formal.
 - Obtener la terminología informal de las cuestiones de competencia.
 - Especificar formalmente esta terminología.
 - Formular cuestiones de competencia formales utilizando la terminología formalizada.

- ❖ Especificar los axiomas y definiciones de los términos en lenguaje formal.
 - Especificar en lógica de primer orden los términos y restricciones.
 - Si se detecta que los términos son insuficientes para axiomatizar, se formulan nuevas cuestiones de competencia y se añaden nuevos términos.
- ❖ Establecer las condiciones que determinan cuando la ontología está completa.

2.2.12. Metodología KACTUS

Desarrollada en el proyecto SPRIT KACTUS, investigaba la posibilidad de la reutilización de conocimiento en procesos técnicos complejos y el beneficio de uso de ontologías. Consta de tres fases:

- ◆ Especificación de la aplicación: Identificar el contexto y los elementos que debemos modelar.
- ◆ Diseño preliminar basado en categorías del nivel de abstracción más general:

- Los términos identificados en la fase previa se utilizan como entrada para tener una visión global del modelo.
- Durante este proceso se puede plantear la reutilización de ontologías.
- ◆ Refinamiento y estructuración de la ontología: Especializamos los términos con la idea de máxima modularidad y homogeneidad en el desarrollo.

2.2.13. Método basado en SENSUS

El método se propone a asociar condiciones específicas al dominio a la ontología enorme y podar, en la ontología enorme, esas condiciones que son irrelevantes para la nueva ontología que deseamos construir. El resultado de este proceso es el esqueleto de la nueva ontología.

2.2.14. Metodología Methontology

Metodología inspirada en la metodología de desarrollo de software. Consta de tres fases:

- ◆ **Actividades de administración del proyecto.**
 - **Planificación:** ¿Qué tareas se han de realizar y cuándo?
 - **Control de calidad:** Asegurar que el resultado tiene la calidad esperada.
- ◆ **Actividades orientadas al desarrollo.**
 - **Especificación:** ¿Porqué y para qué se desarrolla la ontología?, fuente de conocimiento, alcance, etc.
 - **Conceptualización:** Determinar las estructuras que representan el modelo.
 - **Formalización:** Transformar el modelo conceptual en un modelo formal o semicomputable (lógica, frames).
 - **Implementación:** Transformación del modelo a un lenguaje de programación.
 - **Mantenimiento:** Actualizar y corregir la ontología.
- ◆ **Actividades de Soporte.**

- **Adquisición del conocimiento:** Uso de procesos estándar de adquisición del conocimiento.
- **Evaluación:** Evaluar las ontologías resultantes.
- **Integración:** Reutilización de ontologías.
- **Documentación:** Detallar cada una de las fases y productos obtenidos en el proceso de creación de la ontología.

Relatada para las actividades del soporte, la figura también muestra que la adquisición de conocimiento, la integración y la evaluación es mayor durante la conceptualización de ontología, y que decrece durante la formalización y la implementación.

Las razones para este esfuerzo mayor son:

- ◆ La mayor parte del conocimiento es adquirido a principios de la construcción de ontología.
- ◆ La integración de otras ontologías dentro de la que estamos construyendo no está pospuesta para la actividad de implementación. Antes de la integración en

el nivel de implementación, la integración en el nivel de conocimiento debería ser llevada afuera.

- ◆ La conceptualización de la ontología debe ser evaluada correctamente para evitar que los errores se propaguen en las demás etapas del ciclo de vida de la ontología.

Las relaciones entre las actividades ejecutadas durante el desarrollo de la ontología son llamadas intradependencias, o lo que es lo mismo, definen el ciclo de vida de ontología.

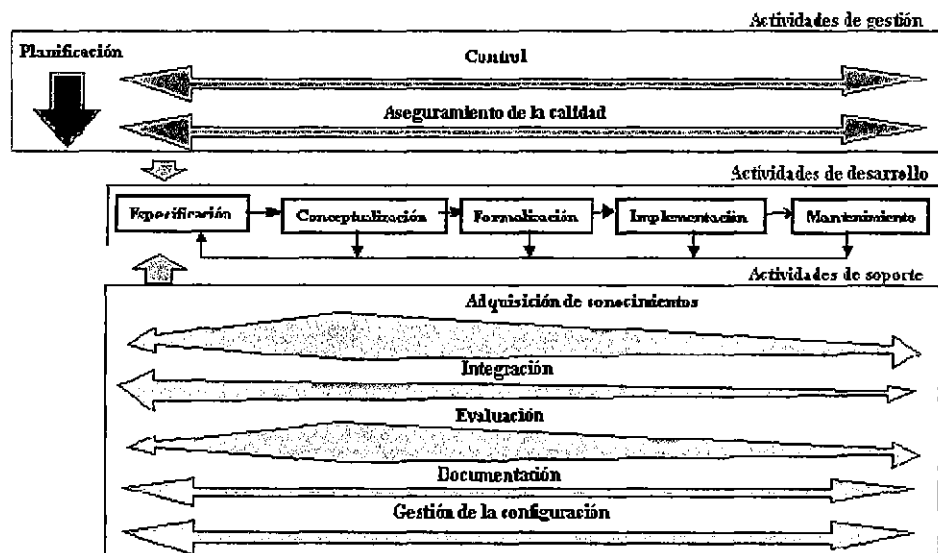


Figura 11: Proceso de desarrollo y ciclo de vida de Methontology

Fuente : Gómez, Fernández & Corcho, (2004)

METHONTOLOGY también considera que las actividades realizadas durante el desarrollo de una ontología pueden involucrar ejecutar otras actividades en otras ontologías ya construidas o bajo construcción. Por consiguiente, METHONTOLOGY no considera sólo intradependencias, sino que también las interdependencias. Éstas están definidas como las relaciones entre actividades ejecutadas cuando construimos ontologías diferentes. En lugar de hablar del ciclo de vida de una ontología, deberíamos hablar de ciclos de vida híbridos de ontologías. La razón es eso, más de las veces y antes de integrar una ontología en una nueva, ésta a ser reusada es modificada o unida con otras del mismo dominio.

La figura 12 presenta las tareas propuestas por METHONTOLOGY para la actividad de conceptualización, haciendo énfasis en los componentes (conceptos, atributos, relaciones, constantes, axiomas formales, reglas e instancias) que se construyen en cada una de las tareas. El orden propuesto en la figura no es estrictamente secuencial.

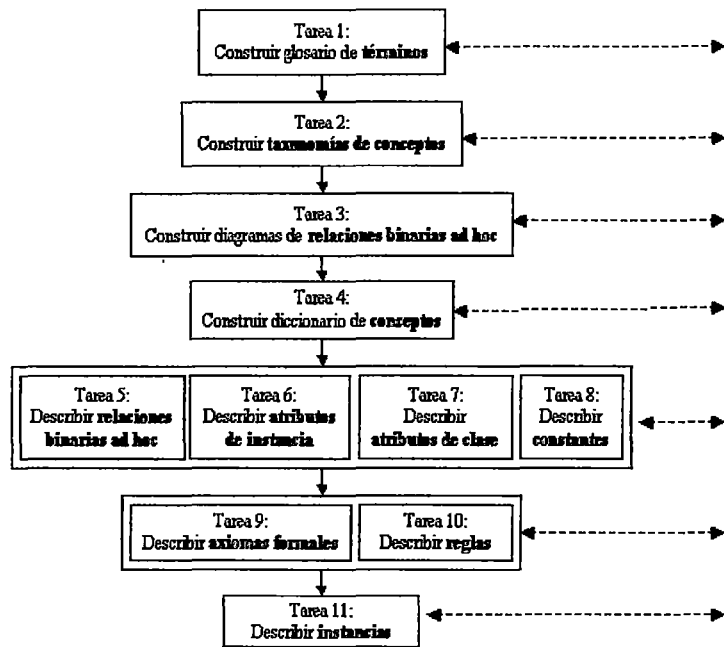


Figura 12: Tareas de las actividades de conceptualización de acuerdo a METHONTOLOGY.

Fuente : Gómez, Fernández & Corcho, (2004).

2.2.15. Metodología On-To-Knowledge

La metodología On-To-Knowledge aplica ontologías a la información disponible electrónicamente para mejorar la calidad de la gestión de conocimiento en organizaciones grandes y distribuidas. Esta incluye la identificación de metas que deberían ser conseguidas por herramientas de gestión de conocimiento y está basada en el análisis de

escenarios de uso y en los diferentes papeles desempeñados por trabajadores de conocimiento y accionistas en las organizaciones (Hernández & Saiz, 2007).

Para el diseño de software orientado a objetos, no hay una sola metodología correcta para diseño y construcción de una ontología. Hay, sin embargo, algunas metodologías que son bien establecidas y parecen funcionar mejor que las otras, y una de ellas es la metodología On-to-Knowledge, que se muestra en la figura 13.

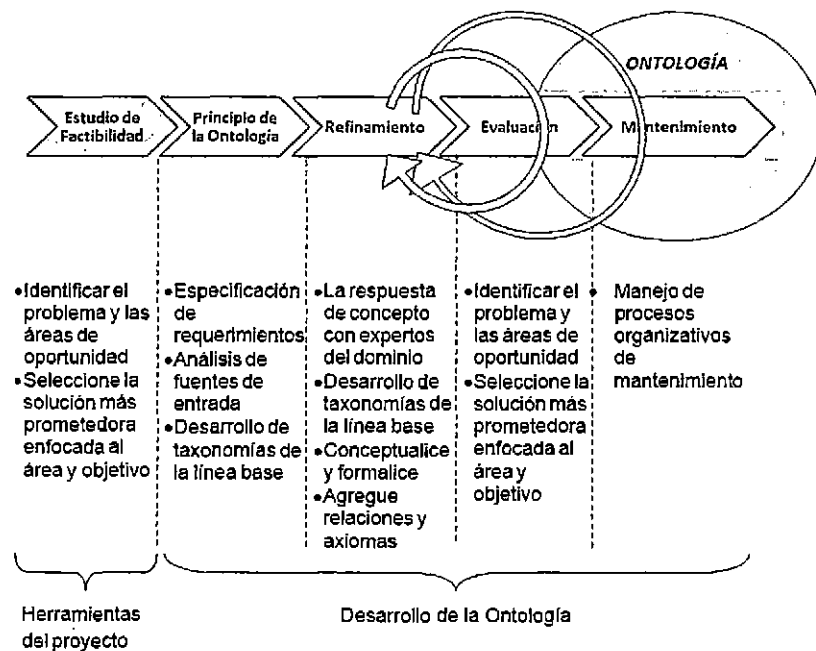


Figura 13: Proceso de On-To-Knowledge.

Fuente : Gómez, Fernández & Corcho (2004).

Proceso 1: Estudio de Factibilidad. Según On-To-Knowledge, el estudio de factibilidad se aplica a la aplicación completa y, Por lo tanto, debe llevarse a cabo antes de elaborar las ontologías. De hecho, el estudio de factibilidad sirve como base para el proceso de inicio del partido.

Proceso 2: Principios de la Ontología. El resultado de este proceso es la ontología de los requisitos de documento de especificaciones que describe lo siguiente: el dominio y el objetivo de la ontología; las directrices de diseño (por ejemplo, las convenciones de nomenclatura), las fuentes de conocimiento disponibles (libros, revistas, entrevistas, etc), los usuarios potenciales y el uso de casos, así como las solicitudes de apoyo de la ontología.

En el proceso de principios de la ontología los desarrolladores deben buscar ontologías potencialmente reutilizables ya desarrolladas.

Proceso 3: Refinamiento. El objetivo aquí es producir una ontología madura y una aplicación orientada al objetivo

según la especificación dada en el proceso de principio de la ontología. Este proceso de refinamiento está dividido en dos actividades:

La actividad 1: El proceso de respuesta de conocimiento con expertos de dominio. La línea base de la ontología, esto es, el primer borrador de ésta obtenido en el proceso 2, están refinados por medio de la interacción con expertos en el dominio. Cuando esta actividad es realizada, los axiomas son identificados y modelados. Durante la respuesta, los conceptos son recogidos a un lado y las condiciones para poner etiquetas a los conceptos por otra parte. Entonces, las condiciones y los conceptos son acotados. La metodología On-To-Knowledge propone el uso de representaciones intermedias para modelar el conocimiento. En este aspecto, sigue las ideas básicas de METHONTOLOGY. Si varios expertos participan en la construcción de la ontología, entonces hay que cumplir un consenso.

La actividad 2: Formalización. La ontología es implementada usando un lenguaje de ontología. Tal lenguaje

es seleccionado según los requisitos específicos de la aplicación. Para llevar a cabo la formalización, On-To-Knowledge recomienda el editor de ontología *OntoEdit*, el cual genera automáticamente el código de ontología en varios lenguajes. Otros editores de ontología que realizan funciones similares pueden ser también usados.

Proceso 4: Evaluación. El proceso de evaluación sirve como una prueba de la utilidad de las ontologías desarrolladas y su ambiente informático asociado. El producto obtenido es llamado aplicación basada en ontología. Durante este proceso dos actividades son efectuadas:

La actividad 1: Comprobando los requisitos y las preguntas de competencia. Los desarrolladores inspeccionan si la ontología satisface los requisitos y " pueden contestar a " las preguntas de competencia.

La actividad 2: Probando la ontología en el ambiente de aplicación objetivo. El último refinamiento de la ontología puede originarse en esta actividad.

Este proceso de evaluación está estrechamente vinculado para el proceso de refinamiento. De hecho, varios ciclos son necesarios hasta que la ontología de objetivo alcance el nivel planeado.

Proceso 5: Mantenimiento. Es importante para aclarar quién es responsable del mantenimiento y cómo esto debería ser efectuado. On-To-Knowledge propone llevar a cabo el mantenimiento de la ontología como parte del software de sistema.

On-To-Knowledge ha sido adoptado para construir empresas virtuales, para organizar memorias corporativas, y ayudar a la funcionalidad del escritorio dentro de centros de llamada. Esta última aplicación provee a clientes de información apropiada en productos y servicios y decide acerca de la habilidad de una compañía a establecer relaciones exitosas y estables con sus clientes.

2.2.16. Comparando métodos y metodologías de desarrollo de ontologías

Para comparar la estrategia de la construcción de métodos y las metodologías proponemos el siguiente conjunto de criterios, resumidos en cuadro 1:

- ◆ **La propuesta de ciclo de vida.** El ciclo de vida identifica el conjunto de etapas a través de las cuales la ontología se mueve durante su tiempo de vida. También describe cuáles actividades tienen que ser realizadas en cada etapa y cómo las etapas son descritas, algo semejante como las relaciones de precedencia, la simultaneidad, etc. Los ejemplos de ciclos de vida son:
 - **El ciclo de vida incremental.** Según este acercamiento, la ontología aumentaría por estratos, permitiendo la inclusión de definiciones nuevas sólo cuando una nueva versión es planeada. Este modelo impide la inclusión de definiciones nuevas si no fueran planeados, por eso permite un desarrollo incremental.

- **Desarrollando prototipos.** Según este acercamiento, la ontología aumenta según las necesidades. Este modelo permite modificar, agregando, y remover definiciones en la ontología en cualquier momento.
- ◆ **La estrategia según la aplicación.** Este criterio está relacionado al grado de dependencia de la ontología con la aplicación usándolo. Considerando este criterio, las metodologías y los métodos pueden estar clasificados en los siguientes tipos:
 - **El dependiente de aplicación.** Las ontologías se construyen en la base de las aplicaciones que los usan.
 - **El semidependiente de aplicación.** Los escenarios posibles de uso de ontología son identificados en la etapa de especificación.
 - **Independiente en la aplicación.** El proceso es completamente independiente de los usos de la ontología en las aplicaciones.
- ◆ **El uso de ontologías núcleo.** En este criterio analizamos si es posible o no usar una ontología núcleo como un

punto de partida en el desarrollo de la ontología de dominio.

- ◆ La estrategia para identificar conceptos. Hay tres estrategias posibles para identificar conceptos: de lo más concreto a lo más abstracto (bottom-up), de lo más abstracto a lo más concreto (top-down), o de lo más relevante a lo más abstracto y más concreto (middle-out).

Acerca del soporte tecnológico, es importante para saber cuáles herramientas dan soporte total o parcial para las metodologías y los métodos. La tabla 2 enumera estas herramientas y las suites de herramienta.

La tabla 3 resume los procesos de desarrollo de ontología en todos los métodos y las metodologías. Cada celda de la tabla puede ser llenada con tres tipos de valores. El valor "descrito" significa que el método o la metodología describe cómo realizar cada tarea en la actividad pensada, cuándo hacerla, quién tiene que hacerla, etc. El valor "propuesto" significa que la metodología solamente reconoce el proceso. El valor "NP" quiere decir que la documentación pública no menciona la actividad poco considerada (Gómez, Fernández & Corcho, 2004).

Tabla 1: El resumen de las estrategias de construcción.

Características	Cyc	Uschold & King	Gruninger & Fox	KACTUS	METHONTOLOGY	SENSUS	On-To-Knowledge
Propuesta del ciclo de vida	Desarrollo de Prototipos	No propuesto	Desarrollo de prototipos o incremental	Desarrollo de prototipos	Desarrollo de prototipos	No propuesto	Incremental y cíclico con desarrollo de prototipos
Estrategia con respecto a la aplicación	Aplicaciones independientes	Aplicaciones independientes	Aplicaciones semi-dependientes	Aplicaciones dependientes	Aplicaciones independientes	Aplicaciones semi-dependientes	Aplicaciones dependientes
Estrategia para identificar conceptos	No específica	Middle-out	Middle-out	Top-down	Middle-out	No específica	Top-down Bottom-up Middle-out
Uso de una ontología núcleo	Si	No	No	No	Depende de la disponibilidad de recursos	Si	Depende de la disponibilidad de recursos

Fuente : (Gómez, Fernández & Corcho, 2004)

Tabla 2: Soporte tecnológico.

Características	Cyc	Uschold & King	Gruninger & Fox	KACTUS	METHONTOLOGY	SENSUS	On-To-Knowledge
Herramientas que dan soporte	Cyc todos	No especificado	No especificado	No especificado	ODE WebODE OntoEdit Protégé	Usualmente OntoSaurus	OntoEdit con sus plugins

Fuente : (Gómez, Fernández & Corcho, 2004)

Tabla 3: Resumen del proceso de desarrollo de ontologías.

