



*Universidad Carlos III de Madrid
Facultad de Humanidades, Comunicación y Documentación
Departamento de Biblioteconomía y Documentación*

**Doctorado en Documentación
Archivos y Bibliotecas en el Entorno Digital**

**Sistema de Recuperación Conceptual
mediante Niveles Semánticos en la
representación de Esquemas de
Metadatos**

Tesis Doctoral

Autor
Vicente Palacios Madrid

Director
**Prof. Dr. D. José Antonio Moreiro González
Prof. Dr. D. Jorge Morato Lara**

Getafe, junio de 2010

Resumen

Los metadatos son descripciones que facilitan la recuperación, utilización y gestión de recursos de información. Así, pueden ser utilizados para organizar recursos electrónicos heterogéneos o facilitar la interoperatividad. Normalmente, la semántica de los elementos de vocabularios se define localmente, con escasa formalización y sin contemplar definiciones consensuadas con otros vocabularios, con el consiguiente perjuicio para la interoperatividad. En esta tesis se propone generar una representación semántica de los vocabularios de metadatos, que permita eliminar la ambigüedad sintáctica y semántica, facilitando así la interoperatividad. La representación propuesta posee dos vertientes: la primera, denominada *esquema cualificado*, tiene como objetivo la sustitución del esquema original y proporciona una estructura homogénea para todos los esquemas, al tiempo que permite incluir la semántica de cada uno de sus elementos; la segunda, denominada *ontología específica*, permite establecer definiciones formales de los elementos incluidos en el esquema original, al tiempo que proporciona soporte a aspectos como la sinonimia y el plurilingüismo. La representación semántica de los esquemas se completa con el uso de un recurso semántico, contra el cual se interrelacionarán los conceptos de las ontologías específicas mediante un proceso de alineamiento, articulado a través de una ontología independiente. De este modo se facilitará la interoperatividad entre esquemas, la recuperación conceptual de documentos y esquemas, así como el uso más amigable de los vocabularios. El desarrollo de la propuesta incluye la definición del modo de generar las representaciones semánticas de los esquemas, la definición de los recursos necesarios y la metodología generada a partir de la aplicación del método sobre un conjunto de esquemas y recursos semánticos seleccionados como ejemplo. Además, se propone una metodología de documentación y cualificación de esquemas, desarrollada a partir de la experiencia con los esquemas seleccionados. La metodología incluye el procedimiento y plantillas documentales, para la creación de un documento de descripción del esquema, completado con dos vistas: una dirigida a usuarios y otra en RDF, procesable automáticamente. Seguidamente, se establecen los requisitos para la selección de la ontología de referencia y, tras un proceso de evaluación de recursos existentes, se selecciona uno a modo de ejemplo. A continuación se establece el modo de interrelacionar los conceptos de los distintos componentes, ontología específica y ontología de referencia, a partir de la evaluación de los métodos existentes. Establecido el método, y mediante un método inductivo, se genera una metodología de interrelación de conceptos, a partir del proceso de alineamiento entre el conjunto de esquemas seleccionados y la ontología de referencia. Para finalizar el planteamiento, éste será validado y verificado con el objetivo de comprobar que responde cubre los objetivos establecidos de forma correcta. La solución complementa y es compatible con propuestas anteriores de definición y gestión de metadatos como los registros de metadatos o las propuestas del DCMI. Además, contempla criterios de modularidad, extensibilidad, refinamiento y plurilingüismo. Una vez establecida y probada la propuesta, se incluye la especificación del Proceso de Desarrollo de un sistema informático que la soporte. Con el fin de evaluar el planteamiento, se realiza una valoración y comparación del método propuesto, con otros métodos de recuperación alternativos, siguiendo la metodología DESMET. Además, se realiza un análisis y valoración de los resultados generados. Finalmente, se incluyen las conclusiones obtenidas durante la elaboración de la propuesta y las posibles líneas de investigación que quedan abiertas para futuros trabajos.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas aquellas personas que, directa o indirectamente, han hecho posible que hoy esté escribiendo estas líneas.

En primer lugar, agradezco a los profesores Dr. D. Jose A. Moreiro y Dr. D. Jorge Morato su confianza, ayuda, apoyo y amistad en los momentos buenos y menos buenos.

En segundo lugar, a mis compañeros, tanto del Servicio de Informática como del Departamento de Informática, sin su ayuda, comprensión y apoyo no habría sido posible la elaboración de esta tesis. En especial a la Prof^a. Dra. D^a. Sonia Sanchez y al Prof. Dr. D. Juan B. Llorens como compañeros y amigos.

Finalmente y con todo mi cariño, agradecer a mi familia su paciencia, comprensión y apoyo durante todos estos años.

Tabla de Contenidos

CAPÍTULO I: INTRODUCCIÓN	1
1.1 Motivación.....	3
1.2 Hipótesis	5
1.3 Objeto.....	5
1.4 Objetivos	5
1.5 Metodología.....	7
1.6 Fuentes	15
CAPÍTULO II: FUNDAMENTOS TEÓRICOS	17
2.1 La Web Semántica	17
2.1.1 Introducción.....	17
2.1.2 Arquitectura.....	18
2.1.3 Lenguajes.....	21
2.1.4 Situación actual de la Web Semántica	28
2.2 Metadatos	31
2.2.1 Esquemas de metadatos.....	31
2.2.2 Esquemas de metadatos seleccionados.....	36
2.2.3 Interoperatividad	39
2.2.4 Correspondencia entre Esquemas	40
2.3 Ontologías	44
2.3.1 Introducción.....	44
2.3.2 Componentes de una ontología	46
2.3.3 Áreas de aplicación	47
2.3.4 Ontologías de acceso libre	48
2.3.5 Repositorios semánticos	50
2.3.6 Métodos de correspondencia entre ontologías	56
2.4 Recuperación de documentos XML	62
2.5 Marco teórico y herramientas del Proceso de Desarrollo	64
2.5.1 Fundamentos.....	64
2.5.2 Actividades de gestión del Proceso Software.....	65
2.5.3 Ciclo de vida del software	66
2.5.4 El Lenguaje de Modelado Unificado: UML	70
2.5.5 Estilos arquitectónicos y patrones de diseño	79
CAPÍTULO III: MARCO EXPERIMENTAL	85
3.1 Representación ontológica de esquemas	87
3.1.1 Definición de la representación semántica de esquemas.....	88
3.1.2 Búsqueda y selección de esquemas	109
3.1.3 Representación semántica de los esquemas seleccionados.....	111
3.1.4 Publicación de esquemas semánticos.....	120