	Características	Cyc	Uschold & King	Grüniger & Fox	KACTUS	METHONTOLOGY	SENSUS	On:To:Knowledge	
Actividades para la gestión de ontologías	Planificación	NP	NP	NP	NP	Propuesto	NP	Descrito	
	Control	NP	NP	NP	NP	Propuesto	NP	Descrito	
	Aseguramiento de la Calidad	NP	NP	NP	NP	Propuesto	NP	Descrito	
Actividades orientadas al desarrollo de ontologías	Proceso de pre-desarrollo	Ambito de estudio	NP	NP	NP	NP	NP	Propuesto	
		Estudio de factibilidad	NP	NP	NP	NP	NP	NP	Descrito
	Proceso de desarrollo	Especificación	NP	Propuesto	Descrito en detalle	Propuesto	Descrito en detalle	Propuesto	Descrito en detalle
		Conceptualización	NP	NP	Descrito en detalle	Propuesto	Descrito en detalle	NP	Propuesto
		Formalización	NP	NP	Descrito en detalle	Descrito	Descrito	NP	Descrito
		Implementación	Propuesto	Propuesto	Descrito	Propuesto	Descrito en detalle	Descrito	Descrito
	Proceso de Post-desarrollo	Mantenimiento	NP	NP	NP	NP	Propuesto	NP	Propuesto
Uso		NP	NP	NP	NP	NP	NP	Propuesto	
Actividades para el soporte de ontologías	Adquisición de conocimiento	Propuesto	Propuesto	Propuesto	NP	Descrito en detalle	NP	Descrito	
	Evaluación	NP	Propuesto	Descrito en detalle	NP	Descrito en detalle	NP	Propuesto	
	Integración	Propuesto	Propuesto	Propuesto	Propuesto	Propuesto	NP	Propuesto	
	Gestión de configuración	NP	NP	NP	NP	Descrito	NP	Propuesto	
	Documentación	Propuesto	Propuesto	Propuesto	NP	Descrito en detalle	NP	Descrito	
	Integración y Alineamiento	NP	NP	NP	NP	NP	NP	NP	

Fuente : (Gómez, Fernández & Corcho, 2004)

2.2.17. Lenguajes para el desarrollo de ontologías

El auge de la internet condujo a la creación de lenguajes de ontología para explotar las características de la Web. Tales son usualmente llamados lenguajes de ontologías basados en la Web o de marcas de ontología. Su sintaxis está basada en lenguajes de marcas existentes como el HTML y XML, cuyo propósito no es desarrollar ontologías sino la presentación y cambio de datos respectivamente. Las relaciones entre éstos son demostradas en la figura 14.

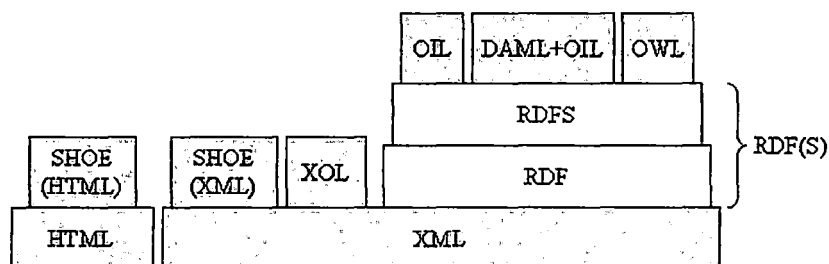


Figura 14: Lenguaje de marcas de Ontologías.

Fuente : Gómez, Fernández & Corcho, (2004).

El primer lenguaje de marcado de ontología que apareció fue SHOE. **SHOE** es un lenguaje que combina marcos y reglas. Se construyó como una extensión de

HTML, en 1996. Usó etiquetas diferentes de las que usa la especificación de HTML, así permitió la inserción de ontologías en documentos de HTML. Más tarde su sintaxis fue adaptada para XML.

Los demás lenguajes de marcado de ontología presentados aquí se basan en XML. *XOL* fue desarrollado como una XMLización de un subconjunto pequeño de primitivas del protocolo OKBC, llamado OKBC-LITE. *RDF* fue desarrollado por la W3C (el Consorcio de la World Wide Web) como un lenguaje basado en la red semántica para describir recursos de la Web. Su desarrollo empezó 1997, y RDF fue propuesto como una recomendación de W3C en 1999. El lenguaje RDF Schema fue también construido por el W3C como una extensión a RDF con primitivas basados en marcos. Este lenguaje fue propuesto como una Recomendación Candidata W3C en 2000 y luego sufrió una revisión principal en noviembre del 2002. Más tarde, fue revisado en enero del 2003. La combinación de ambos RDF y RDF Schema es normalmente conocida como RDF(S).

Estos lenguajes han establecido los fundamentos de la Web Semántica. En este contexto tres lenguajes más han sido desarrollados como extensiones para RDF (S): **OIL**, **DAML+OIL**, y **OWL**. **OIL** fue desarrollado a principios del año 2000 en el marco de trabajo del proyecto europeo IST On-To-Knowledge. Añade a las KR primitivas marco-base para RDF(S) y su semántica formal se basa en lógicas de descripción. **DAML+OIL** fue creado más tarde (entre los años 2000 y 2001) por un comité unido de los USA y el EU en el contexto del proyecto DARPA DAML. Se basó en la especificación previa DAML-ONT, lo cual fue creado al final de 2000, y en OIL. DAML+OIL añade primitivas KR basado en DL para RDF(S). En 2001 el W3C formó un grupo de trabajo llamado Web Ontology (WebOnt) Working Group.

La meta de este grupo fue hacer un lenguaje nuevo de marcado de ontología para la Web Semántica. El resultado de su trabajo es el lenguaje OWL. OWL cubre más de las características de DAML+OIL y ha renombrado

más de las primitivas tan aparecidos en ese lenguaje (McGuinness & Harmelen, 2004).

Estos lenguajes, ordenados de menor a mayor capacidad expresiva, son los siguientes:

XML: XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones, sirve para estructurar, almacenar e intercambiar información.

En resumen podemos decir que:

- ◆ XML es para estructurar datos
- ◆ XML se parece un poco al HTML
- ◆ XML es texto, pero no está pensado para ser leído
- ◆ XML es una familia de tecnologías
- ◆ XML lleva HTML a XHTML
- ◆ XML es modular

- ◆ XML es la base de RDF y de la Web Semántica
- ◆ XML es independiente de la plataforma y bien soportado

Ejemplo de documento XML:

```
<?xml version="1.0" ?>
<alumnos>
  <nombre>Richal</nombre>
  <email>richal@ejemplo.com</email>
</alumnos>
```

Cada documento XML posee una estructura lógica y una física. La estructura lógica del documento es una serie de declaraciones, elementos, comentarios, etc. que se indican en el documento mediante marcas explícitas. La estructura física del documento es una serie de unidades llamadas entidades, es decir, indica los datos que contendrá el documento. Las estructuras lógica y física deben anidarse de forma correcta.

XML Schema: es un lenguaje que permite definir la estructura de la información en documentos XML para que sea conforme a un esquema determinado. También extiende XML con la posibilidad de especificar tipos de datos. Sin

embargo, la semántica acerca del contenido de la información que se puede expresar con este lenguaje es limitada, y queda restringida al ámbito del formato y tipo de los datos en un documento. Por este motivo, no se suele considerar como un lenguaje de ontologías en sentido propio.

SHOE: Simple HTML Ontology Extensions. Fue el primer lenguaje de etiquetado para diseñar ontologías en la Web. Este lenguaje nació antes de que se ideara la Web Semántica. Las ontologías y las etiquetas se incrustaban en archivos HTML. Este lenguaje permite definir clases y reglas de inferencia, pero no negaciones o disyunciones.

A su vez se desarrollaron muchos editores, buscadores, APIS, etc; pero este proyecto fue abandonado a medida que se desarrollaron OIL y DAM; aunque también existe una serialización de este lenguaje en XML.

OIL: Ontology Inference Layer. Este lenguaje, derivado en parte de SHOE, fue impulsado también por el proyecto de la

Unión Europea On-To-Knowledge. Utiliza ya la sintaxis del lenguaje XML y está definido como una extensión de *RDFS*. Se basa tanto en la lógica descriptiva (declaración de axiomas) y en los sistemas basados en frames (taxonomías de clases y atributos). OIL posee varias capas de sublenguajes, entre ellas destaca la capa base que es *RDFS*, a la que cada una de las capas subsiguientes añade alguna funcionalidad y mayor complejidad. La principal carencia de este lenguaje es la falta de expresividad para declarar axiomas.

DAML y OIL: Este lenguaje nació fruto de la cooperación entre OIL y DARPA y unifica los lenguajes DAML (DARPA's Agent Markup Language) y OIL (Ontology Inference Layer). Se basa ya en estándares del W3C. El lenguaje DAML se desarrolló como una extensión del lenguaje XML y de Resource Description Framework (RDF) y para extender el nivel de expresividad de *RDFS*. DAML y OIL hereda muchas de las características de OIL, pero se aleja del modelo basado en clases (frames) y potencia la lógica descriptiva. Es más potente que *RDFS* para expresar ontologías. En la

última revisión del lenguaje (DAML+OIL), ofrece ya un rico conjunto de elementos con los cuales se pueden crear ontologías y marcar la información para que sea legible y comprensible por máquina. También funciona como formato de intercambio. Sin embargo, este lenguaje presenta algunas carencias debido a su complejidad conceptual y de uso, complejidad que se intentó solventar con el desarrollo de OWL. No obstante, se desarrollaron muchas aplicaciones que utilizan DAML-OIL y también existen herramientas para convertir DAML a OWL.

RDF(S): RDF and RDF Schema: The Resource Description Framework (RDF), ha sido desarrollado por el W3C para crear metadatos para describir recursos de Web. El modelo de datos RDF es equivalente al formalismo semántico de redes y consta de tres tipos de objeto: los recursos, las propiedades y las sentencias. Es un lenguaje para referenciar la información de los recursos de la World Wide Web (Gómez, Fernández & Corcho, 2004).

RDF(S) es el término comúnmente utilizado para referirse a la combinación de RDF y RDFS.

Antes de analizar qué componentes de ontología pueden ser expresados en RDF(S), veamos cómo definir una ontología RDF(S). Una ontología en RDF(S) debe comenzar con el nodo raíz RDF. En este nodo raíz los namespaces para las ontologías KR RDF y RDFS deben ser incluidos. Estos namespaces son usualmente identificados con los prefijos `rdf` y `rdfs`, y apuntan hacia el estándar URLs de ambos RDF y RDFS respectivamente. Esta definición es mostrada debajo:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
```

El uso de estos namespaces nos permite usar los prefijos `rdf` y `rdfs` para estas primitivas KR que pertenecen a RDF y RDFS respectivamente.

Las especificaciones RDF proporcionan un sistema de ontología ligera para apoyar el intercambio de conocimientos en la Web.

RDF es un modelo de datos para describir objetos (que se denominan recursos) y las relaciones entre ellos, con una semántica simple. Más concretamente, RDF fue concebido como un marco de trabajo para representar metadatos acerca de recursos Web, tales como, el título, autor y fecha de modificación de una página Web, licencias de uso sobre documentos Web, o el horario en el que pudiera estar disponible un recurso compartido. Los modelos basados en RDF pueden representarse mediante un lenguaje con una sintaxis XML denominado RDF/XML (Manola & Miller, 2004).

El modelo de datos básico consiste en tres tipos de objetos:

- ◆ **Recursos:** Todas las cosas descritas por expresiones RDF se denominan recursos. Un recurso puede ser una

página Web completa; tal como el documento parte de una página Web; Los recursos se designan siempre por URIs.

- ◆ **Propiedades:** Una propiedad es un aspecto específico, característica, atributo, o relación utilizado para describir un recurso. Cada propiedad tiene un significado específico, define sus valores permitidos, los tipos de recursos que puede describir, y sus relaciones con otras propiedades.
- ◆ **Sentencias:** declaraciones, enunciados. Un recurso específico junto con una propiedad denominada, más el valor de dicha propiedad para ese recurso es una sentencia RDF. Estas tres partes individuales de una sentencia se denominan, respectivamente, sujeto, predicado y objeto. El objeto de una sentencia puede ser otro recurso o puede ser un literal; es decir, un recurso o una cadena simple de caracteres.

Ejemplos:

Los recursos se identifican por un identificador de ellos. Un identificador de recursos es un URI más un identificador opcional de ancla.

Considerar como ejemplo simple la sentencia:

Ora Lassila es el creador [autor] del recurso
<http://www.w3.org/Home/Lassila>.

Esta sentencia comprende las siguientes partes:

Cuadro 1: Modelo RDF

Sujeto (Recurso)	http://www.w3.org/Home/Lassila
Predicado (Propiedad)	Creator
Objeto (literal)	"Ora Lassila"

Fuente: Manola & Miller (2004)

El documento XML completo que contiene la descripción citada anteriormente, podría ser:

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:s="http://description.org/schema/">
  <rdf:Description about="http://www.w3.org/Home/Lassila">
    <s:Creator>Ora Lassila</s:Creator>
  </rdf:Description>
</rdf:RDF>
```

RDF Primer ofrece los conocimientos básicos requeridos para usar RDF, introduce los conceptos básicos de RDF y describe su sintaxis XML. También describe cómo definir los vocabularios RDF usando el Lenguaje de Descripción del Vocabulario RDF y ofrece una introducción a algunas de las aplicaciones para usar RDF, además de explicar el contenido y propósito de otros documentos y especificaciones RDF (Lozano, 2007).

Hay dos tipos de sintaxis en RDF: la serializada, que expresa las capacidades totales del modelo de datos de una forma muy regular y la abreviada, que contiene términos adicionales que hacen que se pueda presentar de forma más compacta el modelo de datos. Ambas formas pueden mezclarse. A continuación podemos ver un ejemplo de las dos representaciones en la figura 15.

```

1 <!-- ***** Serializada ***** -->
2
3 <rdf:RDF>
4   <rdf:Description about="http://www.etsii.us.es/user/id/333/documentacion">
5     <d:Creator rdf:resource="http://www.etsii.us.es/user/id/333"
6       p:Name="Elena Lozano"
7       p:Email="elozano@us.es" />
8   </rdf:Description>
9 </rdf:RDF>
10
11 <!-- ***** Abreviada: ***** -->
12
13 <rdf:RDF
14   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns"
15   xmlns:d="http://description.org/" >
16   <rdf:Description about="http://www.etsii.us.es/user/333/docs">
17     <d:Creator>
18       <rdf:Description about="http://www.etsii.us.es/user/333">
19         <rdf:type resource="http://www.etsii.us.es/schema/Person"/>
20         <p:Name>Elena Lozano</p:Name>
21         <p:Email>elozano@us.es</p:Email>
22       </rdf:Description>
23     </d:Creator>
24   </rdf:Description>
25 </rdf:RDF>

```

Figura 15 : Ejemplo de los dos tipos de sintaxis de RDF.

Fuente : Lozano (2007)

OWL: Web Ontology Language o Lenguaje de Ontologías para la Web es un lenguaje de etiquetado semántico para publicar y compartir ontologías en la Web. Se trata de una recomendación del W3C, y puede usarse para representar ontologías de forma explícita, es decir, permite definir el significado de términos en vocabularios y las relaciones entre aquellos términos (ontologías). En realidad, OWL es una extensión del lenguaje RDF y emplea las tripletas de RDF, aunque es un lenguaje con más poder expresivo que éste. Se trata de un lenguaje diseñado para usarse cuando

la información contenida en los documentos necesita ser procesada por programas o aplicaciones, en oposición a situaciones donde el contenido solamente necesita ser presentado a los seres humanos. OWL surge como una revisión al lenguaje DAML-OIL y es mucho más potente que éste. Al igual que OIL, OWL se estructura en capas que difieren en la complejidad y puede ser adaptado a las necesidades de cada usuario, al nivel de expresividad que se precise y a los distintos tipos de aplicaciones existentes (motores de búsqueda, agentes, etc.). Existen 3 sublenguajes que van creciendo con respecto al nivel de expresión: OWL Lite, OWL DL y OWL Full.

En cambio, aunque OWL puede posicionarse como el lenguaje estándar para la definición de ontologías, todavía se encuentra en versión borrador W3C, y el resto de propuestas, aunque demostrada su viabilidad técnica, como ya hemos comentado anteriormente, su complejidad conceptual las alejan de ser ampliamente aceptadas por la comunidad de desarrolladores web (Peis & Hassan, 2003).

Ahora haremos comentarios sobre los resultados principales presentado en tabla 4, tratando de resaltar las principales diferencias entre los lenguajes descritos en esta sección.

Tabla 4: Resumen de las características de lenguajes para la construcción de ontologías

	Concilingas	LOOM	OKBC	OCML	FLegit	SHOE	XDL	RDFS	OIL	DAML-OIL	OWL
CONCEPTS											
Attributes											
Instance attributes	+	-	-	+	+	-	+	+	-	-	+
Class attributes	+	-	+	+	+	-	+	W	-	W	W
Facets											
Type constraints	+	-	-	+	+	+	+	+	-	-	+
Cardinality constraints	+	+	+	+	W	-	+	-	+	+	+
Procedural knowledge	-	+	-	+	-	-	-	-	-	-	-
CONCEPT TAXONOMIES											
Subclass-Of	+	+	-	+	+	+	+	+	-	-	+
Disjoint-Decomposition	+	+	-	+	W	-	-	-	+	+	+
Exhaustive-Decomposition	+	-	-	W	W	-	-	-	W	W	W
Partition	+	-	-	+	W	-	-	-	+	+	W
RELATIONS											
Binary relations	+	+	-	+	+	+	+	+	+	+	+
n-ary relations	-	+	W	+	W	-	W	W	W	W	W
Relation hierarchies	+	+	-	+	W	-	-	+	+	+	+
Integrity constraints	+	+	-	+	+	-	-	-	-	-	-
FUNCTIONS											
Binary functions	+	+	W	+	+	-	W	-	+	+	+
n-ary functions	+	+	-	+	+	-	-	-	-	-	-
OTHER COMPONENTS											
Formal Axioms	+	+	-	+	+	-	-	-	-	-	-
Instances	+	+	+	+	+	+	+	+	+	+	+
Procedures	+	+	-	+	-	-	-	-	-	-	-
Rules	-	-	-	+	-	+	-	-	-	-	-

Fuente: Gómez, Fernández & Corcho (2004)

La conclusión principal obtenida del análisis de la tabla está relacionada con el poder expresivo diferente de cada lenguaje. Mientras algunos permiten representar ontologías pesadas (con axiomas formales, funciones, reglas, procedimientos, y otros componentes ricos de modelado), los otros sólo permiten representar ontologías ligeras (básicamente, conceptos, taxonomías de conceptos y las relaciones entre ellos). Hay diferencias relevantes en la expresividad entre lenguajes tradicionales y los de marcado de ontología. De hecho, la mayor parte de estos componentes pueden ser representados en lenguajes tradicionales (como Ontolingua, LOOM and y OCML), mientras los de marcado de ontologías normalmente no proveen pero tienen la intención de hacerlo. Por ejemplo, en caso de los axiomas formales, los de marcado de ontología confían en un futuro “nivel lógico” que debe ser construido en la parte superior de ellos, y eso proveerá características lógicas adicionales no previstas por los lenguajes mismos.

Ahora comentaremos brevemente sobre algunas de las características principales presentadas en la tabla.

Como hemos dicho arriba, todos los lenguajes de ontología permiten representar conceptos y sus atributos (slots, roles, y propiedades). Los atributos de instancia definen propiedades de concepto cuyo valor(es) están asignados por cada instancia de concepto. Ellos pueden estar definidos en todos los lenguajes. Los atributos de clase especifican propiedades del concepto mismo. No hay primitivas específicas o palabras claves para representarlos en SHOE, RDF (S), y en los lenguajes basados en DL (LOOM, OIL, DAML+OIL, y OWL). Esto está supuesto al hecho que el razonamiento con estos atributos en las lógicas de descripción es difícil. Sin embargo, pueden ser representados en RDF(S), DAML+OIL, y OWL con un workaround, como especifica la tabla. Este workaround consiste en definirles como propiedades cuyo dominio es la clase *rdfs:Class*, *daml:Class*, y *owl:Class* respectivamente, y especificando su valor(es) en la definición de concepto donde el atributo de clase toma un valor.