3.2 Selección de la ontología de referencia	122
3.2.1 Definición de requisitos de la ontología de referencia	123
3.2.2 Evaluación y selección de la ontología de referencia	124
3.2.3 Representación compatible de la ontología de referencia	134
3.2.4 Publicación de la ontología de referencia	138
3.3 Correspondencia entre esquemas y la ontología de referencia	139
3.3.1 Evaluación y selección del método de correspondencia.....	140
3.3.2 Correspondencia de conceptos entre esquemas y la ontología de referencia	145
3.3.3 Publicación de la ontología de correspondencia	158
3.4 Validación y verificación de la propuesta	159
3.5 Proceso de desarrollo del sistema informático de soporte	168
3.5.1 Especificaciones informales.....	169
3.5.2 Especificación de requisitos de usuario	173
3.5.3 Especificación de requisitos software.....	197
3.5.4 Diseño de la arquitectura	197
3.5.5 Diseño detallado, codificación y pruebas.....	198
3.5.6 Desarrollo de infraestructuras de soporte	199
3.5.7 Estado actual de la plataforma	200
CAPÍTULO IV: RESULTADOS, EVALUACIÓN Y DISCUSIÓN	201
4.1 Resultados y discusión	201
4.1.1 Resultados de la representación ontológica de esquemas.....	202
4.1.2 Resultados de la selección de la ontología de referencia.....	212
4.1.3 Resultados de la correspondencia entre esquemas y la ontología de referencia	214
4.1.4 Resultados de la validación y verificación de la propuesta	219
4.1.5 Resultados del proceso de desarrollo del sistema informático de soporte	221
4.2 Evaluación y comparativa de la propuesta	224
4.2.1 Presentación del método de evaluación.....	225
4.2.2 Definición del marco comparativo.....	226
4.2.3 Evaluación de métodos de recuperación de documentos XML.....	230
4.2.4 Presentación, análisis y discusión de resultados.....	233
CAPÍTULO V: CONCLUSIONES	239
CAPÍTULO VI: TRABAJOS FUTUROS	243
CAPÍTULO VII: BIBLIOGRAFÍA	245
CAPÍTULO VIII: GLOSARIO	271
ANEXO A: PLANTILLA DE DESCRIPCIÓN DE ESQUEMAS DE METADATOS	277
ANEXO B: RESULTADOS DE LA EVALUACIÓN DE REPOSITORIOS SEMÁNTICOS	281

Índice de Figuras

Figura I-1: Principales pasos de la metodología propuesta	7
Figura I-2: Tareas de la fase de representación ontológica de esquemas	8
Figura I-3: Tareas de la fase de selección de la ontología de referencia	9
Figura I-4: Tareas de la fase de correspondencia entre esquemas y la ontología de referencia	10
Figura I-5: Tareas del proceso de desarrollo del sistema informático de soporte...11	
Figura II-1: Estructura y lenguajes de la Web Semántica.....	18
Figura II-2: Estructura y estándares de la Web Semántica. Extraído de (Sure, Y.; Studer, R., 2005).....	20
Figura II-3: Representación gráfica de una sentencia RDF. Extraído de (World Wide Web Consortium, 2004d).	24
Figura II-4: Esquema de anotación semántica en KIM extraído de (Ontotext, 2010b).....	52
Figura II-5: Función de alineamiento, tomada de (Euzenat, J.; Shvaiko, P., 2007) ...	59
Figura II-6: Ciclo de vida de los alineamientos, tomado de (Euzenat, J.; Shvaiko, P., 2007)	60
Figura II-7: Representación un proceso de desarrollo software.....	65
Figura II-8: Ciclo de vida en cascada	67
Figura II-9: Ciclo de vida en espiral	69
Figura II-10: Diagrama ejemplo de casos de uso.....	72
Figura II-11. Representación gráfica de un paquete	73
Figura II-12. Representación gráfica de clase.....	73
Figura II-13. Representación gráfica de una interfaz junto a la clase que la realiza	73
Figura II-14. Ejemplo de diagrama de clases	74
Figura II-15. Ejemplo de diagrama de estado	75
Figura II-16. Ejemplo de diagrama de actividades.	75
Figura II-17. Ejemplo de diagrama de secuencia.....	76
Figura II-18. Ejemplo de diagrama de colaboración	77

Figura II-19. Ejemplo de diagrama de componentes	78
Figura II-20. Ejemplo de diagrama de despliegue.....	78
Figura II-21: Patrón arquitectónico MVC modelo 2	79
Figura II-22: Combinación de los estilos arquitectónicos Capas y MVC.....	81
Figura II-23: Estructura del patrón de diseño Value Object	82
Figura II-24: Estructura del patrón de diseño Factory Method.....	83
Figura II-25: Estructura del patrón de diseño Facade.....	83
Figura III-1: Principales pasos de la metodología propuesta	87
Figura III-2: Tareas de la fase de representación ontológica de esquemas	88
Figura III-3: Ejemplo de representación semántica del elemento 'rights' definido en el esquema de metadatos de Dublin Core	109
Figura III-4: Tareas de la fase de selección de la ontología de referencia	123
Figura III-5: Tareas de la fase de correspondencia entre esquemas y la ontología de referencia.....	140
Figura III-6:Diagrama de actividad del proceso de correspondencia de un concepto.	147
Figura III-7:Diagrama de actividad del tratamiento de casos específicos en la correspondencia de propiedades.	153
Figura III-8: Esquema de recuperación de documentos. Extraído de (Baeza, R.; Ribeiro, B., 1999).	160
Figura III-9: Definición de la precisión.	161
Figura III-10: Definición de la exhaustividad.....	161
Figura III-11: Tareas del proceso de desarrollo del sistema informático de soporte	169
Figura III-12: Diagrama inicial de Casos de Uso	171
Figura III-13: Representación gráfica de la jerarquía de usuarios	173
Figura III-14: Diagrama general de Casos de Uso.	174
Figura III-15: Diagrama de casos de uso del rol usuario	175
Figura III-16: Diagrama de gestión de documentos	181
Figura III-17: Diagrama de gestión de perfiles de aplicación.....	181