A continuación nos vamos a detener un poco más en detallar las características del lenguaje OWL, así como los fundamentos en los que se sustenta y que justificarán su elección como lenguaje de implementación de ontologías en el desarrollo de esta monografía.

2.2.18. El lenguaje OWL

El OWL (Web Ontology Language) o Lenguaje de Ontologías para la Web, se ha convertido en recomendación del W3C el 10 de febrero de 2004. La web está diseñada para usarse cuando la información contenida en los documentos necesita ser procesada por programas o aplicaciones, en oposición a situaciones donde el contenido solamente necesita ser presentado a los seres humanos. OWL puede usarse para representar explícitamente el significado de términos en vocabularios y las relaciones entre aquellos términos. Esta representación de los términos y sus relaciones se denomina una ontología.

OWL sobresale por encima del resto (XML Schema, RDF y RDF Schema) al proveer un vocabulario más rico que ninguno y estándar, con una semántica formal, para expresar conceptos y relaciones entre ellos, y también características de dichas relaciones (simetría, transitividad, unicidad, etc.). En realidad, OWL es una extensión del lenguaje RDF y emplea las tripletas de RDF, aunque es un lenguaje con más poder expresivo que éste (Márquez, 2007).

Esto trae consigo que, a diferencia de como ocurre con otros lenguajes basados en XML/Web, se pueda razonar sobre el conocimiento presente en un SI a partir de su representación. En efecto, muchos de los estándares industriales diseñados para la especificación de transacciones en la Web consisten en un conjunto de protocolos que definen el formato de los datos y el orden en que deben aparecer. Estos protocolos están dotados de una semántica operacional que define transacciones del tipo “un *Trabajador* percibe en la *CuentaDestino* una *Pensión* procedente de la *CuentaOrigen*, que es un *Porcentaje* de

IngresosDeducidos en sus *AñosDeVidaLaboral*". Sin embargo, esta especificación no está diseñada para llevar a cabo tareas de razonamiento aparte del formato de los datos que figuran en ella. Por ejemplo, si un *Trabajador* ha nacido en un país extranjero (por tanto, se trata de un inmigrante), con el que existe convenios en materia de prestaciones por desempleo y jubilaciones, se podría deducir que habría que sumar a sus *AñosDeVidaLaboral*, los trabajos en su país de origen.

Por otro lado, para OWL existen diversas herramientas que permiten razonar sobre especificaciones en este lenguaje. Estas herramientas proporcionan un soporte de razonamiento *genérico*, en el sentido de que no es específico a un dominio particular, como podría ocurrir, por ejemplo, en el caso otros estándares basados en XML sobre los que se discutía en el párrafo anterior. Mientras que la construcción de un razonador requiere un esfuerzo considerable, la creación de una ontología es un proceso mucho más abordable.

De esta forma, los diseñadores de ontologías en OWL pueden beneficiarse de las propiedades formales de este lenguaje (Smith, Welty & McGuinness, 2004).

Por último, si se desean obtener razonamientos útiles de forma automática sobre el conocimiento representado en una ontología, se hace necesario contar con un lenguaje que posea la capacidad de expresar una semántica más rica que la proporcionada por RDF-Schema.

Todo ello ha contribuido a que OWL sea considerado actualmente como un estándar de facto para la descripción de ontologías.

2.2.19. Sublenguajes de OWL

La especificación actual del lenguaje OWL definida por el W3C distingue entre los tres sublenguajes siguientes, ordenados de forma creciente según su potencia expresiva:

OWL Lite: Es útil para dar soporte a usuarios que principalmente necesiten poder especificar jerarquías de clasificación y restricciones simples. Por ejemplo, existe la posibilidad de definir restricciones de cardinalidad sobre las relaciones, pero restringidas a tomar como valores 0 ó 1, únicamente. OWL Lite puede ser un lenguaje útil para representar tesauros o taxonomías reducidas.

OWL DL: Proporciona la máxima capacidad de expresión manteniendo la completitud computacional (se garantiza que todas las conclusiones sean computables) y decidibilidad (todos los cálculos terminarán en un tiempo finito) de los procesos de deducción automática que pueda llevar a cabo un razonador diseñado para este lenguaje.

OWL DL contiene todos los constructores propuestos para el lenguaje OWL, pero su uso está sujeto a ciertas restricciones para asegurar la propiedad de decidibilidad que acabamos de mencionar (en otras palabras, que se pueda garantizar que, según el estado de conocimiento actual, pueda existir un proceso de razonamiento decidible para un

razonador basado en OWL DL). Así, entre otras limitaciones, se establece que los identificadores de clases, propiedades (también llamadas slots, roles o relaciones) e instancias han de ser disjuntos entre sí (esto implica, por ejemplo, que una *clase* no puede ser definida simultáneamente como instancia de otra clase) (Dean & Schreiber, 2004).

No obstante, algunas de estas limitaciones han comenzado a desaparecer parcialmente en las revisiones del lenguaje que se están llevando a cabo actualmente.

OWL DL debe su nombre a su semántica formal, basada en Lógica Descriptiva (*Description Logic*, DL).

OWL Full: Máximo nivel de expresión y la libertad sintáctica de RDF. No hay garantía computacional con vistas a procesos de razonamiento automático. Por ejemplo, en OWL Full una clase puede ser tratada simultáneamente como una colección de objetos (individuos) y como un objeto en sí. Asimismo, OWL Full permite aumentar el significado del vocabulario predefinido (OWL o RDF) para este lenguaje.

Como contrapartida aparece el hecho de que es improbable que llegue a existir algún software de razonamiento que soporte todas las características de OWL Full.

2.2.20. Lenguajes de consultas o de recuperación de una ontología

Los Lenguajes de Recuperación (también denominados query languages) son muy importantes para las ontologías debido a que son una herramienta que facilita la recuperación y organización de la información representada por diferentes vocabularios y esquemas.

Especifican las normas que rigen la formulación de las consultas o queries, aportando precedencia, posibilidades de combinación y orden para que las búsquedas sean más eficientes. Los lenguajes de recuperación pueden dividirse en dos grandes grupos, distinguiendo si su uso se enfoca a las bases de datos relacionales o a la recuperación de información. Una consulta en estos lenguajes se compone de términos y operadores. Los términos serán las palabras

que informan al sistema sobre lo que debe ejecutarse. Los operadores son los encargados de expresar las relaciones que mantienen entre si los términos que definen.

Algunos lenguajes de recuperación son los siguientes:

RQL: RDF Query Language. Es un lenguaje tipado, que siguiendo un enfoque funcional, define un conjunto de consultas básicas e iteradores. A continuación se presentan los elementos más importantes del lenguaje, así como algunos ejemplos de consultas.

Define algunas funciones que le sirven para obtener resultados básicos de la ontología, al mismo tiempo que auxilian en la construcción de consultas más complejas.

RDQL: RDF Data Query Language. Es un lenguaje de consultas para RDF. Tiene un enfoque totalmente declarativo. Considera un modelo RDF como un conjunto de tripletas (sujeto, predicado y objeto). Permite especificar

patrones que son matcheados contra las tripletas del modelo para retornar un resultado.

SPARQL: Structured Protocol and RDF Query Language. Este lenguaje de recuperación basado en RDF es una recomendación para crear un lenguaje de consulta dentro de la Web Semántica. Su utilidad está en que puede representar y utilizar los resultados obtenidos en las búsquedas a través de una gran variedad de información. SPARQL tiene unos elementos denominados triples, al igual que los de RDF, que están formados por un sujeto, un predicado y un objeto. Estos triples son los que describen un recurso y se utilizan en las consultas de este lenguaje.

A continuación vamos a detallar las características del lenguaje de recuperación **SPARQL**, así como los fundamentos en los que se sustenta y que justificarán su elección como lenguaje de recuperación de ontologías en el desarrollo de esta tesis.

2.2.21. SPARQL: necesidad de un lenguaje de consulta

- ◆ Los datos en RDF no servirían de nada si no se pueden utilizar.
- ◆ Los lenguajes de la Web Semántica necesitan interactuar con los datos almacenados en la “base de datos” RDF.
- ◆ Necesidad parecida al lenguaje SQL de bases de datos relacionales.

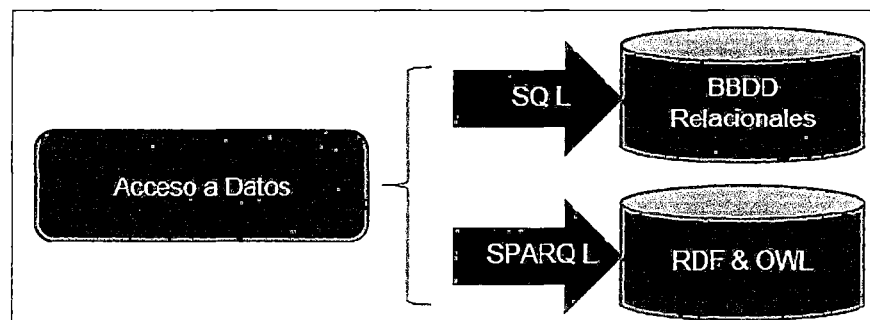


Figura 16: Acceso a Datos

Fuente : Qaissi, (2009)

2.2.22. ¿Qué es SPARQL?

Simple Protocol and RDF Query Language, Lenguaje de consulta del ámbito de la Web Semántica de W3C. Define la sintaxis y la semántica necesarias para una

expresión de consulta sobre un grafo RDF y las diferentes formas de resultados obtenidos. Las consultas SPARQL cubren tres objetivos:

- ◆ Extraer información en forma de URIs y literales.
- ◆ Extraer sub-estructuras RDF.
- ◆ Construir nuevas estructuras RDF partiendo de resultados de consultas.

```
SELECT name
FROM users
WHERE contact='011-201-0000'
```

```
SELECT ?name
FROM <http://semantic/users.rdf>
WHERE {
  ?user rdf:type :User.
  ?user :name ?name.
  ?user :contact='011-201-0000'.
}
```


Instruments	15	16	17	18	19	Apr 19	Apr 18	2000
Federal funds (effective)	1.84	1.68	1.73	1.71	1.67	1.78	1.71	1.73
Nonfinancial commercial paper								
1-month	1.76	1.74	1.74	1.74	1.72	1.74	1.77	1.78
2-month	1.73	1.73	1.80	1.75	1.75	1.77	1.78	1.82
3-month	1.82	1.82	1.83	1.79	1.81	1.81	1.82	1.86
Financial commercial paper								
1-month	1.76	1.77	1.76	1.75	1.74	1.76	1.77	1.80
2-month	1.80	1.80	1.79	1.78	1.77	1.79	1.79	1.82
3-month	1.83	1.84	1.83	1.83	1.81	1.83	1.84	1.87

Figura 17: Ejemplo de comparación entre SQL y SPARQL

Fuente : Qaissi, (2009)

2.2.23. SPARQL: sintaxis y cláusulas

a) Cláusulas Prologue I.

- Esta parte es opcional en la consulta y es Repetible.
- Tiene dos posibles componentes: BASE y PREFIX.

BASE<iri>

Permite cambiar la dirección URI de la base de referencias relativas dentro de la consulta. Por ejemplo, los fragmentos siguientes pueden hacer referencia a la misma IRI (Internationalized Resource Identifier).

<code><http://example.org/book/book1></code>
<code>BASE http://example.org/book/ <book1></code>
<code>PREFIX book: http://example.org/book/ book:book1</code>

Figura 18: Ejemplo de uso de BASE y PREFIX

Fuente : Qaissi, (2009)

PREFIX

- ◆ Equivalente a la declaración namespace en XML.
- ◆ Define los prefijos para los espacios de nombres.
- ◆ Asocia una URI a una etiqueta que se usará mas adelante en la consulta.
- ◆ Podemos tener cero o más de las siguientes cláusulas

PREFIX <prefijo> : <URI>

- ◆ Se hace referencia al prefijo con ":" en la consulta.
- ◆ Se puede definir un prefijo por defecto:

PREFIX : <URI>

b) Cláusulas Query Result Form

SELECT

- ◆ Cláusula requerida en toda consulta.
- ◆ Similar al uso especificado por el estándar SQL.
- ◆ Define las variables a retornar como resultados, cuyos nombres deben empezar con "?", y separadas por ","
ejemplo: **?apellido**

DESCRIBE

- ◆ Devuelve un grafo RDF describiendo los recursos encontrados.
- ◆ Puede ser fuente de datos para otra consulta.
- ◆ Es opcional.

ASK

- ◆ Devuelve una variable booleana indicando si la combinación Sujeto-Predicado-Objeto de consulta existe en la ontología RDF consultada.
- ◆ Es opcional.

c) Cláusulas Query DataSet Sources

FROM

- ◆ La consulta SPQRL da respuesta a un grafo RDF.
- ◆ La cláusula FROM identifica los datos sobre los que se ejecutará la consulta.
- ◆ Es opcional.

- ◆ En caso de no estar incluida en la consulta, el motor de proceso de consultas toma como fuente de datos todos los grafos RDF disponibles.
- ◆ En caso de múltiples FROM, la consulta se realiza sobre la fusión de todos los grafos RDF nombrados.

d) Cláusulas Graph Pattern

WHERE

- ◆ Patrón de la consulta con una o más tripletas encerradas entre {}.
- ◆ Cláusula requerida en caso del Query Result Form ASK.

OPTIONAL

- ◆ Para indicar patrones opcionales. Es opcional.
- ◆ Para imponer constricciones adicionales al patrón de búsqueda.

DISTINCT

- ◆ Asegura la unicidad de los resultados.

REDUCED

- ◆ Permite eliminar las respuestas duplicadas.

FILTER

- ◆ Para imponer restricciones adicionales al patrón de búsqueda.

e) Cláusulas. Query Results Ordering

ORDER BY

- ◆ Tiene la misma funcionalidad que la definida en SQL.
- ◆ Es opcional.

f) Cláusulas. Query Results Selection.

LIMIT n

- ◆ Restringe el número de soluciones devueltas.
- ◆ Es opcional.

OFFSET m

- ◆ Tiene la misma funcionalidad que la definida en SQL.
- ◆ Es opcional.

2.2.24. Ejemplos de consultas en SPARQL

<pre>@prefix foaf: <http://xmlns.com/foaf/0.1/> . _:a foaf:name "Johnny Lee Outlaw" . _:a foaf:mbox <mailto:jlow@example.com> . _:b foaf:name "Peter Goodguy" . _:b foaf:mbox <mailto:peter@example.org> . _:c foaf:mbox <mailto:carol@example.org> .</pre>	1						
<pre>PREFIX foaf: <http://xmlns.com/foaf/0.1/> SELECT ?name ?mbox WHERE { ?x foaf:name ?name . ?x foaf:mbox ?mbox }</pre>	2						
<table border="1"><thead><tr><th>name</th><th>mbox</th></tr></thead><tbody><tr><td>"Jonny Lee Outlaw"</td><td>mailto:jlow@example.com</td></tr><tr><td>"Peter Goodguy"</td><td>mailto:peter@example.org</td></tr></tbody></table>	name	mbox	"Jonny Lee Outlaw"	mailto:jlow@example.com	"Peter Goodguy"	mailto:peter@example.org	3
name	mbox						
"Jonny Lee Outlaw"	mailto:jlow@example.com						
"Peter Goodguy"	mailto:peter@example.org						

Figura 19: Ejemplo de consulta en SPARQL

Fuente : Qaissi (2009)

1. Corresponde a los datos del grafo RDF a consultar.
2. La consulta busca 2 variables, ambas corresponden a un objeto.
3. Es la respuesta y no contiene la 6ª línea del grafo RDF por no cumplir la condición.

```

@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .
:book1 dc:title "SPARQL Tutorial" .
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web" .
:book2 ns:price 23 .

```

```

PREFIX dc: <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE
{
    ?x dc:title ?title
    FILTER regex(?title, "^SPARQL")
}

```

title	price
"SPARQL Tutorial"	42

```

PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE {
    ?x ns:price ?price .
    FILTER (?price < 30.5)
    ?x dc:title ?title .
}

```

title	price
"The Semantic Web"	23

Figura 20 : Ejemplo de consulta en SPARQL usando FILTER

Fuente : Qaissi, (2009)

2.2.25. SPARQL/Update (SPARUL)

- ◆ Derivado de SPARQL para actualizar grafos RDF's.
- ◆ Cubre las sentencias Create, Update y Delete del conjunto CRUD.
- ◆ Desarrollado por Hewlett Packard.
- ◆ No es recomendación de W3C aún.
- ◆ Clausulas: INSERT, DELETE y MODIFY, LOAD, CLEAR, CREATE y DROP.

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
DELETE DATA FROM <http://example/bookStore>
{ <http://example/book3> dc:title "Fundamentals of Compiler Desing" }
INSERT DATA INTO <http://example/bookStore>
{ <http://example/book3> dc:title "Fundamentals of Compiler Design" }
```

Figura 21: Ejemplo de consulta en SPARQL usando FILTER

Fuente : Qaissi (2009)

Esta solicitud borra una tripleta y añade otra de la fuente RDF <http://example/bookStore>.

La diferencia entre MODIFY/INSERT/DELETE e INSERT DATA y DELETE DATA es que éstas últimas no

tienen una plantilla patrón (como en la Cláusula WHERE), y necesitan datos concretos.

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
DELETE { ?book ?p ?v }
WHERE {
    ?book dc:date ?date .
    FILTER ( ?date < "2000-01-01T00:00:00"^^xsd:dateTime )
    ?book ?p ?v
}
```

Figura 22 : Ejemplo de consulta usando FILTER

Fuente : Qaissi, (2009)

Esta consulta borra todas las tripletas de libros con fecha inferior al año 2000.

2.2.26. Herramientas de ontologías

También existen herramientas y programas para realizar anotaciones en páginas web con lenguajes de marcados propios. La mayoría de estos programas permiten describir el contenido de los documentos en forma de metadatos, soportados sobre una ontología representada en RDF Schema (RDFS) o basados en grafos conceptuales.

A continuación se menciona algunas de ellas:

- ◆ **DOME:** Distributed Ontology Management Environment.
- ◆ **JENA:** Es un formato no propietario que ofrece un marco de recursos Java para contribuir a las aplicaciones de la web semántica. Ofrece un entorno para RDF, Esquema RDF y OWL e incluye un motor basado en reglas de inferencia.
- ◆ **KAON:** Es un gestor de ontologías de código abierto. Incluye un conjunto de herramientas para crear y gestionar ontologías y otras herramientas para construir aplicaciones basadas en ontologías.
- ◆ **Kowari:** Escrita en Java y de código abierto que soporta RDF y OWL.
- ◆ **KPOntology:** Es una biblioteca para gestionar ontologías, permite usar diferentes ontologías. Permite utilizar los lenguajes RDF, OWL y ODE.
- ◆ **OntoEdit:** Para construir ontologías usando significaciones gráficas.
- ◆ **OntoLingua:** Provee un entorno colaborativo, un buscador, generador y modificador de ontologías.

- ◆ **Ontopia:** Ofrece un conjunto de herramientas para desarrollar y mantener ontologías.
- ◆ **OntoShare:** Es una ontología basada para la WWW basada en RDF y pensada para compartir conocimiento en un ambiente compartido.
- ◆ **SWOOP:** Editor de ontologías hipermedia.
- ◆ **PROTÉGÉ:** Editor de ontologías y bases de conocimiento, de código abierto para construir ontologías sobre RDF (S), OWL y XML Schema. Está basado en Java. Es uno de los editores más utilizados.