Figura III-18: Diagrama de casos de uso del rol Experto Responsable del Dominio	184
Figura III-19: Diagrama de casos de uso del rol Experto del Dominio.....	185
Figura III-20: Diagrama gestión de ontologías de referencia	187
Figura III-21: Diagrama de casos de uso del rol Ingeniero Responsable del Dominio	192
Figura III-22: Diagrama de gestión de perfiles.....	195
Figura III-23: Diagrama de gestión de usuarios.....	196
Figura III-24: Arquitectura de la aplicación y tecnologías utilizadas	198
Figura IV-1: Estructura de la exposición de la propuesta y de los resultados obtenidos.....	201
Figura IV-2: Directrices de los logros conseguidos.	202
Figura IV-3: Resumen de los logros conseguidos en la fase de representación ontológica de esquemas.	202
Figura IV-4: Detalles del resultado I: Esquema de Cualificadores Semánticos.....	203
Figura IV-5: Detalles del resultado II: Modo de cualificación de esquemas.....	203
Figura IV-6: Detalles del resultado III: Representación ontológica de la semántica.	205
Figura IV-7: Detalles del resultado IV: Recomendaciones para la cualificación de esquemas.....	207
Figura IV-8: Detalles del resultado V: Plantilla de descripción de esquemas de metadatos.	207
Figura IV-9: Detalles del resultado VI: Cualificación del conjunto de esquemas seleccionado.....	208
Figura IV-10: Detalles del resultado VII: Recomendaciones para la generación de ontologías específicas.....	209
Figura IV-11: Detalles del resultado VIII: Generación de las ontologías específicas de los esquemas seleccionados.	209
Figura IV-12: Detalles del resultado IX: Publicación de esquemas y ontologías en un servidor público.	211
Figura IV-13: Resumen de los logros conseguidos en la fase de selección de la ontología de referencia.....	212

Figura IV-14: Detalles del resultado X: Requisitos de la ontología de referencia..	212
Figura IV-15: Detalles del resultado XI: Representación compatible y extensible de la ontología de referencia.....	213
Figura IV-16: Detalles del resultado XII: Publicación de la representación de la ontología de referencia.....	213
Figura IV-17: Resumen de los logros conseguidos en la fase de correspondencia entre esquemas y la ontología de referencia.	214
Figura IV-18: Detalles del resultado XIII: Requisitos del método de correspondencia entre ontologías.	215
Figura IV-19: Detalles del resultado XIV: Definición del método de correspondencia.....	215
Figura IV-20: Detalles del resultado XV: Metodología de correspondencia entre ontologías.....	216
Figura IV-21: Detalles del resultado XVI: Correspondencia del conjunto de esquemas seleccionado con la ontología de referencia.	218
Figura IV-22: Detalles del resultado XVII: Publicación de los alineamientos generados en un servidor público.....	219
Figura IV-23: Resumen de los logros conseguidos en la fase de validación y verificación de la propuesta.....	219
Figura IV-24: Detalles del resultado XVIII: Ejemplificación, validación y verificación de la propuesta.	220
Figura IV-25: Resumen de los logros conseguidos en la fase de desarrollo del proyecto.....	222
Figura IV-26: Detalles del resultado XIX: Documentación del proyecto.	222
Figura IV-27: Detalles del resultado XX: Código del proyecto.....	223
Figura IV-28. Resultados de la evaluación de los métodos candidatos para la categoría Gestión de Esquemas.....	234
Figura IV-29. Resultados de la evaluación de los métodos candidatos para la categoría Gestión Semántica.	235
Figura IV-30. Resultados de la evaluación de los métodos candidatos para la categoría Consulta.	236
Figura IV-31. Comparativa final entre los métodos candidatos.	237

Índice de Tablas

Tabla II-1: Resumen del esquema de metadatos Dublin Core Element Set v1.1.....	37
Tabla II-2: Resumen del esquema de metadatos vCard.....	37
Tabla II-3: Resumen del esquema de metadatos FOAF.....	37
Tabla II-4: Resumen del esquema de metadatos DOAC.....	38
Tabla II-5: Resumen del esquema de metadatos DOAP.....	38
Tabla II-6: Resumen del esquema de metadatos PIM.....	39
Tabla II-7: Fases de un ciclo de vida.....	67
Tabla III-1: Metadatos para la definición de Esquema de Cualificadores Semánticos.....	92
Tabla III-2: Definición original del término ‘relation’ del esquema Dublin Core.....	92
Tabla III-3: Metadatos para la definición del cualificador semántico ‘tieneSemantica’.....	93
Tabla III-4: Ejemplo de cualificación semántica del elemento ‘rights’ del esquema Dublin Core.....	95
Tabla III-5: Metadatos para la definición de la ontología SEMSE.....	102
Tabla III-6: Metadatos para la definición de la ontología específica DCELEMENTSSO.....	107
Tabla III-7: Conjunto inicial de esquemas y valoración según los criterios de selección propuestos.....	111
Tabla III-8: Registro de estado del esquema Dublin Core Cualificado Semánticamente.....	113
Tabla III-9: Conjunto mínimo de atributos para la definición de un término.....	114
Tabla III-10: Definición del elemento ‘rightsSem’ del esquema Dublin Core Cualificado Semánticamente.....	114
Tabla III-11: Resumen de los esquemas cualificados generados.....	116
Tabla III-12: Resumen de las ontologías específicas generadas.....	120
Tabla III-13: Resumen de los resultados de las consultas sobre el término ‘Agent’ y valores de los indicadores de calidad.....	165

Tabla III-14: Resumen de los resultados de las consultas sobre el término 'title' y valores de los indicadores de calidad	166
Tabla III-15: Resumen de los resultados de las consultas sobre el término 'imagen' y valores de los indicadores de calidad	166
Tabla IV-1: Resumen de los esquemas cualificados generados	208
Tabla IV-2: Resumen de las ontologías específicas generadas.	211
Tabla IV-3: Importancia de cada característica y breve justificación.....	229
Tabla IV-4: Sistema de valoración de características simples.....	229
Tabla IV-5: Sistema de valoración de características compuestas	229
Tabla IV-6: Sistema de valoración de la importancia de las características	230
Tabla IV-7: Resumen de la valoración de los métodos candidatos.	233

Índice de Ejemplos

Ejemplo II-1: Representación textual de una sentencia RDF. Extraído de (World Wide Web Consortium, 2004d).....	24
Ejemplo III-1: Codificación del cualificador semántico ‘hasSemantics’ en RDF	97
Ejemplo III-2: Codificación completa del esquema de cualificadores semánticos en RDF	98
Ejemplo III-3: Codificación en RDF del término ‘rights’ de Dublin Core	99
Ejemplo III-4: Codificación de la clase SubjectConcept incluida en la ontología SEMSE.....	103
Ejemplo III-5: Codificación de la clase SubjectProperty incluida en la ontología SEMSE.....	104
Ejemplo III-6: Codificación de la clase Semset incluida en la ontología SEMSE.....	105
Ejemplo III-7: Codificación de las propiedades hasSemset y hasPropertySemset incluidas en la ontología SEMSE	105
Ejemplo III-8: Codificación del término dc:rights y sus semsets asociados.....	107
Ejemplo III-9: Extracto de codificación en RDF del esquema Dublin Core Cualificado Semánticamente	116
Ejemplo III-10: Extracto de codificación de la ontología específica correspondiente al esquema DOAP	119
Ejemplo III-11: Extracto de codificación de la ontología específica correspondiente al esquema PIM.....	119
Ejemplo III-12: Extracto del fichero de configuración .htaccess	121
Ejemplo III-13: Extracto de codificación de la ontología de aplicación correspondiente a la ontología de referencia PROTON	136
Ejemplo III-14: Extracto del fichero de configuración .htaccess	139
Ejemplo III-15: Extracto de la ontología de alineamiento interno correspondiente a la ontología de referencia PROTON	143
Ejemplo III-16: Extracto de la ontología SEMSE correspondiente a la codificación de la relación equivalentConcept	144
Ejemplo III-17: Extracto de la ontología de alineamiento correspondiente a la relación entre los conceptos foaf:Agent y proton:Agent	149