A continuación vamos a detallar las características del editor de ontologías **PROTÉGÉ**, así como los fundamentos en los que se sustenta y que justificarán su elección como editor de ontologías de esta monografía.

2.2.27. Protégé

Protégé es un editor de código abierto para construir ontologías y un marco general para representar el conocimiento, fue creada por la Universidad de Stanford.

Está desarrollada en JAVA y puede funcionar perfectamente bajo LINUX y WINDOWS.

Es software libre y de código abierto.

La plataforma Protégé soporta dos formas principales de modelado de ontologías a través del Editor Protégé-Frames y Editor Protégé-OWL. Ontologías Protégé pueden ser exportadas a una variedad de formatos incluyendo RDF(S), OWL, and XML Schema.

Protégé es extensible, y ofrece un ambiente plug-and-play que la convierte en una base flexible para la creación rápida de prototipos y desarrollo de aplicaciones.

Protégé cuenta con el apoyo de una fuerte comunidad de desarrolladores y académicos, gobierno y usuarios corporativos, que están utilizando Protégé para soluciones de conocimiento en ámbitos tan diversos como la biomedicina, recolección de inteligencia y modelado corporativo.

La interfaz de desarrollo tiene la siguiente apariencia:

La figura 23 muestra el editor de clases de Protégé, en el que es posible visualizar su estructura jerárquica y cuáles son las propiedades y las restricciones inherentes a ellas que caracterizan al concepto en la ontología del ejemplo, en este caso, de Pizza.

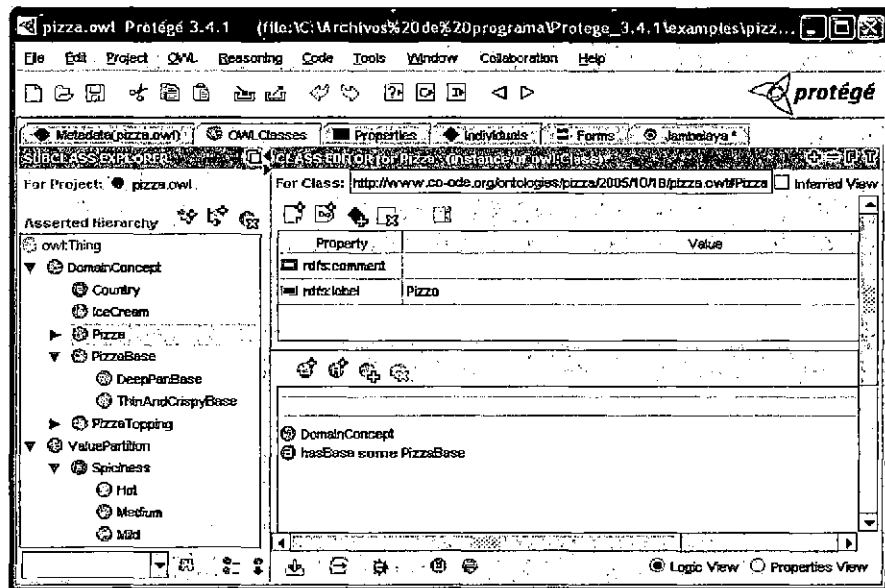


Figura 23 : Editor de clases de Protégé.

Fuente : Elaboración propia

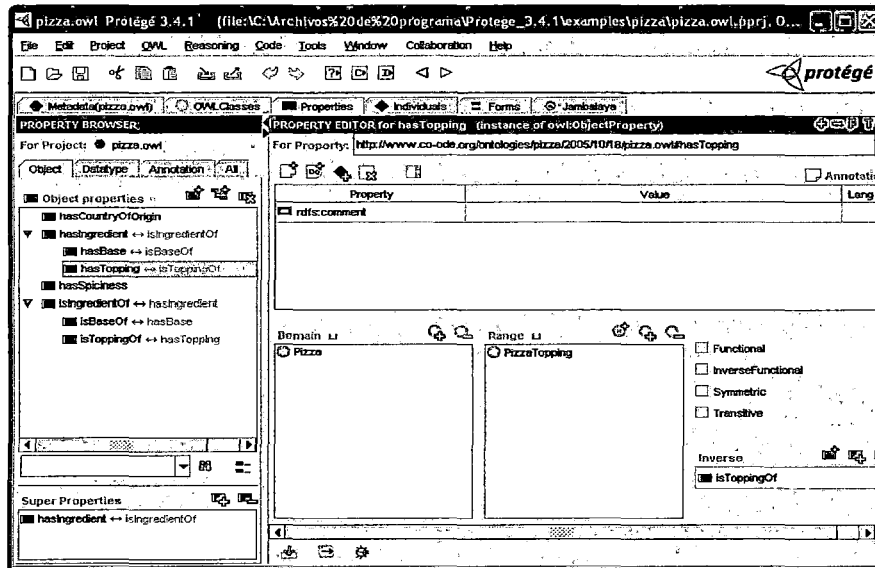


Figura 24: Propiedades de tipo objeto de la Ontología Pizza

Fuente : Elaboración propia

La figura 24 muestra las propiedades de tipo objeto de la ontología Pizza como hasCountryOfOrigin, hasIngredient, hasBase, hasTopping, hasSpiciness, isIngredientOf, isBaseOf, isToppingOf.

Ahora procedemos a visualizar los individuos que son instancias de las clases que tenemos, para esto vamos a Individuals. Se puede visualizar los individuos de la clase Country en la figura 25.

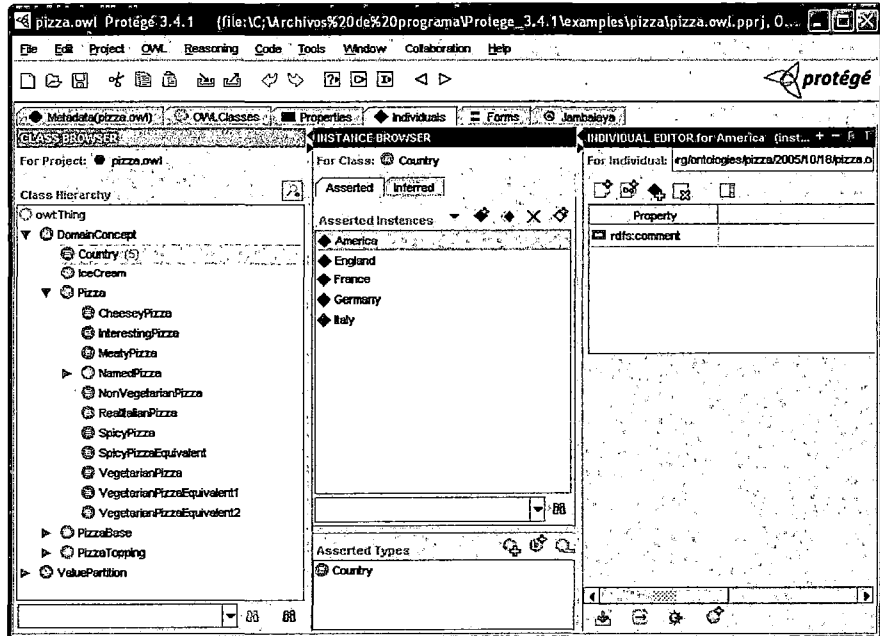


Figura 25 : Individuos de la clase Country

Fuente : Elaboración propia

Tabla 5: Protégé-OWL Syntax

OWL Element	Symbol	Key	Example	Meaning of example
allValuesFrom	\forall	*	children \forall Male	All children must be of type Male
someValuesFrom	\exists	?	children \exists Lawyer	At least one child must be of type Lawyer
hasValue	\ni	\$	rich \ni true	The rich property must have the value true
cardinality	=	=	children =3	There must be exactly 3 children
minCardinality	\geq	>	children \geq 3	There must be at least 3 children
maxCardinality	\leq	<	children \leq 3	There must be at most 3 children
complementOf	\neg	!	\neg Parent	Anything that is not of type Parent
intersectionOf	\cap	&	Human \cap Male	All Humans that are Male
unionOf	\cup		Doctor \cup Lawyer	Anything that is either Doctor or Lawyer
enumeration	{ ... }	{ }	{male female}	The individuals male or female

Fuente: Colomb, (2007)

2.2.28. Aplicaciones de las ontologías

Las ontologías son ahora ampliamente usadas en

- ◆ Ingeniería de Conocimiento
- ◆ Inteligencia Artificial
- ◆ Ciencia de la Computación

Aplicaciones relacionadas a la gestión del conocimiento:

- ◆ Procesamiento del lenguaje natural
- ◆ Comercio electrónico
- ◆ Inteligencia en la integración de la información

Intercambio de información a formatos de aplicaciones específicas. La posibilidad de traducir la información a formatos de otras aplicaciones, como pueden ser aplicaciones educativas, permite aumentar la rentabilidad de la codificación de la misma. Actualmente el gasto de las empresas en hacer compatibles a sistemas heterogéneos supone un 30% del gasto de toda la industria de tecnologías de la información.

- ◆ Recuperación de información

Recuperación de información mediante buscadores semánticos: las búsquedas semánticas, al contrario que las tradicionales; basadas en palabras clave; trabajan

con el significado de las palabras de acuerdo al modelo subyacente asegurando la precisión del 100% en las búsquedas. El resultado presentado al usuario pasa a ser la información solicitada en forma de conceptos del modelo, en lugar de los documentos posiblemente relacionados, tal como hacen los buscadores actuales.

- ◆ Integración de bases de datos.
- ◆ Bio-informática.
- ◆ Educación
- ◆ Web Semántica

2.2.29. Construyendo una ontología para la web

Tradicionalmente, las ontologías se construyen para representar conocimiento genérico acerca de un área de objetivo. Una ontología define un conjunto de términos representativos, referidas a manera de conceptos, que describe ideas abstractas en el área de objetivo y que puede estar relacionado a otro. Por ejemplo, en una ontología representando todas las criaturas vivas, “el humano” y “el mamífero” podrían ser dos de los conceptos

y estos dos conceptos podrían estar conectados con una relación “es un” (por ejemplo, el humano “es un” mamífero). Típicamente, los conceptos de las ontologías son representados como nodos en un grafo y las relaciones entre conceptos como arcos. Por ejemplo, la figura 26 muestra un fragmento de una ontología para el tema de **Arts**, representadas como una grafo acíclico dirigido, donde cada nodo denota un concepto que está interconectado a otros conceptos por una relación de especialización (“es un”), representado por los arco. Los conceptos que son asociados con un solo concepto padre por un enlace “es un” es considerado disjunto.

Dependiendo de la aplicación, hay formas diferentes de desarrollar una ontología. La utilidad, sin embargo, de una ontología consiste en el hecho de que presenta conocimiento en una fácil de forma para entender por la humanidad.

Parte de la ontología está ilustrado en la figura 26. Esta ontología tiene tres niveles diferentes: El nivel máximo

corresponde a los temas (Arts en nuestro caso), el nivel medio a los subtemas (por ejemplo photography, dance, etc), y nivel inferior corresponde a las jerarquías de WordNet, cuyos elementos son hipónimos del nivel medio de conceptos.

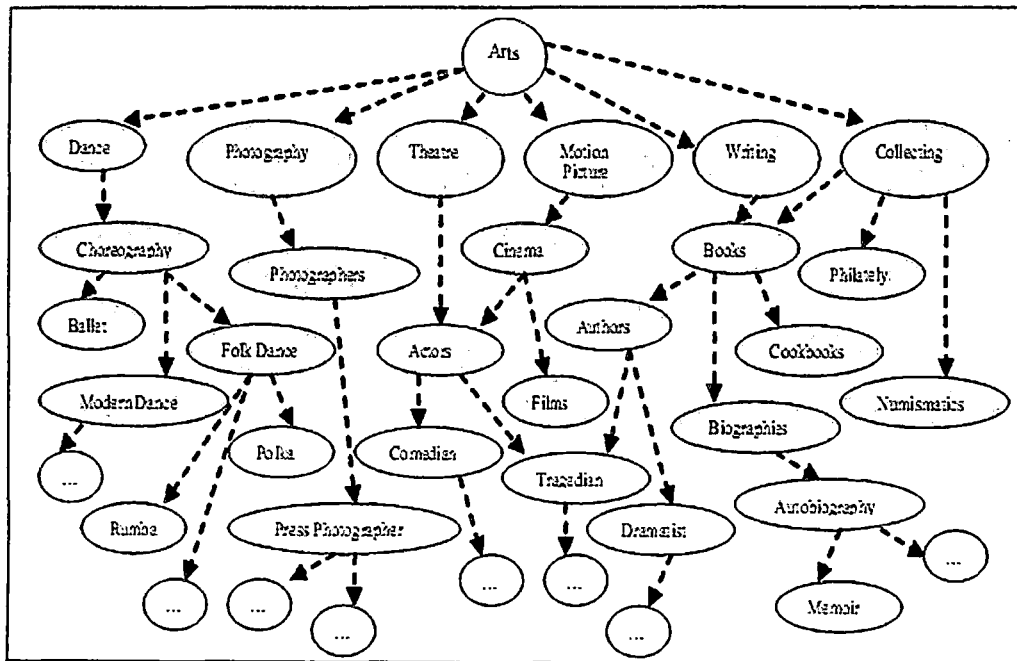


Figura 26 : Una porción de la ontología para el tema de artes.

Fuente : Nigro, Gonzales & Xodo, (2008).

2.2.30. Los sistemas y sus componentes

Un sistema se define como: “un conjunto de elementos interactivos, que pueden ser diseñados para que, en forma cooperativa, logren cumplir una función dada, o alcanzar propósitos determinados” (Levaggi, 1999).

Los componentes de los sistemas son los siguientes, aunque los mismos pueden variar como una combinación de elementos:

Elementos de entrada (inputs), que son los recursos que ingresan al sistema para ser procesados. Por ejemplo: materias primas, energía, datos, etc.

Procesamiento, que comprende los procesos de transformación que convierten los elementos de entrada en elementos de salida. Ejemplos: procesos industriales, cálculos matemáticos, etc.

Elementos de salida (outputs) es la transferencia de elementos que han sido generados mediante un proceso de transformación hacia su destino final. Ejemplos: productos terminados, servicios prestados, información procesada, etc.

Retroalimentación, que se refiere a la comunicación de los resultados y consecuencias de las acciones y decisiones a quien las origina.

Control, que comprende el monitoreo del desempeño del sistema y la evaluación de la retroalimentación para verificar los resultados alcanzados y compararlos con los previstos, determinar las desviaciones, también adoptar las medidas correctivas necesarias para regular la acción y asegurar el cumplimiento de objetivos.

Una característica importante de los sistemas es que éstos no existen en forma aislada, porque forman parte de un entorno que contiene otros sistemas, separándose de estos por el límite o frontera.

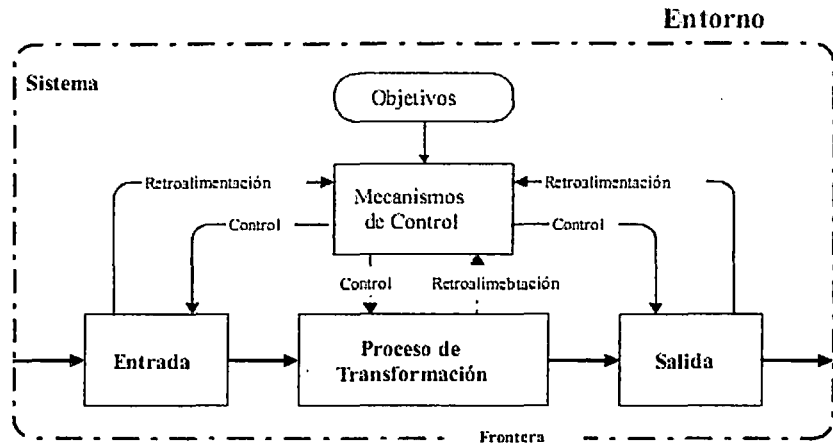


Figura 27 : Modelo general de un sistema

Fuente : Levaggi, (1999).

2.2.31. Los datos y la información

La transformación de datos en información está a cargo de un procesador de información (sistema), el cual puede incluir elementos computacionales, elementos no computacionales o una combinación de los dos.

Los atributos de la calidad de la información se enfocan a tres dimensiones:

- Tiempo
- Contenido
- Forma

Tabla 6: Atributos de la Información

Dimensión	Cualidad de la información
Tiempo:	
Oportunidad	La información debe suministrarse en el momento en que sea necesaria.
Actualidad	La información debe ser reciente al momento de suministrarse.
Frecuencia	La información debe suministrarse con la frecuencia que sea necesaria.
Período	La información puede proporcionarse sobre periodos pasados, presentes y futuros.
Contenido:	
Exactitud	La información debe estar libre de errores.
Pertinencia	La información debe estar relacionada con las necesidades de información de un destinatario específico para una determinada situación.
Integridad	Debe suministrarse toda la información que sea necesaria.
Brevidad	Debe proporcionarse sólo la información que se necesite, cuando se necesite.
Alcance	La información puede tener un alcance amplio o estrecho, o un enfoque interno o externo.
Desempeño	La información puede revelar el desempeño, al mediar las actividades logradas, el progreso alcanzado o los recursos acumulados.
Forma:	
Claridad	La información debe suministrarse en un formato que sea fácil de entender.
Detalle	La información puede proporcionarse en un formato detallado o resumido.
Orden	La información puede ordenarse en una secuencia predeterminada.
Presentación	La información puede presentarse en forma narrativa, numérica, gráfica u otras formas.
Medios	La información puede proporcionarse en la forma de documentos de papel impresos, presentaciones de video u otros medios.

Fuente : Colomb, (2007).

2.2.32. Los sistemas de información

El sistema de información tiene como fin “procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas” (Senn, 1995).

Por lo tanto podemos definir un sistema de información como un conjunto de subsistemas que incluyen hardware, software, medios de almacenamiento de datos ya sea primarios, secundarios y bases de datos relacionadas entre si con el fin de procesar entradas para realizar transformaciones a esas entradas y convertirlas en salidas de información importantes en la toma de decisiones.

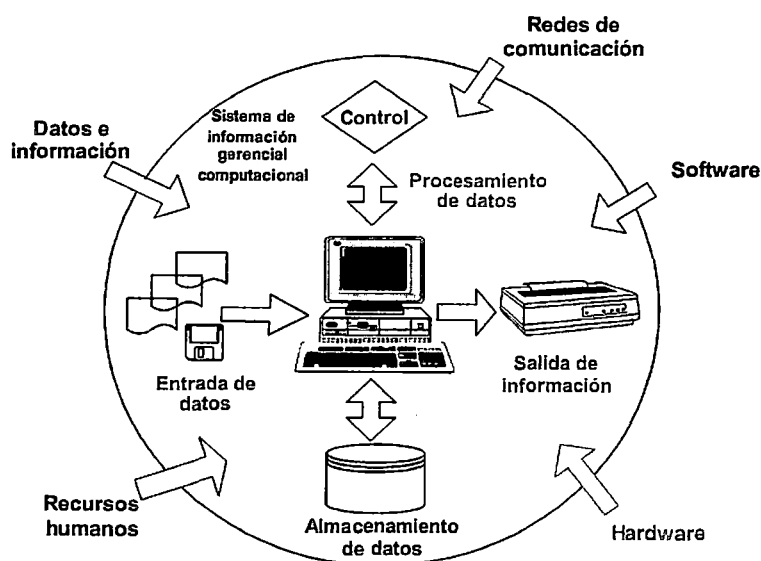


Figura 28 : Componentes de los sistema de información.