Ejemplo III-18: Extracto de la ontología de alineamiento, correspondiente a la relación entre las propiedades dcelements:title y dcelements:creator con proton:title y proton:statedBy	149
Ejemplo III-19: Extracto de la ontología de alineamiento, correspondiente a la relación entre la propiedad doap:maintainer y la clase proton:Maintainer	150
Ejemplo III-20: Extracto de la ontología de aplicación asociada a la ontología de referencia PROTON correspondiente a la inclusión de la clase Image	150
Ejemplo III-21: Extracto de la ontología de aplicación protona, correspondiente a la inclusión del concepto EmailReader	151
Ejemplo III-22: Extracto de la ontología de aplicación protona, correspondiente a la inclusión de la propiedad hasRelation	152
Ejemplo III-23: Extracto del fichero de configuración .htaccess	159

Capítulo I: Introducción

Originalmente, la Web se diseñó para su utilización por personas en vez de para aplicaciones. Este planteamiento ha limitado las posibilidades de las aplicaciones Web, tales como la recuperación de información, los sistemas pregunta-respuesta o el razonamiento mediante inferencias. Según Díaz (Díaz Ortuño, P.M., 2003) las limitaciones están relacionadas con: la tecnología utilizada, el número y calidad de los metadatos y el tamaño de la Web. La solución más adecuada consiste en transformar la información en un formato más comprensible para el software, con una mayor carga de semántica codificada. Para conseguir este objetivo es necesario crear una “capa semántica” que pueda ser “comprendida” por el software. Este es el objetivo de la Web Semántica, y para conseguirlo utiliza como elementos clave metadatos y la sintaxis del lenguaje XML.

Además de ser el corazón de la Web Semántica, el estándar XML se ha establecido en los últimos años como pieza clave de la interconexión de aplicaciones para comercio electrónico y para compartir recursos de información¹. Una de las características de los documentos XML que justifican su popularidad es su vinculación a determinadas DTDs y Esquemas.

No obstante, existen problemas en la recuperación e interpretación de los documentos, que pueden resumirse en:

- Dificultades en la interpretación semántica de sus elementos componentes (esquemas XML, DTD's y vocabularios de metadatos, en adelante *esquemas*).
- Problemas en la recuperación mediante motores de búsqueda Web.
- Ambigüedad, ya que la referencia a vocabularios de metadatos es opcional.
- Dificultades en la recuperación, al no existir un repositorio donde se almacenen de forma uniforme y conceptualmente relacionados, al menos aquellos esquemas con mayor popularidad.

Las consecuencias de estos problemas son la necesidad de que el usuario tenga conocimiento del esquema para poder utilizarlo y la falta de reutilización de esquemas, por desconocimiento del usuario. Todo ello redundará en el desarrollo de esquemas ad-hoc, con frecuencia escasamente formalizados, frente a lo que sería deseable, la reutilización de los ya existentes.

¹ Actualmente, mayo de 2010, una búsqueda en Google sobre el número de documentos XML muestra la existencia de, aproximadamente, 19.800.000 documentos, 32.200 DTDs y 103.000 Esquemas.

Para paliar estos problemas, este trabajo presenta una propuesta consistente en dotar a los esquemas de metadatos de un plano semántico superior, que permita la asignación de semántica y desambiguación de los términos, mediante una estructura conceptual flexible.

Con todo ello, a través de representaciones semánticas de los esquemas y el uso de ontologías de referencia, se consigue un mapa conceptual y relacional que incluye y permite equivalencias léxicas entre los términos de las etiquetas de los documentos y la estructura ontológica general.

Las principales ventajas de la propuesta parten del hecho de no presuponer ningún conocimiento de la estructura del documento XML por parte del usuario. Para lograrlo, es preciso desarrollar tanto la estructura como los recursos para la interconexión de elementos XML. La relación semántica entre conceptos se realizará gracias a una estructura de tipo ontológico. La propuesta incluye tanto el marco teórico, como la metodología para la cualificación e interrelación semántica de esquemas, todo ello soportado por ejemplos de aplicación. Finalmente, se incluye la especificación del sistema informático que soportará el proceso de interconexión y gestión, tanto de esquemas como de ontologías.

La propuesta ha sido validada mediante la publicación de diversos artículos nacionales e internacionales como: la *Fifth International Semantic Web Conference* (Palacios, V. et al., 2006a), la *Conferência Ibérica de Sistemas e Tecnologias de Informação* (Palacios, V. et al., 2006b) o la *International Conference on Dublin Core and Metadata Applications* (Palacios, V. et al., 2006c). También ha sido financiada parcialmente mediante la concesión de un proyecto PRICIT de la Comunidad Autónoma de Madrid titulado “*Sistema de Representación y Recuperación Semántica de Esquemas Basado en Ontologías Multinivel*” presentado en el año 2006. Además, desde octubre de 2007, se está desarrollando un proyecto, dentro del Plan Nacional de I+D+i, titulado “*Desarrollo de un sistema de Recuperación Conceptual mediante Niveles Semánticos en la representación de Esquemas de Metadatos*”, basado en el planteamiento propuesto en esta tesis.

El resto de este trabajo se estructura del siguiente modo:

En el *Capítulo I: Introducción*, se presenta la motivación, objetivos, hipótesis y metodología de la propuesta.

El siguiente apartado, *Capítulo II: Estado del Arte*, se da una visión del estado actual de las distintas teorías, herramientas y tecnologías relacionadas con la propuesta.

En el *Capítulo III: Marco Experimental*, se muestra el proceso de experimentación desarrollado, a partir de la metodología propuesta. Finalmente, incluye la especificación del sistema informático de soporte a la propuesta.

El *Capítulo IV: Conclusiones*, analiza los resultados del proceso y recoge las ideas extraídas del proceso de desarrollo del presente trabajo, contrastándolas con las planteadas al inicio del mismo.