Fuente : Senn (2005).

El sistema de información forma parte de una serie de actividades de adición de valor para adquirir, transformar y distribuir información que los gerentes pueden utilizar para

mejorar la toma de decisiones, reforzar el desempeño de la organización y, a fin de cuentas, incrementar la rentabilidad de la empresa (Laudon & Laudon, 2004).

2.2.33. Características de los sistemas de información modernos

- ◆ Sistemas sencillos sirviendo a funciones y niveles múltiples dentro de la empresa.
- ◆ Acceso inmediato en línea a grandes cantidades de información.
- ◆ Fuerte confiabilidad en la tecnología de telecomunicaciones.
- ◆ Mayor cantidad de inteligencia y conocimientos implícita en los sistemas.
- ◆ La capacidad para combinar datos y gráficas.

2.2.34. Seis tipos principales de sistemas

La organización cuenta con sistemas de apoyo a ejecutivos (ESS, por sus siglas en inglés) en el nivel estratégico; sistemas de información gerencial (MIS, por

sus siglas en inglés) y sistemas de apoyo a la toma de decisiones (DSS, por sus siglas en inglés) en el nivel administrativo; sistemas de trabajo del conocimiento (KWS, por sus siglas en inglés) y sistemas de oficina en el nivel del conocimiento, y sistemas de procesamiento de transacciones (TPS, por sus siglas en inglés) en el nivel operativo. A su vez, los sistemas de cada nivel se especializan en apoyar a cada una de las principales áreas funcionales. Así, los sistemas típicos que se encuentran en las organizaciones están diseñados para asistir a los trabajadores y a los gerentes en cada nivel y en las funciones de ventas y marketing, manufactura, finanzas, contabilidad y recursos humanos (Kendall & Kendall, 1997).

Cuadro 2: Resumen de las características de los seis tipos de sistemas de información

Tipo de sistema	Entrada de información	Procesamiento	Salida de información	Usuarios
<i>ESS</i>	Datos acumulados; externos, internos	Gráficas; simulaciones; interactivo	Proyecciones; respuestas a consultas	Directores
<i>DSS</i>	Datos de bajo volumen o bases de datos masivas optimizadas para el análisis de datos; modelos analíticos y herramientas de análisis de datos	Interactivo; simulaciones; análisis	Informes especiales; análisis de decisiones; respuestas a consultas	Profesionales; gerentes de personal
<i>MIS</i>	Datos resumidos de transacciones; datos de alto volumen; modelos simples	Informes de rutina; modelos simples; análisis de bajo nivel	Informes resumidos y excepciones	Gerentes de nivel medio
<i>KWS</i>	Especificaciones de diseño; base del conocimiento	Modelado; simulaciones	Modelos; gráficos	Profesionales; personal técnico
<i>Sistema de oficina</i>	Documentos, programas	Administración de documentos; programación; comunicación	Documentos, programas; correo	Oficinistas
<i>TPS</i>	Transacciones, eventos	Clasificación; listados; fusión; actualización	Informes detallados; listas; resúmenes	Personal de operaciones; supervisores

Fuente: Kendall & Kendall (1997).

El cuadro 2 resume las características de los seis tipos de sistemas de información. Debe notarse que cada uno de los diversos sistemas podría tener componentes que se utilicen en niveles y grupos de organización distintos a los de su área principal. Una secretaria puede encontrar información en un MIS, y un gerente de nivel medio podría requerir datos de un TPS.

2.3. TÉRMINOS BÁSICOS

DAML (DARPA Agent Markup Language) El DAML es un lenguaje para modelar ontologías creado con una extensión de RDF. El DAML provee un rico juego de construcciones con el que crear ontologías y aumentar la información para que sea legible y comprensible por las máquinas. En diciembre de 2000 el DAML pasó a decirse DAML+OIL, debido a la revisión de las especificaciones del lenguaje. (Cabe recordar que el XML, por si sólo, no aporta la semántica suficiente para ser "comprensible" por las máquinas).

HTML (Hypertext Markup Language). Es el lenguaje de marcado que se utiliza para codificar el formato de presentación y enlaces de hipertexto de las páginas Web. Se basa en un sistema de etiquetas estandarizadas (tipo <H1> i <BODY>), el significado y la interpretación de las cuales son establecidas de forma universal por el Consorcio World Wide Web.

Ontologías. Las ontologías son colecciones de entidades que definen las relaciones entre conceptos y especifica reglas lógicas para realizar razonamientos sobre estas relaciones. Las máquinas

entenderán el significado de la semántica asociada a una página Web siguiendo links que especificarán las ontologías. Las ontologías también contienen la información sobre equivalencias semánticas. Por ejemplo, gracias a la ontología si buscas en un buscador la palabra "automóvil" como resultado también te ofrecerá lo que encuentre por la palabra "coche".

OWL *Web Ontology Language*, lenguaje de especificación de ontologías definido por W3C.

RDF (Resource Description Framework). Se trata de un modelo para definir relaciones semánticas entre distintas URIs. RDF está basado en la sintaxis XML, y permite describir semánticamente una URI asociándole un conjunto de propiedades y valores. Los modelos RDF se construyen como grafos dirigidos especificando "triples" (URI, propiedad, valor). Los metadatos especificados con RDF son "comprensibles por las máquinas, y por tanto, procesables de forma automática.

URI (Universal Resource Identifier). Un URI define o especifica una entidad (y puede ser cualquier cosa: documento, dispositivo,

vídeo, audio, imagen) Las URLs (Uniform Resource Locator), o sea, lo que todos conocemos como la dirección de una página Web (por ejemplo, <http://www.ibit.org>) , es el tipo más familiar de URI. Una URL es una dirección única que te permite visitar una página. Un URI se utiliza para explicar el significado de una entidad o concepto. Una palabra puede tener diferentes significados, pero sólo tiene un significado dentro de cada contexto donde se utiliza (por ejemplo "partida", es presupuestaria cuando está dentro de un contexto económico, y puede ser, por ejemplo, de "póker" cuando está dentro de un contexto de juego). Utilizando diferentes URIs para cada concepto específico se soluciona este problema. De esta forma se mejorará la búsqueda de información útil a través de buscadores, por ejemplo de un link o URL (ya que vendrán descritos, informando por que son interesantes, y quizás con una puntuación, a partir de la visita y el análisis de diferentes usuarios).

WordNet Gran base de datos léxica disponible en varios idiomas

XML (Extensible Markup Language). Es un lenguaje de marcado, basado en una sintaxis similar a HTML, que permite a los usuarios definir sus propias etiquetas y estructuras de documentos. El

potencial que ofrece XML es utilizado para el intercambio de datos, la creación de protocolos e infraestructuras para registros de empresas (ebXML y UDDI), y la adaptación de interfaces de presentación a múltiples dispositivos (WAP, PDA,...). Actualmente, organizaciones y empresas cooperan en la definición de conjuntos de etiquetas para sectores de la industria específicos (turismo, banca, transporte, medicina,...), coordinados por el organismo sin ánimo de lucro OASIS.

CAPÍTULO III

MARCO METODOLÓGICO

3.1. TIPO Y DISEÑO DE LA INVESTIGACIÓN

3.1.1. Tipo de investigación

Esta investigación es de carácter prospectivo porque los datos recogidos son a propósito (primarios). Es transversal porque todas las variables son medidas al mismo tiempo, para realizar una comparación en muestras independientes.

Es analítico porque veremos en qué medida un sistema de información ontológico tiene relación sobre la administración académica.

3.1.2. Tratamiento y análisis de la información

Para medir el grado de eficiencia y usabilidad en la administración académica de la facultad de Ciencias de la UNJBG se desarrolló un cuestionario de diez preguntas, las cuatro primeras corresponden a la medida del indicador eficiencia y las seis preguntas siguientes al indicador usabilidad. Para el desarrollo del cuestionario y buscar la medida de los indicadores se utilizó la escala de Likert de cinco valores.

3.1.3. Diseño de la investigación

R G1: O1 X O2

R G2: O1 O2

3.1.4. Diseño experimental o no experimental

Experimental, cuasi-experimental.

3.2. POBLACIÓN Y MUESTRA

3.2.1. Población

Población de docentes de las cuatro escuelas de la Facultad de Ciencias; un total de 94 docentes; de la Universidad Nacional Jorge Basadre Grohmann de Tacna.

3.2.2. Muestra

Se ha analizado la variable independiente (Sistema de Información Ontológico) en una muestra de 47 docentes, la otra muestra será el grupo de control para comparar los resultados.

El objetivo de estas dos muestras es comparar, y analizar qué relación tiene el Sistema de Información Ontológico en la administración académica de la facultad de Ciencias de la UNJBG.

3.3. OPERACIONALIZACIÓN DE VARIABLES

3.3.1. Variables de estudio:

Variable Independiente: Sistema de Información Ontológico: Prototipo de Sistema de información basado en el diseño de una ontología.

Variable Dependiente: Administración Académica: Se refiere a todo lo concerniente al manejo académico específicamente a la investigación de los docentes de la Facultad de Ciencias de la UNJBG.

3.3.2. Operacionalización de variables

Variable independiente: Sistema de Información Ontológico

Indicadores:

- Funcionalidad.
- Estandarización.

Variable dependiente: Administración Académica

Indicadores:

- Eficiencia.
- Usabilidad

3.3.3. Clasificación de las variables

Variable independiente: Sistema de Información Ontológico:

- Por la función que cumple en la hipótesis: Independiente.
- Por su naturaleza: Cuantitativo.
- Por la escala: de Intervalo.

- Por la posesión de la característica: Discreto.

Variable dependiente: Administración Académica:

- Por la función que cumple en la hipótesis: Dependiente.
- Por su naturaleza: Cuantitativo.
- Por la escala: Intervalo.
- Por la posesión de la característica: Discreto.

3.4. TÉCNICAS DE RECOLECCIÓN DE INFORMACIÓN

3.4.1. Técnica de recolección de datos

3.4.1.1. Métodos interactivos:

- Encuestas
- Entrevistas.
- JAD (Join Application developement)

3.4.1.2. Métodos no intrusivos:

- Observación
- Revisión de la documentación

3.4.2. Instrumento de recolección de datos

Encuesta validada por expertos véase al Anexo A y B

3.4.3. Construcción de la encuesta

La encuesta (Anexo A) cuenta con diez preguntas cerradas. Para su construcción, se ha considerado la escala de Likert de cinco dimensiones.

Ponderación de Likert

Alternativas:

1 = Muy Malo;

2 = Malo;

3 = Regular;

4 = Bueno;

5 = Muy Bueno

Puntuación (P):

$$P = \frac{PT}{NT}$$

Donde:

PT = Puntuación Total.

NT = Número total de afirmaciones.

Evaluación

Si el valor de P es menor e igual a 3 entonces diremos que el docente está insatisfecho, de manera alterna cuando el valor de P es mayor a 3 diremos que el docente está satisfecho.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

4.1. PRESENTACIÓN, ANÁLISIS E INTERPRETACIÓN DE LOS DATOS

Comparación de datos en dos grupos independientes.

Número de Variables: Bivariable.

La primera variable es el Sistema de Información Ontológico y la segunda variable es la Administración Académica.

Carácter: Exploratorio.

Naturaleza: Cualitativo

Medir el grado de eficiencia de la administración académica.

Objetivo: Causal.

4.1.1. RESULTADOS DE LA ENCUESTA

Datos de los indicadores

De la eficiencia sin software. Los docentes de la facultad de Ciencias consideran que la eficiencia en la administración académica es:

Cuadro 3: Eficiencia sin software, ponderación de Likert.

Docentes/Preguntas	Promedio de Likert				
	P1	P2	P3	P4	Puntuación
Docente 1	2	4	5	2	3,25
Docente 2	4	3	2	3	3
Docente 3	5	1	3	4	3,25
Docente 4	5	3	2	4	3,5
Docente 5	4	4	4	2	3,5
Docente 6	2	1	2	3	2
Docente 7	5	5	4	3	4,25
Docente 8	1	2	2	3	2
Docente 9	1	3	3	3	2,5
Docente 10	3	3	2	2	2,5
Docente 11	3	1	1	5	2,5
Docente 12	2	5	1	1	2,25
Docente 13	5	3	3	4	3,75
Docente 14	2	5	5	3	3,75
Docente 15	4	1	5	3	3,25
Docente 16	4	3	1	2	2,5
Docente 17	2	2	5	3	3
Docente 18	1	4	2	1	2
Docente 19	3	5	5	1	3,5
Docente 20	3	5	2	1	2,75

Continua...

Docente 21	3	2	5	3	3,25
Docente 22	4	5	5	4	4,5
Docente 23	4	4	5	1	3,5
Docente 24	1	1	5	2	2,25
Docente 25	4	3	5	2	3,5
Docente 26	4	2	5	4	3,75
Docente 27	3	2	4	2	2,75
Docente 28	5	2	4	5	4
Docente 29	3	5	2	3	3,25
Docente 30	4	3	5	1	3,25
Docente 31	5	3	3	2	3,25
Docente 32	5	2	2	1	2,5
Docente 33	2	2	3	4	2,75
Docente 34	2	5	2	5	3,5
Docente 35	5	2	4	2	3,25
Docente 36	4	4	2	4	3,5
Docente 37	3	3	4	4	3,5
Docente 38	3	5	4	5	4,25
Docente 39	2	5	3	4	3,5
Docente 40	3	5	1	1	2,5
Docente 41	5	3	5	4	4,25
Docente 42	5	3	4	2	3,5
Docente 43	4	2	2	3	2,75

Continúa ...

Docente 44	1	4	2	4	2,75
Docente 45	4	2	5	4	3,75
Docente 46	5	4	3	5	4,25
Docente 47	4	2	4	1	2,75

Fuente: Elaboración propia

En donde:

1 = Muy Malo; 2 = Malo; 3 = Regular; 4 = Bueno;

5 = Muy Bueno

Puntuación (P):

$1 \leq P \leq 3 \rightarrow$ Insatisfecho

$3 < P \leq 5 \rightarrow$ Satisfecho

Cuadro 4: Resumen eficiencia sin software

Evaluación	Cantidad	Porcentaje
Insatisfecho	28	59,5744681
Satisfecho	19	40,4255319

Fuente: Elaboración propia

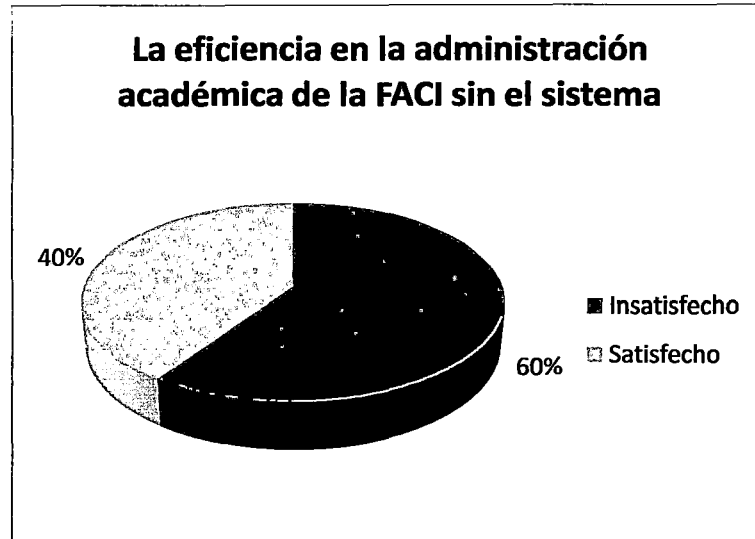


Figura 29: Resumen eficiencia sin software

Fuente : Elaboración propia.

Los resultados obtenidos indican que de los 47 docentes encuestados solo 19 (40,42%) se consideran satisfechos con el grado de eficiencia en la administración académica de la facultad de Ciencias sin usar el software.

De la usabilidad sin software. Los docentes de la Facultad de Ciencias consideran que la usabilidad en el sistema de administración académica de es:

Cuadro 5: Usabilidad sin software, ponderación de Likert.

Docentes / Preguntas	Promedio de Likert						Puntuación
	P5	P6	P7	P8	P9	P10	
Docente 1	5	4	4	2	1	3	3,5
Docente 2	2	1	4	4	1	3	2
Docente 3	1	1	4	4	1	3	1,75
Docente 4	4	1	1	4	1	4	1,75
Docente 5	4	2	4	4	3	1	3,25
Docente 6	4	3	1	4	5	1	3,25
Docente 7	2	3	2	1	3	5	2,5
Docente 8	2	3	4	2	4	4	3,25
Docente 9	4	2	4	3	3	4	3,25
Docente 10	2	3	3	1	5	2	3,25
Docente 11	5	2	4	1	4	2	3,75
Docente 12	1	3	4	4	5	1	3,25
Docente 13	1	2	5	3	5	5	3,25
Docente 14	1	1	2	3	2	4	1,5
Docente 15	2	5	2	4	4	2	3,25
Docente 16	2	3	4	4	2	1	2,75

Continua ...

Docente 17	2	1	5	4	2	1	2,5
Docente 18	3	2	2	3	3	1	2,5
Docente 19	3	1	1	2	5	2	2,5
Docente 20	4	4	3	3	2	1	3,25
Docente 21	5	4	4	4	5	5	4,5
Docente 22	2	1	2	5	2	1	1,75
Docente 23	5	4	1	2	2	2	3
Docente 24	5	3	5	4	2	4	3,75
Docente 25	4	3	2	3	5	4	3,5
Docente 26	4	2	2	3	5	5	3,25
Docente 27	2	3	1	5	3	5	2,25
Docente 28	4	5	4	4	5	1	4,5
Docente 29	4	5	3	4	4	5	4
Docente 30	1	3	3	1	1	3	2
Docente 31	1	5	2	3	3	3	2,75
Docente 32	3	2	4	2	5	4	3,5
Docente 33	4	1	5	2	4	3	3,5
Docente 34	5	3	5	1	4	1	4,25
Docente 35	4	3	1	4	5	4	3,25

Continúa ...

Docente 36	4	1	3	1	1	2	2,25
Docente 37	3	3	1	1	5	3	3
Docente 38	5	4	5	2	2	2	4
Docente 39	5	3	1	1	5	2	3,5
Docente 40	2	3	1	5	5	5	2,75
Docente 41	2	4	1	3	3	2	2,5
Docente 42	5	1	5	1	2	4	3,25
Docente 43	3	2	3	3	1	1	2,25
Docente 44	5	3	3	2	2	5	3,25
Docente 45	3	5	1	4	5	1	3,5
Docente 46	5	2	4	2	1	5	3
Docente 47	3	3	3	4	3	2	3

Fuente: Elaboración propia

En donde:

1 = Muy Malo; 2 = Malo; 3 = Regular; 4 = Bueno; 5 = Muy Bueno

Puntuación (P):

$1 \leq P \leq 3 \rightarrow$ Insatisfecho

$3 < P \leq 5 \rightarrow$ Satisfecho

Cuadro 6: Resumen usabilidad sin software

Evaluación	Cantidad	Porcentaje
Insatisfecho	26	55,3191489
Satisfecho	21	44,6808511

Fuente: Elaboración propia

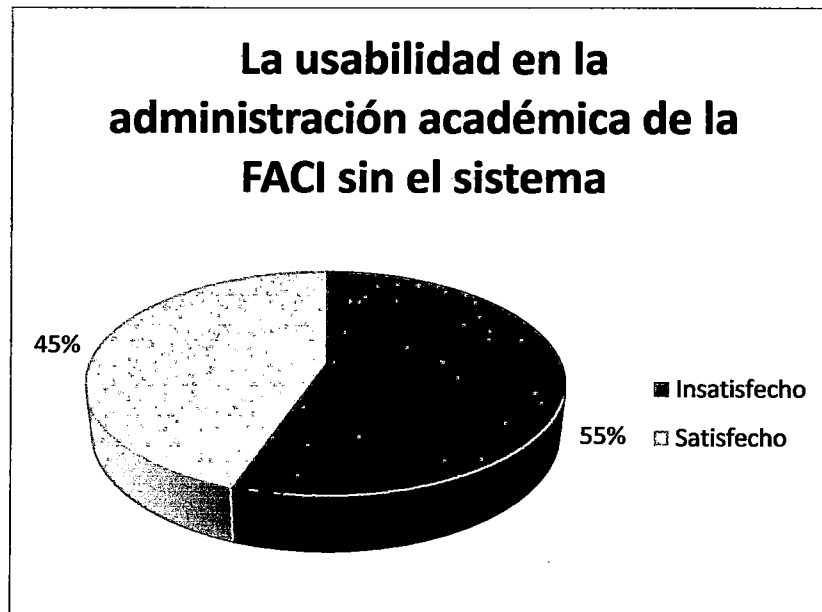


Figura 30: Resumen usabilidad sin software

Fuente : Elaboración propia

Los resultados obtenidos indican que de los 47 docentes encuestados solo 21 (44,7%) se consideran

satisfechos con el grado de usabilidad en la administración académica de la Facultad de Ciencias sin usar el software.

De la eficiencia con software. Los docentes de la facultad de Ciencias consideran que la eficiencia en la administración académica utilizando el sistema de información ontológico:

Cuadro 7: Eficiencia con software, ponderación de Likert.

Docentes /Alumnos	Promedio de Likert				
	P1	P2	P3	P4	Puntuación
Docente 1	2	2	5	4	3,25
Docente 2	5	1	1	3	2,5
Docente 3	3	3	1	2	2,25
Docente 4	4	3	2	4	3,25
Docente 5	2	2	1	1	1,5
Docente 6	2	3	3	2	2,5
Docente 7	2	1	1	3	1,75
Docente 8	5	2	2	3	3
Docente 9	2	2	1	2	1,75
Docente 10	4	5	4	4	4,25

Continua ...

Docente 11	5	4	5	4	4,5
Docente 12	2	2	5	2	2,75
Docente 13	4	3	3	1	2,75
Docente 14	3	5	4	2	3,5
Docente 15	4	2	2	5	3,25
Docente 16	2	1	4	3	2,5
Docente 17	2	4	5	3	3,5
Docente 18	2	2	4	1	2,25
Docente 19	3	1	3	2	2,25
Docente 20	3	4	4	2	3,25
Docente 21	1	3	5	3	3
Docente 22	4	4	3	1	3
Docente 23	3	5	1	2	2,75
Docente 24	5	1	1	3	2,5
Docente 25	5	4	4	1	3,5
Docente 26	3	4	3	1	2,75
Docente 27	2	2	3	2	2,25
Docente 28	3	5	1	5	3,5
Docente 29	5	4	2	4	3,75

Continua ...

Docente 30	4	3	5	2	3,5
Docente 31	4	1	5	2	3
Docente 32	2	2	3	3	2,5
Docente 33	3	5	1	2	2,75
Docente 34	3	2	2	5	3
Docente 35	3	4	2	3	3
Docente 36	4	1	3	2	2,5
Docente 37	4	5	3	2	3,5
Docente 38	1	3	5	1	2,5
Docente 39	4	2	1	4	2,75
Docente 40	2	3	2	2	2,25
Docente 41	4	4	1	1	2,5
Docente 42	3	2	4	5	3,5
Docente 43	1	2	3	4	2,5
Docente 44	1	2	1	3	1,75
Docente 45	4	2	5	3	3,5
Docente 46	5	4	4	5	4,5
Docente 47	3	5	2	1	2,75

Fuente: Elaboración propia

En donde:

1 = Muy Malo; 2 = Malo; 3 = Regular; 4 = Bueno;

5 = Muy Bueno

Puntuación (P):

1 <= P <= 3 -> Insatisfecho

3 < P <= 5 -> Satisfecho

Cuadro 8: Resumen eficiencia con software

Evaluación	Cantidad	Porcentaje
Insatisfecho	16	34,0425532
Satisfecho	31	65,9574468

Fuente: Elaboración propia

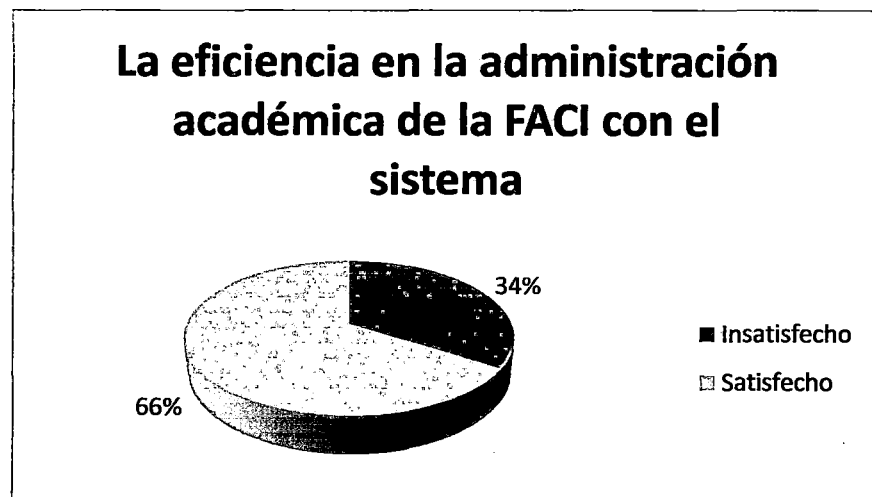


Figura 31: Resumen eficiencia con software

Fuente : Elaboración propia

Los resultados obtenidos indican que de los 47 docentes encuestados, 31 (65,95%) se consideran satisfechos con el grado de eficiencia en la administración académica de la facultad de Ciencias usando el software.

De la usabilidad con software. Los docentes de la facultad de Ciencias consideran que la usabilidad en la administración académica de la facultad de Ciencias utilizando un sistema de información ontológico es:

Cuadro 9: Usabilidad con software, ponderación de Likert.

Docentes / Alumnos	Promedio de Likert						Puntuación
	P5	P6	P7	P8	P9	P10	
Docente 1	1	1	1	5	3	2	1,5
Docente 2	2	2	2	4	4	1	2,5
Docente 3	4	1	3	2	3	2	2,75
Docente 4	1	2	1	2	2	3	1,5
Docente 5	2	2	1	4	5	4	2,5
Docente 6	2	4	3	4	3	2	3
Docente 7	4	5	2	2	4	2	3,75

Continua ...

Docente 8	5	1	2	3	1	2	2,25
Docente 9	1	1	5	1	4	4	2,75
Docente 10	1	3	4	3	1	3	2,25
Docente 11	3	1	1	2	3	1	2
Docente 12	3	3	1	2	2	1	2,25
Docente 13	1	4	3	3	2	2	2,5
Docente 14	2	3	2	5	1	4	2
Docente 15	1	2	5	4	4	5	3
Docente 16	1	2	1	4	3	1	1,75
Docente 17	5	5	1	5	5	5	4
Docente 18	1	2	3	5	3	3	2,25
Docente 19	3	3	1	3	1	4	2
Docente 20	4	5	2	5	1	1	3
Docente 21	2	2	4	2	4	1	3
Docente 22	5	2	3	4	2	2	3
Docente 23	3	4	4	3	3	3	3,5
Docente 24	3	4	1	4	4	1	3
Docente 25	3	4	5	2	4	3	4
Docente 26	1	2	5	1	2	5	2,5

Continua ...

Docente 27	4	1	5	4	1	2	2,75
Docente 28	5	1	5	4	2	4	3,25
Docente 29	1	2	4	4	3	5	2,5
Docente 30	2	2	3	1	1	3	2
Docente 31	4	2	3	3	1	5	2,5
Docente 32	2	4	4	3	4	2	3,5
Docente 33	5	4	1	4	3	5	3,25
Docente 34	4	5	5	3	2	5	4
Docente 35	4	1	2	3	1	4	2
Docente 36	2	5	4	1	2	3	3,25
Docente 37	5	2	3	2	4	5	3,5
Docente 38	5	4	5	3	1	3	3,75
Docente 39	3	1	1	4	4	4	2,25
Docente 40	4	4	1	1	5	4	3,5
Docente 41	1	3	2	4	1	3	1,75
Docente 42	2	3	4	3	5	2	3,5
Docente 43	3	4	1	2	2	5	2,5
Docente 44	2	3	2	2	2	2	2,25
Docente 45	1	4	2	5	2	1	2,25
Docente 46	5	2	3	4	2	4	3
Docente 47	2	5	1	4	2	1	2,5

Fuente: Elaboración propia

En donde:

1 = Muy Malo; 2 = Malo; 3 = Regular; 4 = Bueno;

5 = Muy Bueno

Puntuación (P):

$1 \leq P \leq 3 \rightarrow$ Insatisfecho

$3 < P \leq 5 \rightarrow$ Satisfecho

Cuadro 10: Resumen usabilidad con software

Evaluación	Cantidad	Porcentaje
Insatisfecho	13	27,6595745
Satisfecho	34	72,3404255

Fuente: Elaboración propia

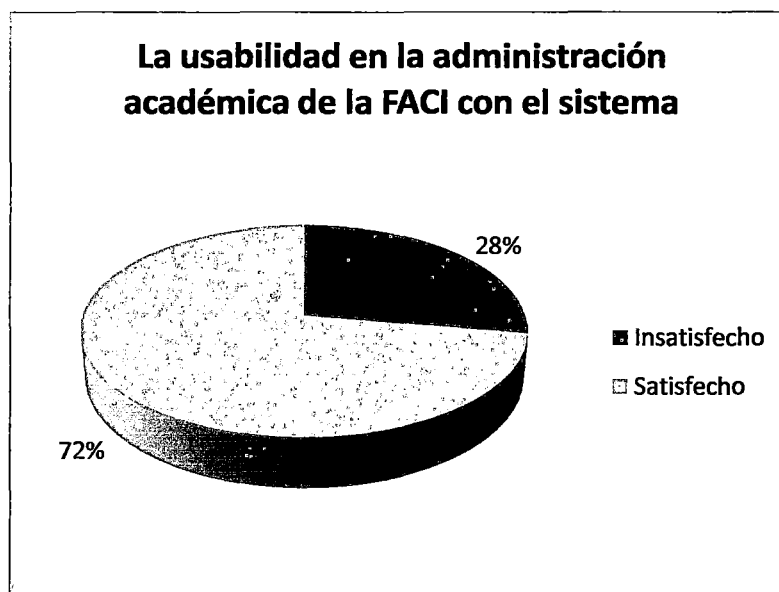


Figura 32: Resumen usabilidad con software

Fuente : Elaboración propia

Los resultados obtenidos indican que de los 47 docentes encuestados, 34 (72,3%) se consideran satisfechos con el grado de usabilidad en administración académica de la facultad de Ciencias usando el software.

Resultados de los datos obtenidos.

- ◆ El número de docentes satisfechos, con la eficiencia de la administración académica, se ha incrementado de 19 a 31 o sea en (26%).
- ◆ El número de docentes satisfechos con la usabilidad de la administración académica, se ha incrementado de 21 a 34 o sea en 13 (27%).

4.1.2. Contrastación de la hipótesis

DISCUSIÓN

Cumplimiento de los objetivos

“Determinar el nivel de dependencia entre el sistema de información basado en Ontologías y el soporte a la administración académica de la facultad de Ciencias”.

Este objetivo logró, determinar la fuerte dependencia de un Sistema de Información Ontológica en la administración académica de la facultad de ciencias además de brindar recomendaciones sobre la implantación del sistema mencionado.

Objetivos Específicos

“Determinar el nivel de dependencia de la ontología del dominio de aplicación con el grado de eficiencia la administración académica de la facultad de Ciencias.”

Este objetivo logró, determinar la no independencia que existe entre el sistema de información ontológica y la eficiencia de la administración académica de la facultad de ciencias.

“Determinar el nivel de dependencia entre la calidad de producto del sistema de información basado en ontologías utilizando RAP (RDP API for PHP) y el grado de usabilidad del sistema de administración académica de la facultad de Ciencias.”

Este objetivo logró, determinar la dependencia de un sistema de información ontológico en la usabilidad del sistema de administración académica de la facultad de ciencias, además de ser una cualidad para mostrar la satisfacción de los docentes sobre el sistema mencionado.

Prueba de Hipótesis : Hipótesis Global

H: "El sistema de información basado en ontologías no tendrá independencia con el soporte a la administración académica de la facultad de Ciencias."

Evaluación de las subhipótesis

A:

Ho: La estandarización de la ontología del dominio de aplicación no tendrá independencia con el grado de eficiencia de la administración académica de la facultad de ciencias.

H1: La estandarización de la ontología del dominio de aplicación tendrá independencia con el grado de eficiencia de la administración académica de la facultad de ciencias.

Cuadro 11: Resumen eficiencia en dos grupos

		Estadísticos de grupo			
Software		N	Media	Desviación típ.	Error típ. de la media
Eficiencia	"Sin Uso de SIO"	47	3,1809	,64627	,09427
	"Con Uso de SIO"	47	2,8883	,67124	,09791

Fuente: Elaboración propia.

Nivel de significancia: $\alpha = 0,05$ (5%)

T-Student de tabla (esperado): 3,841

T-Student calculado: 2,152

Entonces como $X_{\text{calculado}} < X_{\text{esperado}}$ cae en la zona de aceptación en consecuencia se acepta la hipótesis nula y se rechaza la hipótesis alterna, es decir, la estandarización de la ontología del dominio de aplicación no es independiente con el grado de eficiencia de la administración académica de la facultad de ciencias.

B:

Ho: El producto del sistema de información basado en ontologías utilizando RAP (RDP API for PHP) no tendrá

independencia con el grado de usabilidad del sistema de administración académica de la Facultad de Ciencias.

H1: El producto del sistema de información basado en ontologías utilizando RAP (RDP API for PHP) tendrá independencia con el grado de usabilidad del sistema de administración académica de la Facultad de Ciencias.

Cuadro 12: Resumen usabilidad en dos grupos

		Estadísticos de grupo			
Software		N	Media	Desviación típ.	Error típ. de la media
Usabilidad	"Sin Uso de SIO"	47	3,0266	,71520	,10432
	"Con Uso de SIO"	47	2,7287	,67331	,09821

Fuente: Elaboración propia

Nivel de significancia: alfa =0,05 (5%)

T-Student de tabla (esperado):3,841

T-Student calculado: 2,079

Si $X_{\text{calculado}} < X_{\text{esperado}}$ entonces cae en la zona de aceptación, por lo tanto, se acepta la hipótesis nula y se rechaza la hipótesis alterna, es decir, el producto del sistema

de información basado en ontologías utilizando RAP (RDP API for PHP) no es independiente con el grado de usabilidad del sistema de administración académica de la Facultad de Ciencias.

Por lo tanto:

$(A) \vee (B) = H$

$(V) \vee (V) = V$

Se acepta la hipótesis H: “El sistema de información basado en ontologías no es independiente con el soporte a la administración académica de la facultad de Ciencias.”

CONCLUSIONES

- Se pudo determinar que existe una dependencia entre el sistema basado en ontologías y el soporte a la administración académica de la facultad de Ciencias, esto se comprobó con el nivel de satisfacción de los usuarios e indicadores de gestión positivos que fueron observados en el post-test. Esto significa que nuestra propuesta funciona y es adecuada.
- Además se logró medir el grado de dependencia entre la definición de la ontología y el nivel de eficiencia en la administración académica de la facultad de Ciencias.
- También se observó el alto grado de dependencia entre la calidad de producto del sistema de información basado en ontologías utilizando RAP (RDF API for PHP) y el grado de usabilidad del sistema de administración académica de la facultad de Ciencias.

- Se creó una ontología llamada Gestión Académica para la Facultad de Ciencias. donde se muestra el vocabulario explícito del dominio de aplicación.

RECOMENDACIONES

- Es importante insistir en el desarrollo de sistemas de información basados en ontologías para la administración académica de las Facultades y de toda la Universidad Nacional Jorge Basadre Grohmann de Tacna.
- En lo que concierne al diseño de la ontología aun puede refinarse más este proceso de manera que el vocabulario definido sea más estándar y flexible además de reutilizable.
- El prototipo que se desarrollo aun está en fase inicial, se recomienda poder realizar trabajos futuros basados en esta propuesta.
- En cuanto al uso de RAP, considero que fue una buena elección, sin embargo como mostramos a en el marco teórico existen otras plataformas algunas basadas en Java que entiendo podrían ser mucho más robustas y portables.

- Sugiero que se investigue más acerca de ontologías porque es un tema nuevo y que recién comienza a cobrar importancia, y considero que se convertirá en una herramienta semántica muy potente y usada.

REFERENCIAS BIBLIOGRÁFICAS

Libros:

- ❖ Calero C., Ruiz F., & Piattini M. (2006). *Ontologies for Software Engineering and Software Technology*. NY: Springer.
- ❖ Cohen, D. (1998). *Sistemas de información para la toma de decisiones*, 2da Ed., México: Mc Graw-Hill.
- ❖ Colomb, R. (2007). "Ontology and the Semantic Web". Amsterdam, Netherlands: IOS Press.
- ❖ Gómez A., Fernández-López M., & Corcho O. (2004). *Ontological Engineering*. Primera Edición. London: Springer.
- ❖ Kendall K., & Kendall J. (1997). *Análisis y diseño de sistemas*. Tercera edición, México : Prentice-Hall.
- ❖ Laudon, K., & Laudon, J. (2004). *Sistemas de Información Gerencial*. Octava Edición. México : Prentice Hall.
- ❖ Levaggi, G. (1999). *Teoría General de Sistemas*. Buenos Aires : Editorial UGERMAN.
- ❖ Márquez Solís, S. (2009). *La Web Semántica*. Primera Edición.

- ❖ Münch Galindo, L. (1997). Fundamentos de administración: Casos y Prácticas. Segunda edición. México : Trillas.
- ❖ Nigro H., Gonzales S. & Xodo D.(2008). Data Mining with Ontologies: Implementations, findings and frameworks”. Hershey, PA: Information Science Reference.
- ❖ O’Brien, J., (2001). Sistemas de Información Gerencial. Cuarta Edición. Bogotá: Irwin-McGraw-Hill.
- ❖ Reyes Ponce, A.(1997). Administración moderna, Primera edición, México : LIMUSA.
- ❖ Senn, J. (1995). “Análisis y diseño de sistemas de información”, Segunda edición, México: Mc Graw-Hill.
- ❖ Waterfield, Ch., Ramsing, N.(1998). Management Information Systems for Microfinance institutions: A Handbook. Technical Tool Series N°1 . CGACP/World Bank.