El siguiente apartado, *Capítulo V: Trabajos Futuros*, recoge las líneas de desarrollo en

las que la propuesta podría tener continuidad.

El *Capítulo VI: Bibliografía*, incluye el conjunto de referencias bibliográficas citado en la propuesta, estructurado por apartados, para facilitar su uso en futuros trabajos de investigación.

Además de los anteriores apartados, el trabajo incluye dos anexos que aportan información complementaria en relación con la propuesta:

El *Anexo A: Plantilla de descripción de Esquemas de Metadatos*, que contiene la plantilla propuesta para desarrollar descripciones de esquemas de metadatos interpretables por personas.

El *Anexo B: Resultados de la evaluación de repositorios semánticos*, que contiene el resumen de la evaluación de repositorios semánticos, elaborada en la fase de diseño de la plataforma informática de soporte.

1.1 Motivación

Los principales problemas que presentan actualmente los esquemas de metadatos, se resumen en la necesidad de que el usuario tenga conocimiento del esquema para poder utilizarlo. Por *usuario* debe entenderse, tanto personas como computadoras. *Conocimiento del esquema* debe interpretarse en sentido amplio; no sólo conocer de la existencia o localización del esquema, también conocer la estructura y significado de sus términos, además de las recomendaciones para su uso. El desconocimiento de uno o varios de estos aspectos, provoca la carencia en la reutilización de esquemas, dificultades en la interoperatividad y la imposibilidad de realizar búsquedas conceptuales, lo que redundaría en el desarrollo de esquemas ad-hoc, con frecuencia escasamente formalizados, frente a lo que sería deseable, la reutilización de los ya existentes.

Más concretamente, los principales problemas de los esquemas de metadatos pueden resumirse en:

- Dificultades en la interpretación semántica de sus elementos componentes, especialmente relevante para lograr la interoperatividad entre esquemas.
- Problemas en la recuperación mediante motores de búsqueda Web.
- Ambigüedad, ya que la referencia a vocabularios de metadatos es opcional.
- Dificultades en la recuperación, al no existir un repositorio donde se almacenen de forma uniforme y conceptualmente relacionados, al menos aquellos esquemas con mayor popularidad.

Sobre las dificultades en la interpretación semántica de los elementos, la causa principal es de naturaleza terminológica. En la interoperatividad de esquemas, la solución suele consistir en realizar correspondencias, uno contra uno, de los elementos de ambos esquemas. Este enfoque presenta inconvenientes debidos a que:

- Existen problemas de correspondencia entre etiquetas XML, en concreto Buscaldi (Buscaldi, D. et al., 2003) ha señalado, en relación con la desambiguación automática: “...Often tags [in XML] were a combination of lexemes (e.g. ProductList, clubname), others were unintelligible abbreviations (e.g. msrb, cnames, bklong), someone existed but in a different form (e.g. lastname instead of last_name), and others were simply stop list words (e.g. from, to). In all other cases, the approaches were able to assign a sense to the tags (the correctly disambiguated tags were about 40%)...”.
- Existen problemas de Escalabilidad: si se realizan correspondencias uno contra uno, dentro de un conjunto elevado de documentos, la incorporación de un nuevo documento supone múltiples correspondencias contra el resto, de lo que existen numerosos ejemplos (Day, M., 2002). Concretamente, la correspondencia uno contra uno, implica $n!/2*(n-2)!$ correspondencias, frente a las $n-1$ que supone realizarlas de forma centralizada. La correspondencia centralizada precisa de un recurso central, más completo y con un mayor nivel de abstracción, contra el cual relacionar los elementos incluidos en los esquemas.
- Las correspondencias son incompletas debido al grado de especificidad en el dominio. Conceptos presentes en un esquema pueden no estar presentes en el otro esquema, bien por ser conceptos muy específicos, bien por pertenecer a dominios distintos. Del mismo modo, un concepto puede representar varios conceptos del otro vocabulario, por ejemplo, en el caso de términos compuestos.
- La presencia de etiquetas con denominación no identificativa (por ejemplo, poner “persona” cuando se quiere expresar autor del documento), niveles erróneos de abstracción (por ejemplo, poner “autor” cuando se quiere expresar compilador de la obra) o denominaciones poco ilustrativas (por ejemplo “etiquetal”).
- Dificultades de carácter lingüístico: polisemia, sinonimia, finalidad del discurso, problemas idiomáticos o uso de lenguas para fines específicos, semánticas específicas en función del dominio, etc.

También existen problemas en la recuperación de esquemas mediante motores Web que, en relación al lenguaje XML, muestran usualmente dos tipos de deficiencias:

- Los buscadores genéricos, como Google, permiten conocer el número de documentos XML e incluso si tienen cierta combinación de términos en su interior, pero no se puede saber si esos términos se encuentran bajo determinada etiqueta. Además sus arañas (*crawlers*) no están optimizados para este tipo de documentos.
- Los buscadores especializados permiten la recuperación de documentos XML basada en un conjunto predefinido y limitado de DTDs o esquemas. Además, sus interfaces de búsqueda presuponen el conocimiento por parte del usuario de la estructura y finalidad del documento XML (Martin, P.; Eklund, P.W., 2000), (Martin, P.; Eklund, P.W., 2002).

1.2 Hipótesis

Los problemas mencionados hacen imposible realizar una recuperación conceptual. Lamentablemente este tipo de recuperación no es factible sin:

- Consenso de las unidades léxicas y terminológicas representativas para los conceptos. Sin éste se produce ambigüedad conceptual.
- Aspectos sintagmáticos y paradigmáticos de las unidades léxicas y las unidades terminológicas representativas de los conceptos, esto es la estructura y significado de las mismas.
- Mapa conceptual y relacional, que deberá incluir las equivalencias léxicas.

Como solución para la ambigüedad conceptual se propone interrelacionar los elementos, incluidos en los esquemas, con los conceptos incluidos en una ontología de referencia. La ontología de referencia realizaría las funciones de diccionario general mientras que los esquemas de metadatos harían las funciones de diccionarios especializados, superando las limitaciones de los aspectos fraseológicos y las irregularidades en cuanto a la información en las definiciones.

Además de servir como vocabulario controlado y proporcionar significado a los elementos, a través de la ontología de referencia se conseguiría un mapa conceptual y relacional que incluiría y permitiría equivalencias léxicas entre los términos de las etiquetas de los documentos y la estructura ontológica general.

Las principales ventajas de la propuesta parten del hecho de no presuponer ningún conocimiento de la estructura del documento XML por parte del usuario, al darle la posibilidad de realizar no sólo consultas conceptuales sobre un repertorio de esquemas, sino también de incluir sus propios esquemas y documentos en dicho repositorio. Para lograrlo, es preciso desarrollar tanto la estructura como los recursos para la interconexión de elementos XML.

1.3 Objeto

Dotar de un plano semántico a esquemas de metadatos y sus documentos derivados, mediante la relación de sus elementos con conceptos definidos en un recurso semántico compartido vía Web.

1.4 Objetivos

La presente propuesta plantea añadir un plano semántico a la representación de Esquemas de forma que posibilite su recuperación conceptual basada en el significado y contenido de sus elementos.

Objetivos:

- Mejorar la interoperatividad de esquemas de metadatos heterogéneos, permitiendo correspondencias más precisas, desambiguando la representación y

la definición de los términos, así como las relaciones entre conceptos.

- Posibilitar la reutilización de esquemas dentro de un marco extensible que permita incluir definiciones desambiguadas, comprensibles por el usuario y accesibles vía Web.
- Posibilitar una recuperación con un árbol de exploración más amplio, lo que se traduce en un conjunto de resultados más completos, derivados de las relaciones entre conceptos presentes en el modelo conceptual.
- Eliminar la ambigüedad semántica tanto estructural como conceptual mediante una representación más completa de esquemas.
- Recopilar y almacenar un Corpus de referencias a los esquemas más populares, facilitando así su acceso y reutilización.
- Permitir la recuperación conceptual de documentos semánticos, incluyendo búsquedas por el contenido conceptual, así como búsquedas por esquemas y vocabularios de metadatos con un concepto determinado.
- Diseñar un sistema informático que proporcione soporte a la presente propuesta, maximizando su usabilidad.
- Incluir en el diseño del sistema informático de soporte, incorporar metadatos en la representación ontológica de los esquemas, que serán utilizados tanto en la búsqueda como por el algoritmo de posicionamiento.
- Incluir en el diseño del sistema informático la funcionalidad que permita al usuario incorporar sus propios esquemas, que en última instancia serán validados por un sistema de votación de expertos.
- Incluir en el diseño del sistema informático la funcionalidad que permita la recuperación semántica de documentos semánticos y Esquemas asociados, mediante una interfaz Web de acceso libre.

Los objetivos planteados deberán cumplir los siguientes requisitos:

- La solución debe de poderse aplicar a cualquier conjunto de esquemas.
- Debe garantizarse la compatibilidad con sistemas que hagan uso del esquema original, posibilitando la aplicación de principios de simplificación en sistemas no compatibles con el esquema semántico.
- Debido a su popularidad se seguirán las recomendaciones de nombrado de términos y espacios de nombres del DCMI y el W3C.
- La asignación semántica debe permitir la reutilización de elementos en el futuro mediante definiciones consensuadas, no ambiguas, comprensibles y accesibles.
- Se generará la documentación de los esquemas cualificados de forma que se

facilite su interpretación, tanto por usuarios como por computadoras.

- Se debe proporcionar una solución que soporte los criterios de modularidad, extensibilidad, refinamiento y plurilingüismo (Duval, E. et al., 2002).
- Los esquemas cualificados semánticamente constarán de dos documentos: uno formal dirigido a los usuarios; y otro en RDF para el procesamiento automático.

Destacar que el consenso de las definiciones se pretende alcanzar mediante el desarrollo de un recurso Web colaborativo que permita al usuario incluir y gestionar modelos, el diseño de este recurso está incluida dentro del apartado de desarrollo de la plataforma informática de soporte a esta propuesta.

1.5 Metodología

Una vez planteados los objetivos y requisitos de la solución, en este apartado se presenta la metodología desarrollada para la consecución de los mismos. Destacar que la metodología presentada, en el ámbito de cualificación de esquemas, ha sido refinada y validada mediante diversas publicaciones nacionales e internacionales como: la *Fifth International Semantic Web Conference* (Palacios, V. et al., 2006a), la *Conferência Ibérica de Sistemas e Tecnologias de Informação* (Palacios, V. et al., 2006b) o la *International Conference on Dublin Core and Metadata Applications* (Palacios, V. et al., 2006c). La metodología se divide en fases presentadas en la siguiente figura:

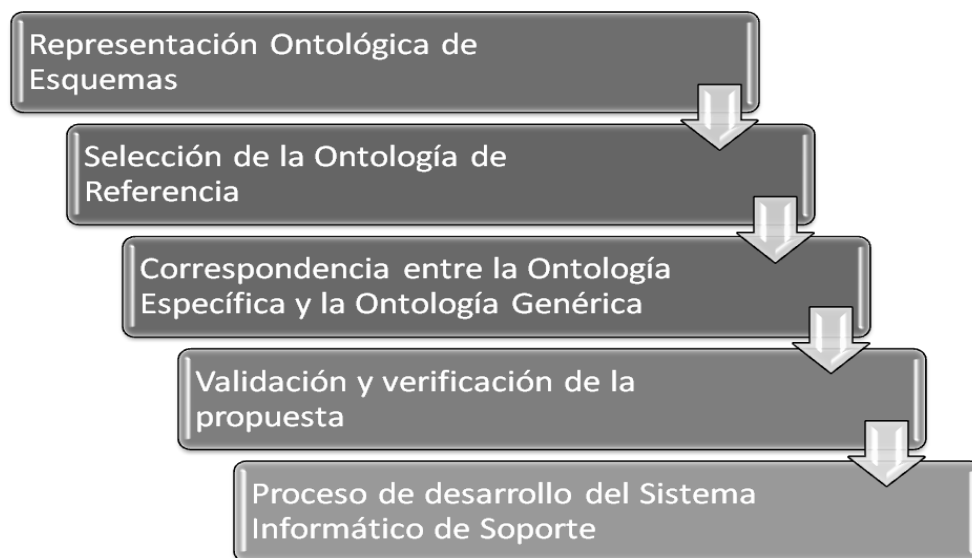


Figura I-1: Principales pasos de la metodología propuesta

A continuación se describen cada una de las fases:

1. Representación ontológica de esquemas: El objetivo de esta fase es lograr la incorporación de semántica a un conjunto de esquemas seleccionados a modo de muestra. Para ello se establecen las siguientes tareas, representadas en la *Figura I-2* y descritas a continuación.



Figura I-2: Tareas de la fase de representación ontológica de esquemas

1.1. Definición de la representación semántica de esquemas: Se definirá el modo de incluir semántica en esquemas de metadatos con el fin de paliar la ambigüedad sintáctica y semántica. Una vez definido, se valorará en función de los siguientes criterios de calidad: extensibilidad, compatibilidad y reutilización.

1.2. Búsqueda y selección de esquemas: Se evaluarán distintos esquemas de metadatos, con el objetivo de seleccionar un conjunto representativo en el cual poner en práctica la propuesta. En el proceso de selección se tendrán en cuenta criterios de popularidad, estabilidad, ámbito y estandarización de los esquemas.