Artículos de Revistas:

- ❖ Barchini, G., Álvarez M., & Herrera S. (2006). Sistemas de Información: Nuevos Escenarios Basados en Ontologías. *Journal of Information Systems and Technology Management*, 3(1), 3-18.
- ❖ Castells P. (2003). La Web Semántica. *Sistemas Interactivos y Colaborativos en la Web*, 195-212.

- ❖ Gruber, T. R. (1993). Towards Principles for Design of Ontologies Used for Knowledge Sharing. *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Disponible en: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.6025&rep=rep1&type=pdf> (visitado en Enero 2011).
- ❖ Heeks, R. (1998). Information System for Public Sector Management, Working Paper Series, Paper N°5. University of Manchester. Disponible en: http://www.sed.manchester.ac.uk/idpm/research/publications/wp/igovernment/documents/igov_wp05.pdf (visitado en Enero del 2011).
- ❖ Hernández, H., & Saiz, M. (2007). Ontologías mixtas para la representación conceptual de objetos de aprendizaje. *Procesamiento del lenguaje natural*. 38, 99-106. Disponible en : www.sepln.org/revistaSEPLN/revista/38/11.pdf (visitado en Diciembre del 2010).
- ❖ Neches R., Fikes R., Finin T., Gruber T., Patil R., Senator T., & Swartout W. (1991). Enabling Technology for Knowledge Sharing. *AI Magazine*. 12(3):37-56 Disponible en: <http://tomgruber.org/writing/AIMag12-03-004.pdf> (visitada en Octubre 2010).

- ❖ Peis, E., Herrera-Viedma, E., Hassan, Y., & Herrera, J.C. (2003). Ontologías, metadatos y agentes: recuperación semántica de la información. En: Actas de las II Jornadas de Tratamiento y Recuperación de la Información (JOTRI), pp. 157-165. Disponible en: <http://www.nosolousabilidad.com/hassan/jotri2003.pdf> (visitada en Diciembre del 2010)
- ❖ Smith, B. *Ontology and Information Systems*. Pág. 1-5. Disponible en: [http://ontology.buffalo.edu/ontology\(PIC\).pdf](http://ontology.buffalo.edu/ontology(PIC).pdf) (visitado en Enero 2011)

Tesis:

- ❖ Bulcão Neto, R. (2006). *Um processo de software e um modelo ontológico para apoio ao desenvolvimento de aplicações sensíveis a contexto*. Tesis doctoral publicada, Instituto de Ciências Matemáticas e de Computação – ICMC-USP São Paulo, Brasil.
- ❖ Lozano Rosh, E. (2007). *Portal de colaboración con capacidades semánticas*. Tesis Doctoral publicada. Universidad de Sevilla.
- ❖ Noguera García, M. (2009). *Modelado y Análisis de Sistemas CSCW Siguiendo un Enfoque de Ingeniería Dirigida por Ontologías*. Tesis doctoral publicada. Universidad de Granada. Disponible en:

<http://digibug.ugr.es/bitstream/10481/2325/1/18014094.pdf> (visitada en Enero del 2011).

- ❖ Timm Hinz, Verlani. (2008). Algoritmos para Interoperabilidad entre Ontologías. Tesis presentada para la obtención del grado de Magister en Ciencias de la Computación, Universidade Católica de Pelotas, Brasil.

Otras

- ❖ Contreras, J., & Martínez, J.A., Tutorial Ontologías. Pág. 1-24. Disponible en: www.sedic.es/gt_normalizacion_tutorial_ontologias.pdf (visitada en Enero del 2011).
- ❖ Qaissi, Hicham.(2009). Lenguaje de Recuperación de Datos en la Web Semántica SPARQL. Curso de Doctorado N° 584, Universidad Politécnica de Madrid. Disponible en: <http://sinbad.dit.upm.es> (visitado en Diciembre del 2010).
- ❖ W3C World Wide Web Consortium, (Editores) Brickley, D., Guha, R. 2004. "RDF Vocabulary Description Language 1.0: RDF Schema". W3C Recommendation. Disponible en: <http://www.w3.org/TR/rdf-schema/> (visitado en Noviembre del 2010)

- ❖ W3C World Wide Web Consortium, (Editores) Dean, Mike, Schreiber, Guus. 2004. "OWL Web Ontology Language Reference". W3C Recommendation. Disponible en: <http://www.w3.org/TR/owl-ref/>(visitado en Noviembre del 2010)
- ❖ W3C World Wide Web Consortium, (Editores) McGuinness, Deborah, Harmelen, Frank. 2004. "OWL Web Ontology Language Overview". W3C Recommendation. Disponible en: <http://www.w3.org/TR/owl-features/> (visitado en Noviembre del 2010)
- ❖ W3C World Wide Web Consortium, (Editores) Manola, F., Miller, E. 2004. "RDF Primer". W3C Recommendation. Disponible en: <http://www.w3.org/TR/rdf-primer/> (visitado en Noviembre del 2010).
- ❖ W3C World Wide Web Consortium, (Editores) Smith, Michael, Welty, Chris, McGuinness, Deborah. 2004. "OWL Web Ontology Language Guide". W3C Recommendation. Disponible en: <http://www.w3.org/TR/owl-guide/> (visitado en Noviembre del 2010)
- ❖ W3C World Wide Web Consortium. (2007). "SPARQL Query Language for RDF." Retrieved 10/08/2007, from <http://www.w3.org/TR/rdf-sparqlquery/#basicpatterns> (visitado en Noviembre del 2010)
- ❖ W3C World Wide Web Consortium. (2007). "SPARQL Protocol for RDF." Retrieved 10/08/2007, from <http://www.w3.org/TR/rdf-sparqlprotocol/> (visitado en Noviembre del 2010).

- ❖ W3C World Wide Web Consortium. (2007). "SPARQL Query Results XML Format." Retrieved 10/08/2007, from <http://www.w3.org/TR/rdfsparql-XMLres/>. (visitado en Noviembre del 2010)

ANEXOS

ANEXO A

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN - TACNA

FACULTAD DE CIENCIAS

ENCUESTA

Objetivo: Medir el grado de satisfacción de los docentes de la Facultad respecto a la eficiencia y usabilidad del sistema de administración académico de la Facultad de Ciencias.

Señor Docente, sírvase completar la siguiente encuesta que es para un trabajo de investigación solamente.

1. Considera usted qué el tiempo de respuesta después de realizar la solicitud de Información académica es:

a) Muy Malo

b) Malo

c) Regular

d) Bueno

e) Muy bueno

2. El Tiempo total para realizar el proceso de acceso a información académica es:

a) Muy alto

b) Alto

c) Regular

d) Bajo

e) Muy bajo

3. ¿Considera usted que la Cantidad de documentos y/o formularios necesarios para el acceso a la información académica es?

a) Demasiada

b) Mucha

c) Regular

d) Poca

e) Muy Poca

4. El Tiempo de tramitación de los documentos para el acceso a la información académica es:

a) Muy largo

b) Largo

c) Regular

d) Corto

e) Muy corto

5. ¿Comprende todos los pasos para acceder a la información académica?

a) Ninguno

b) Muy pocos

c) Algunos

d) Muchos

e) Todos

6. ¿Considera usted que el nivel de comprensión del proceso de administración académica es?

a) Muy difícil

b) Difícil

c) Regular

d) Fácil

e) Muy fácil

7. ¿Considera usted que el proceso de aprendizaje del sistema es?

a) Muy difícil

b) Difícil

c) Regular

d) Fácil

e) Muy fácil

8. De todas las peticiones de información académica realizadas,
¿Cuántas se cumplen en la fecha solicitada?

a) Ninguna

b) Muy pocas

c) Regular cantidad

d) Suficientes

e) Todas

9. De todas las peticiones de información académica, ¿Cuántas se
cumplen completamente (es decir, recibe lo que pide)?

a) Ninguna

b) Muy pocas

c) Regular cantidad

d) Suficientes

e) Todas

10. ¿El horario de atención para tener acceso a la información académica es?

a) Muy restrictivo

b) Restrictivo

c) Regular

d) Accesible

e) Muy accesible

ANEXO B

VALIDACIÓN DEL INSTRUMENTO

La validación de los instrumentos se llevó a cabo mediante dos procedimientos complementarios: una evaluación cualitativa, y la determinación de la consistencia interna. La validación cualitativa fue concretada a través de la consulta a expertos: otros docentes especialistas actuaron como jueces externos que juzgaron críticamente los enunciados de la encuesta permitiendo realizar los ajustes necesarios. La consistencia interna se determinó de acuerdo con la bibliografía referenciada en el marco teórico.

Se procedió a consultar con 10 expertos acerca de la congruencia, consistencia y redacción de las preguntas de la encuesta, obteniéndose los siguientes resultados.

Cuadro 13. Puntajes y promedio de los expertos

EXPERTO	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	PROM
1	4	3	4	3	5	2	3	4	3	3	3,4
2	3	3	3	4	4	5	2	1	4	3	3,2

Continúa ...

3	3	3	2	3	4	2	3	4	5	3	3,2
4	4	3	4	5	4	4	3	3	2	3	3,5
5	4	3	4	4	3	3	2	3	4	5	3,5
6	4	4	3	4	3	4	3	4	3	4	3,6
7	5	3	2	3	4	3	2	5	4	3	2,6
8	5	4	3	4	5	4	3	4	4	4	4
9	5	5	4	4	3	4	4	5	5	4	4,3
10	5	5	4	4	3	4	3	5	4	3	4
PROMEDIO	3,9	3,6	3,3	3,8	3,7	3,5	2,8	3,5	3,7	3,5	

Fuente: Elaboración propia

Si el promedio de valoración de las 10 preguntas por parte del experto es mayor a 3 entonces decimos que el experto está de acuerdo con la consistencia del instrumento, de lo contrario en desacuerdo:

Cuadro 14: Resumen y calculo de X^2

Frecuencias	De Acuerdo	En Desacuerdo
Observada f_o	9	1
Esperada f_e	5	5
$(f_o - f_e)$	4	-4
$(f_o - f_e)^2$	16	16
$(f_o - f_e)^2 / f_e$	16/5	16/5

Fuente: Elaboración propia

$$x_c^2 = \sum \frac{(f_o - f_e)^2}{f_e} = 6,4$$

Confiabilidad: $C = \frac{9}{10} * 100 = 90\%$

X^2 de tabla al 0,05 de significancia y 1 grado de libertad= 3,841

H_0 : No hay relación entre la encuesta y la opción de los expertos.

H_A : Hay relación entre la encuesta y la opción de los expertos.

Por lo tanto como $X_{calculado} > X_{esperado}$, cae en la zona de rechazo lo que significa que se rechaza la hipótesis nula y se acepta la hipótesis alternativa, es decir existe relación entre la encuesta y la opción de los expertos.

ANEXO C

ORGANIZACIÓN DE LA FACULTAD DE CIENCIAS

Entre las funciones que el estatuto universitario asigna a las facultades tenemos:

La elección de sus representantes entre sus miembros;

- La elaboración de sus reglamentos y de su Plan de Desarrollo y Funcionamiento;
- La elaboración y ejecución de su presupuesto anual;
- La elaboración de los planes de Estudio de las Escuelas Académico Profesionales y Secciones de Segunda Especialidad Profesional.
- La planificación, ejecución y evaluación de la enseñanza, la investigación, la extensión y proyección universitaria, la producción de bienes y la prestación de servicios;
- La organización y ejecución de los servicios de apoyo académico, técnico, administrativo y de otra índole para el logro de los objetivos de las Escuelas Académico Profesionales y Secciones de Segunda Especialidad Profesional.

- La organización del sistema de matrícula, tutoría y consejería para los estudiantes;
- La capacitación, perfeccionamiento y evaluación permanente de sus profesores y de su personal administrativo y de los servicios;
- La organización de las prácticas pre-profesionales y la coordinación pertinente con entidades públicas y/o privadas.
- La determinación de las equivalencias de estudio y certificados obtenidos en otras universidades en las áreas de su competencia y dar las normas para convalidación de cursos, traslados internos y externos.
- La evaluación del rendimiento académico de los estudiantes;
- La vigilancia de la buena formación profesional y científica del estudiante;
- Mantener estrecha relación con sus graduados con fines de capacitación y actualización profesional;
- Responsabilizarse por la seguridad, limpieza, mantenimiento e integridad de los bienes, instalaciones y locales asignados a la facultad;

Como se menciona estas organizan y dan operatividad a los servicios de apoyo académico, técnico, administrativo y de otra índole.

Entre los servicios brindados por la facultad encontramos la prestación o acceso a equipo informático; servicio que es materia de análisis en el presente estudio.

Ubicación geográfica

El proyecto ha de ejecutarse en el siguiente ámbito geográfico:

- Región : Tacna
- Provincia : Tacna
- Distrito : Tacna
- Lugar : Fundo “Los Granados” (Ciudad Universitaria)

El fundo “Los Granados” se halla situado a escasos 5 minutos del centro de la ciudad, enclavado entre la avenida Cusco (vía arterial de la ciudad que comunica al distrito de Tacna con el distrito Coronel Gregorio Albarracín Lanchipa) y las avenidas Jorge Basadre (ex Circunvalación), Miraflores y la calle N° 28. Ocupa a la fecha una extensión aproximada de 181 874,76 m² (esto es 18,19 ha de terreno) con un perímetro total de 1 853,23 m.

El acceso a la misma es masivo a través de las distintas rutas de transporte público que recorren la ciudad (Rutas: 2B, 10B, 11, 13, 14, 15,

90, entre otras) las que movilizan a los 6 050 estudiantes matriculados en las 30 carreras o especialidades que imparte la universidad a la fecha (2006; Boletín Estadístico N° 016 - Oficina de Planificación UNJBG); cabe citar sin embargo que la facultad de Ciencias Agrícolas y sus tres carreras profesionales (Agronomía, Economía Agraria y Veterinaria Zootecnia) disponen de un ambiente propio sito contiguo al fundo Los Granados (al que se denomina Fundo Los Pichones, también de propiedad de la entidad universitaria); del mismo modo otras unidades académicas de la Universidad como ser la Escuela de Pos Grado, Centro de Idiomas e Instituto de Telecomunicaciones y Sistemas desarrollan su accionar en local ubicado en la esquina de las avenidas Pinto y Bolognesi en el denominado "Local Central" de la UNJBG.

Para observar con mayor detalle la ubicación del ya mencionado Fundo "Los Granados" apreciamos a continuación mapa de macro y micro localización.

Descripción de la situación actual

Antecedentes

La Universidad Nacional de Tacna fue creada por decreto ley N° 1894, del 26 de agosto de 1971, en los considerandos de creación de los recursos naturales y la promoción social y económica de la región sur del país. Inicio sus actividades académicas el 13 de mayo de 1972 con una sesión solemne de apertura del año académico en el salón sesiones del ex – palacio municipal, con la presencia de distinguidas personalidades civiles, militares, eclesiásticas, en la que destacamos a quien luego fue nuestro patrono, el Dr. Jorge Basadre Grohmann.

El 11 de julio de 1980, se expidió la resolución N° 3058 – 80 – UNTAC denominando a la Institución como: “Universidad Nacional de Tacna Jorge Basadre Grohmann” en reconocimiento al Dr. Jorge Basadre Grohmann por sus virtudes personales, a su consagración total a la causa del Perú a su prolífico labor como estudioso de la república, escritor, ensayista y como hijo ilustre de Tacna. Cuando se promulga la ley universitaria N° 23733, el 09 de diciembre de 1983 por el presidente de la república Arq. Fernando Belaúnde Terry, oficialmente tomo la denominación de Universidad Nacional Jorge Basadre Grohmann.

El histórico local central, ubicado entre la avenida Bolognesi y la calle Pinto, cobijo a los primeros ingresantes y pasarían 27 años para que el 07 de noviembre de 1998 se transfiera definitivamente y a título gratuito este local a la universidad.

Actualmente la Universidad Nacional Jorge Basadre Grohmann "UNJBG" cuenta con dos (2) sedes donde se imparte la enseñanza profesional una en la ciudad universitaria (los granados y los pichones) ubicada en la Av. Miraflores s/n. y la otra sede en el local central ubicada en la Av. Pinto con Bolognesi, la distancia que separa una de otra es aproximadamente un (1) km. La Facultad de Ciencias "FACI" cuenta con una infraestructura construida en el año 1971 bajo la gestión de las autoridades universitarias de ese entonces y los estudiantes, y que se localizada en la ciudad universitaria en los granados.

Actualmente la Facultad de Ciencias "FACI" cuenta con cuatro (4) Escuelas Académico Profesional "EAP", las cuales son: Biología-microbiología, física aplicada, ingeniería química e ingeniería informática y de sistemas.

Según el documento generado en el año 2007 denominado “diagnóstico de la Universidad Nacional Jorge Basadre Grohmann – Tacna” se menciona que se tiene como misión: “Formar profesionales, investigadores y post – graduados creativos, competentes y de excelencia, comprometidos con la solución de los problemas de la comunidad; genera y desarrolla conocimientos científicos, tecnológicos y humanísticos, útiles para desarrollar alternativas y soluciones integrales, que permitan engrandecer la cultura, promover la justicia, la equidad social y la igualdad de oportunidades en nuestro país”.

Para el caso del presente diagnóstico de la facultad de Ciencias “FACI” tomaremos como eje de análisis el problema del desarrollo de la formación académica e investigación con respecto al acceso a equipos informáticos en la facultad de Ciencias “FACI” que es trascendental en pleno siglo XXI donde toda información está al alcance de los usuarios siempre en cuando se cuente con la tecnología apropiada.

ANEXO D

DESARROLLO DE APLICACIÓN

A continuación presentaremos la secuencia (descrita en nueve pasos) del desarrollo del prototipo de sistema de información ontológico usado en la presente tesis:

PRIMERO: Se revisó la documentación relacionada a la Facultad de Ciencias, para poder empaparnos totalmente de los conceptos relacionados con nuestro dominio de aplicación:

SEGUNDO: Se modeló la Ontología usando Protégé 3.4.1

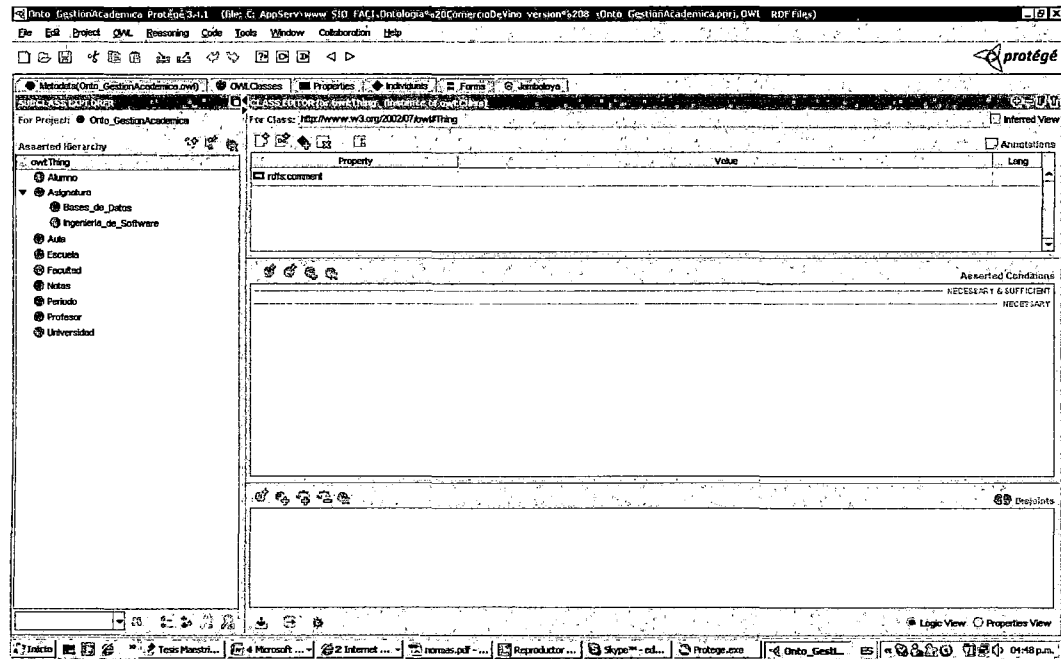


Figura 33: Modelo de clases de la ontología "Onto_GestiónAcadémica" en Protégé

Fuente : Elaboración propia.