1.3. Representación semántica de los esquemas seleccionados: Una vez seleccionado el conjunto esquemas, se aplicará el método propuesto en el primer punto, obteniéndose la representación semántica de los mismos. Durante el proceso se generará una metodología para la representación semántica de esquemas, inducida a partir de los casos de ejemplo, de forma que pueda ser aplicada a cualquier esquema, fuera del conjunto seleccionado.

1.4. Publicación de esquemas semánticos: Se procederá a la publicación de las representaciones semánticas de los esquemas, generadas en el paso anterior. De este modo, los esquemas semánticos quedarán a disposición de los usuarios, incluidos sistemas informáticos, para su uso y reutilización.

2. Selección de la ontología de referencia: El objetivo de esta fase es la evaluación y selección de una ontología existente que, representada de modo compatible con el resto de componentes de la propuesta, pueda ser utilizada como ontología de referencia para la desambiguación semántica de los esquemas seleccionados. Para ello se establecen las siguientes tareas, representadas en la *Figura I-3* y descritas a continuación.

2.1. Definición de requisitos de la ontología de referencia: Se establecerá el conjunto de requisitos que deberá satisfacer la ontología para poder ser utilizada como ontología de referencia en la propuesta. Derivados de los criterios de calidad anteriormente expuestos, valorarán si la ontología es extensible, modificable, consultable, navegable,

su lenguaje de definición, si es referenciable mediante un sistema de identificación formal o popularidad, entre otros.

2.2. Evaluación y selección de la ontología de referencia: Partiendo de los requisitos establecidos, se llevara a cabo la selección de una ontología entre varias de las ontologías de acceso libre existentes.

2.3. Representación compatible de la ontología de referencia: Una vez seleccionada la ontología se definirá el modo de representación de la misma, de modo que sea compatible con el resto de componentes de la propuesta, permitiéndose así su integración. La representación atenderá a los requisitos establecidos para la selección de la ontología de referencia y será evaluada en función de los criterios de calidad relativos a: extensibilidad, compatibilidad y reutilización.

2.4. Publicación de la ontología de referencia: Se procederá a la publicación de la ontología de referencia, en su representación compatible con la propuesta, de modo que quede disponible para los usuarios, incluyendo sistemas informáticos, para su uso y reutilización.

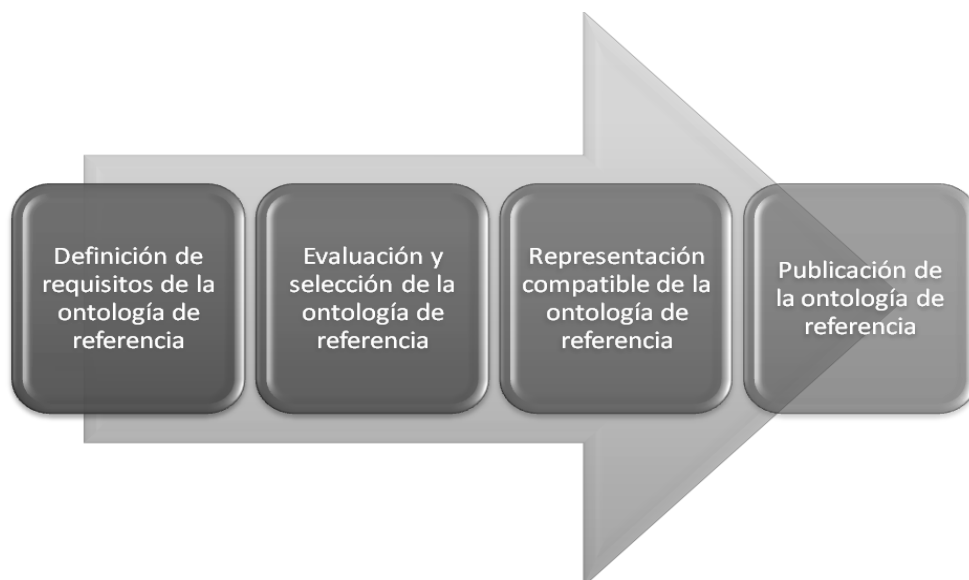


Figura I-3: Tareas de la fase de selección de la ontología de referencia

3. Correspondencia entre esquemas y la ontología de referencia: Una vez desarrolladas las representaciones semánticas de los esquemas y seleccionada la ontología de referencia, se iniciará el proceso cuyo objetivo es la interrelación de los conceptos incluidos en los esquemas semánticos con los incluidos en la ontología de referencia. Para ello se establecen las tareas mostradas en la *Figura I-4* y detalladas a continuación.



Figura I-4: Tareas de la fase de correspondencia entre esquemas y la ontología de referencia

3.1. Evaluación y selección del método de correspondencia: Deberá definirse el modo de interrelacionar las representaciones semánticas de los esquemas con la ontología de referencia. Con este fin se analizarán los métodos existentes y se seleccionará aquel que mejor se adecúe a la propuesta, esto es, en base a criterios de extensibilidad, modificabilidad, navegabilidad, accesibilidad, etc. Una vez seleccionado el método de correspondencia, se definirá el modo de articularlo, tomando como referencia los criterios de calidad anteriormente expuestos, a saber: extensibilidad, compatibilidad y reutilización.

3.2. Correspondencia de conceptos entre esquemas y la ontología de referencia: Una vez establecido el método de correspondencia, se llevará a cabo el proceso de correspondencia del conjunto de esquemas seleccionado con la ontología de referencia seleccionada. Durante el proceso se generará, mediante el método de inducción, una metodología de correspondencia que, a través de un conjunto de pasos, permita la correspondencia futura de esquemas no incluidos en el conjunto seleccionado inicialmente. La metodología analizará y dará solución a aquellos casos especiales encontrados durante el proceso de correspondencia.

3.3. Publicación de la ontología de correspondencia: Como última actividad dentro de esta fase, se procederá a la publicación de los artefactos generados durante el proceso de correspondencia. De este modo, quedarán disponibles para su uso y reutilización por parte de usuarios y aplicaciones.

4. Validación y verificación de la propuesta: Una vez planteada a nivel teórico, se procederá a validar y verificar la propuesta con el fin de garantizar: por un lado, que satisface los requisitos para los que se diseñó, esto es, validar que realiza su función; y por otro, que realiza su función correctamente, esto es, verificar la corrección de los resultados que proporciona. Con este fin, se definirán un conjunto de consultas conceptuales que la propuesta deberá satisfacer y cuyos resultados serán comparados con los que teóricamente deberían obtenerse.