TERCERO: Se definieron las propiedades y los tipos de datos.

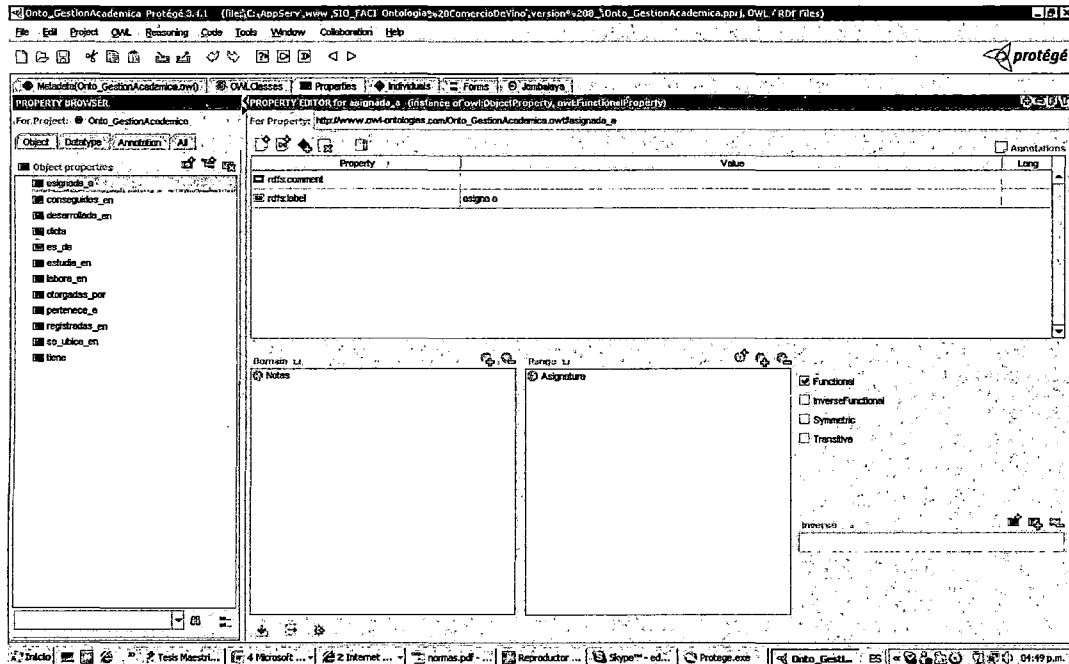


Figura 34: Objetos de propiedades de la ontología "Onto_GestiónAcadémica"

Fuente : Elaboración propia.

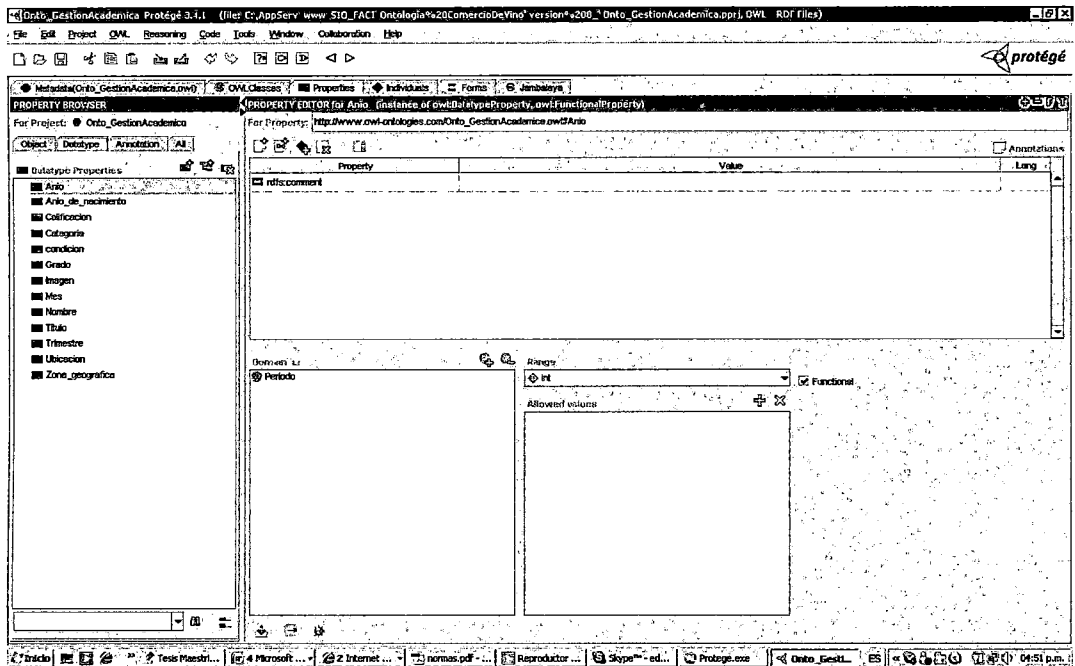


Figura 35 : Definición de los tipos de datos de la ontología
 “Onto_GestiónAcadémica”

Fuente : Elaboración propia.

QUINTO: Definición de las instancias de las clases para poder poblar nuestra ontología.

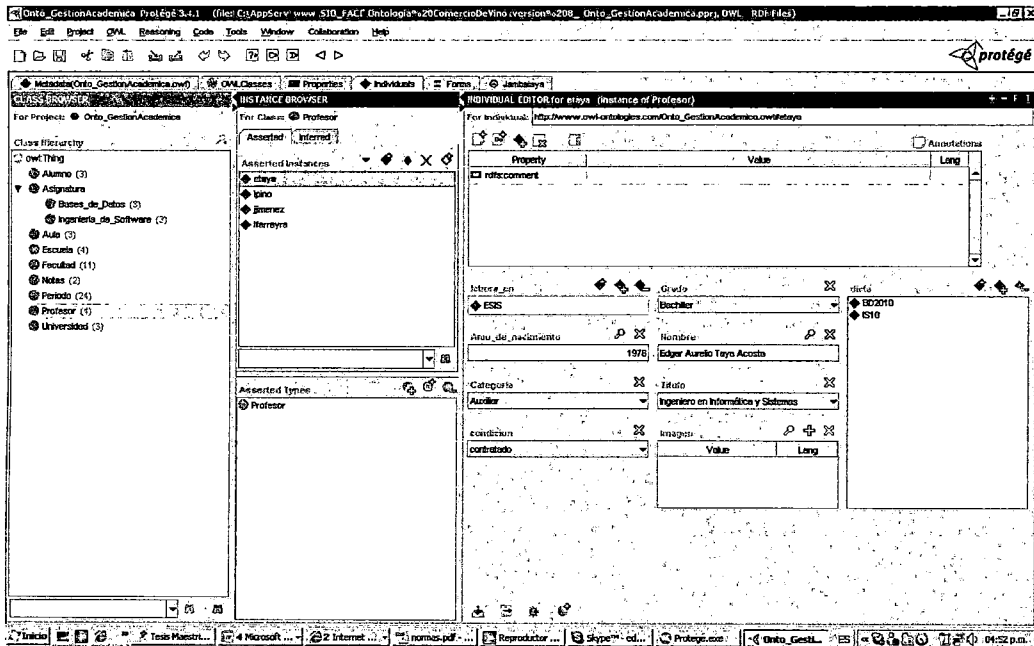


Figura 36: Instancia de Profesor con sus propiedades y tipos de datos de la ontología “Onto_GestiónAcadémica”

Fuente : Elaboración propia.

SEXTO: Luego se desplegó toda la ontología gráficamente

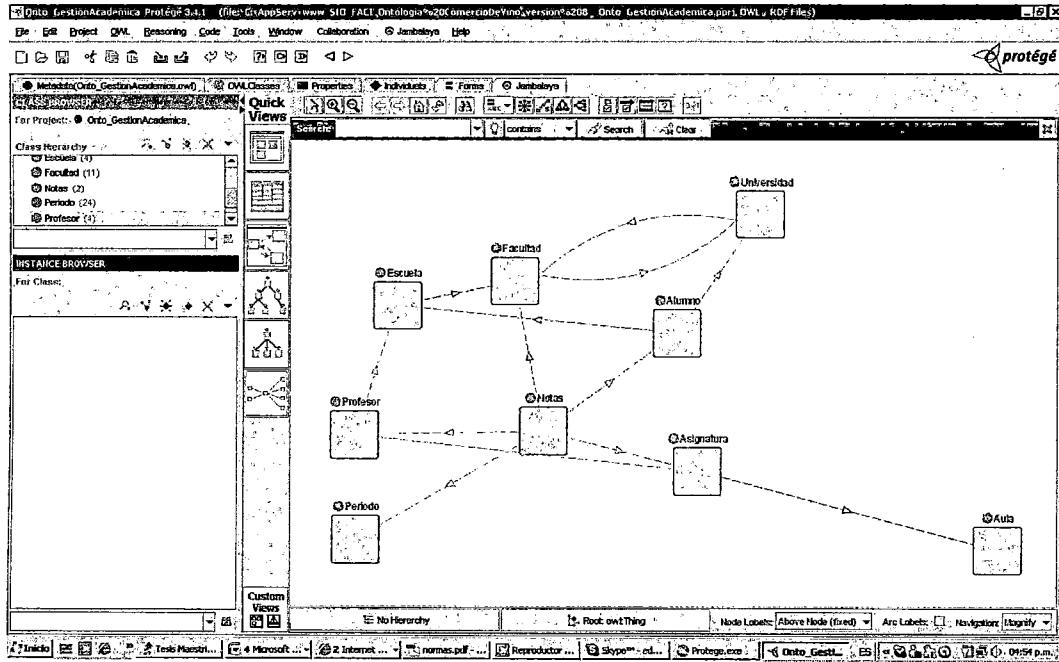


Figura 37 : Visualización de la ontología “Onto_GestiónAcadémica”
utilizando Jambalaya de Protégé

Fuente: Elaboración propia.

SEPTIMO: Se generó el código OWL correspondiente de la Ontología “Onto_GestiónAcadémica”

```
<owl:FunctionalProperty rdf:ID="labora_en">
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >labora en</rdfs:label>
  <rdfs:domain rdf:resource="#Profesor"/>
  <rdfs:range rdf:resource="#Escuela"/>
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="otorgadas_por">
  <rdf:type
rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  <rdfs:domain rdf:resource="#Notas"/>
  <rdfs:range rdf:resource="#Profesor"/>
  <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >otorgadas por</rdfs:label>
</owl:FunctionalProperty>
```

Figura 38 : Segmento de código en OWL de la ontología

“Onto_GestiónAcadémica” generado por Protégé.

Fuente : Elaboración propia

OCTAVO: Utilizamos RAP (RDF API for PHP) para poder trabajar con el archivo OWL y de esta manera generar la BD de la Ontología en MySQL

id	subject	predicate	object
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/2000/01/rdf-schema#label	universidad
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/2000/01/rdf-schema#label	bases de datos
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/2002/07/owl#disjointWith	http://www.owl-ontologies.com/Onto_GestionAcademica
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/2000/01/rdf-schema#subClassOf	http://www.owl-ontologies.com/Onto_GestionAcademica
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.owl-ontologies.com/Onto_GestionAcademica	teacher
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/2000/01/rdf-schema#comment	
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/2000/01/rdf-schema#label	profesor
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/2000/01/rdf-schema#label	aula
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Class
1	http://www.owl-ontologies.com/Onto_GestionAcademica	http://www.w3.org/2000/01/rdf-schema#label	periodos

Figura 39 : Registros de la tabla statements de la base de datos

onto_gestionacademica creada con RAP en MySQL

Fuente : Elaboración propia

NOVENO: Se construyó el prototipo de la aplicación utilizando PHP y MySQL, con algunas funcionalidades como la búsqueda:

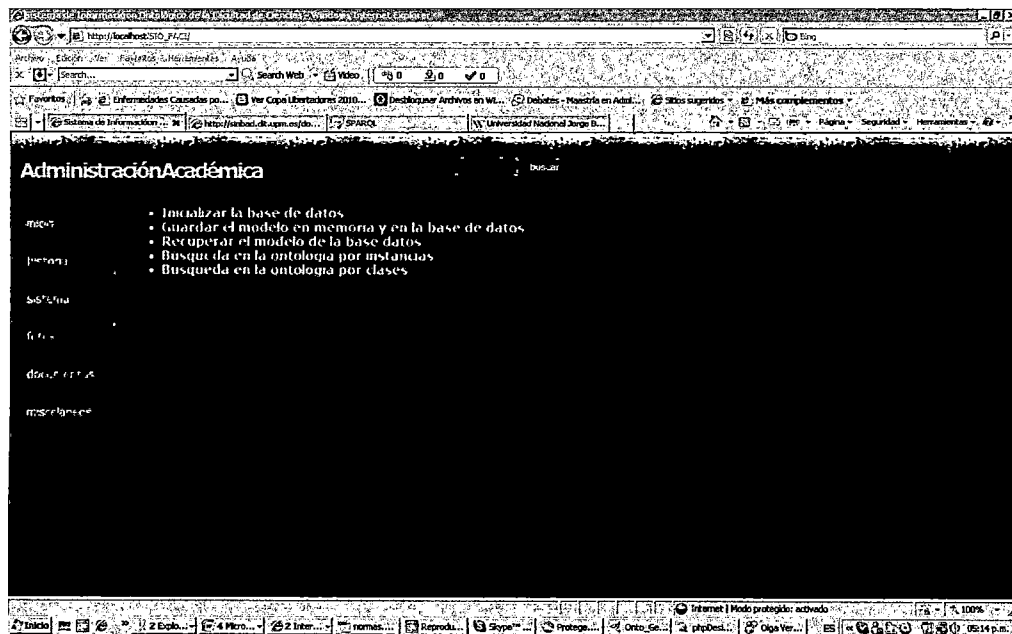


Figura 40 : Interfaz del prototipo del sistema de información ontológico.

Fuente : Elaboración propia

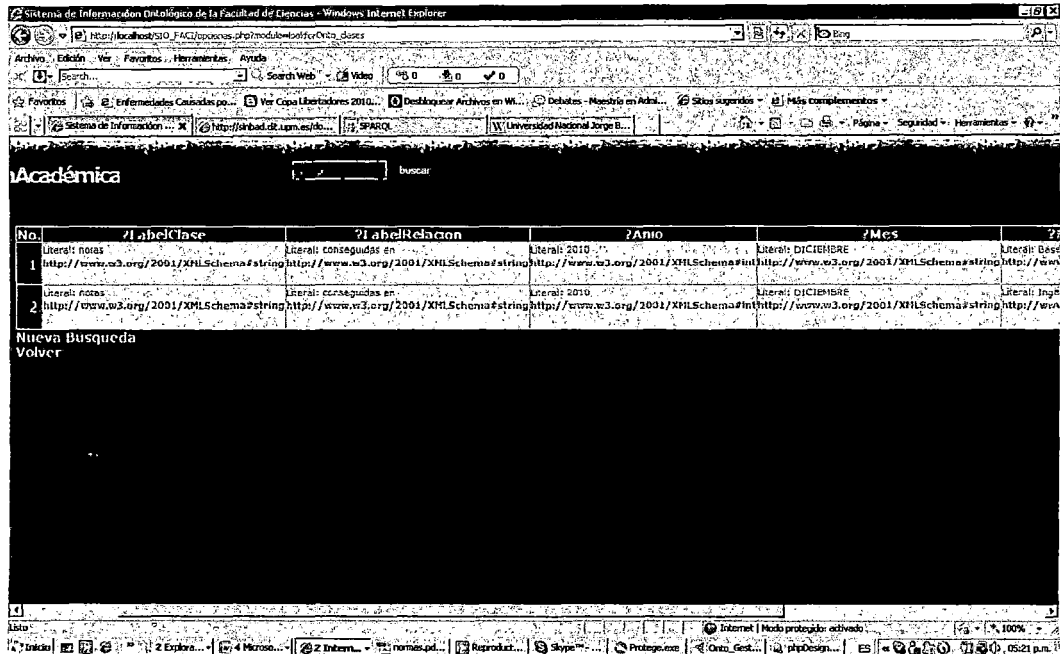


Figura 41 : Búsqueda por clases de notas

Fuente : Elaboración propia.

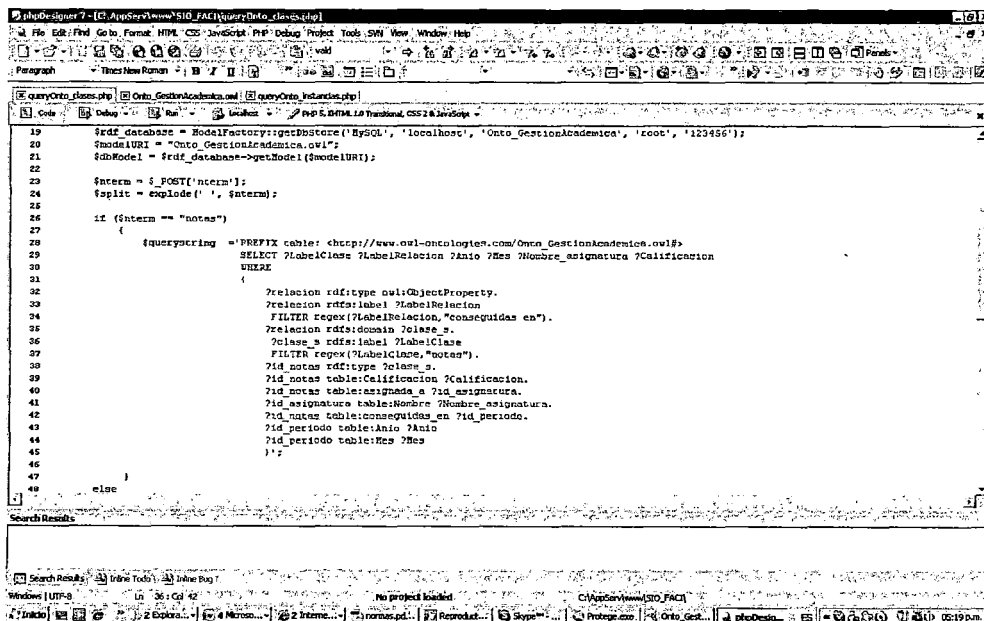


Figura 42 : Código fuente en PHP y SPARQL de la búsqueda por clases.

Fuente : Elaboración propia