5. Proceso de desarrollo del sistema informático de soporte: El objetivo que se aborda en este apartado es el diseño de un sistema informático que soporte la gestión y acceso a los esquemas definidos, según la solución propuesta. Para ello se definen las tareas incluidas en la *Figura I-5* y detalladas a continuación.

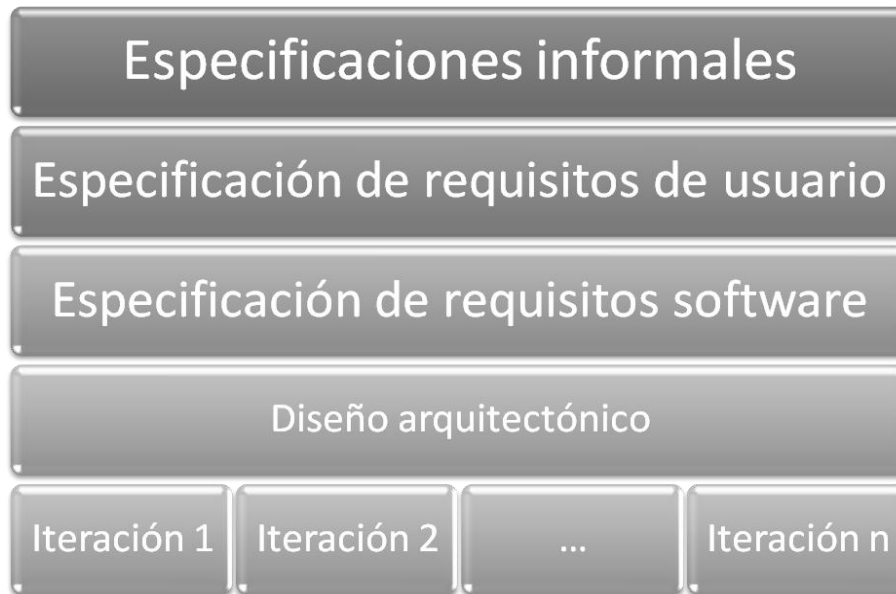


Figura I-5: Tareas del proceso de desarrollo del sistema informático de soporte

5.1. Especificaciones informales: Se evaluará el alcance y necesidades del sistema mediante el desarrollo de un conjunto de *Especificaciones Informales*, esto es, un conjunto de requisitos no formalizado que proporcionen una visión general del alcance del proyecto. En función de dichas especificaciones, se establecerá el ciclo de vida más adecuado para el desarrollo del sistema.

5.2. Especificación de requisitos de usuario: Independientemente del ciclo de vida seleccionado, se realizará una primera especificación de requisitos de usuario completa. El objetivo es obtener, lo antes posible, el conjunto más completo posible de funcionalidades y restricciones que debe satisfacer el sistema. Dentro de estos requisitos deberán encontrarse los siguientes:

5.2.1. Se generará un primer corpus con los esquemas seleccionados en la propuesta. La inclusión en la colección ha atendido a criterios de popularidad, estandarización, ámbito y estabilidad.

5.2.2. La adición de nuevos esquemas al corpus se priorizará cuanto mayor sea el número de documentos asociados y citas (por ejemplo, vía espacio de nombres). La selección y búsqueda de *esquemas* para su almacenamiento, se realizará en base a criterios, tales como el número de documentos asociados, estadísticas públicas (por ejemplo, las publicadas por Swoogle (Swoogle, 2007a)), citas y la autoría por la que viene avalada. Para conseguir este fin, se combinará la búsqueda en repositorios de *esquemas* con la búsqueda automatizada en motores Web. Se pondrá especial cuidado

en seleccionar recursos relevantes para usuarios y hablantes de nuestro entorno.

5.2.3. Se incluirá un subsistema para representar los esquemas y la ontología de referencia. Una vez modelados como ontologías de dominio, los elementos de los esquemas serán relacionados con los conceptos de la ontología genérica, haciendo uso del método propuesto anteriormente, logrando así una correspondencia de los esquemas siguiendo un *modelo centralizado*.

5.2.4. Los elementos de los esquemas, una vez relacionados con los conceptos representados en la ontología genérica, serán compartidos vía Web y se incluirán mecanismos de actualización, control y revisión de esquemas. Este proceso de asignación entre elementos y conceptos se caracterizará por:

5.2.4.1. Una evaluación de las estadísticas públicas y propias sobre grado de utilización de los distintos esquemas y vocabularios, teniendo especial consideración en la relevancia de la autoría, se establecerá una prioridad en el procesamiento de los esquemas recopilados en el corpus, realizándose en orden descendente de impacto en uso. De este modo se asegura, desde un principio, que el recurso beneficiará al máximo de usuarios posible.

5.2.4.2. En la etapa de asignación, los expertos en el dominio² asociarán cada uno de los elementos de los esquemas, transformados en ontologías de dominio, a un concepto representado en la ontología genérica. En esta etapa se aplicará la metodología de correspondencia propuesta para solventar los problemas emergentes de:

5.2.4.3. Tratamiento de atributos y los subelementos en la correspondencia de elementos

5.2.4.4. Tratamiento de términos y/o conceptos inexistentes en la ontología genérica.

5.2.4.5. Presencia de definiciones heterogéneas en la descripción del elemento del espacio de nombres.

La ontología será modificada y/o extendida, dependiendo de los problemas encontrados, para mejorar su capacidad de descripción de los conceptos expresados en los esquemas. Este proceso puede conllevar acotar, ampliar, eliminar o añadir conceptos al recurso. Se

² Los expertos en el dominio realizarán la asignación de los elementos a conceptos, asesorados por un ingeniero responsable en el dominio. Si la mayoría de expertos coincide en la asignación elemento-concepto, ésta se da por válida. Si no existe consenso, el experto responsable en el dominio tiene la última palabra. Los roles en el dominio serían:

Ingeniero responsable en el dominio: Instancia última de los ingenieros en el dominio con juicio decisorio.

. Experto en el dominio: persona con conocimientos en el dominio habilitado para realizar la asignación entre elementos y conceptos.

. Experto responsable en el dominio: Instancia última de los expertos en el dominio con juicio decisorio.

evaluará la adición de recursos léxicos a través de correspondencias externas a las ontologías de referencia, tal y como prevé el método de correspondencia propuesto.

En caso de que no exista un consenso entre los expertos, el responsable de los expertos decidirá objetivamente la asignación correcta.

5.2.4.6. Además del método descrito para la incorporación de esquemas y las asignaciones de sus elementos a conceptos, se proporcionará a los usuarios la funcionalidad necesaria para que puedan añadir, vía Web, sus propios esquemas. En este proceso, el usuario deberá asignar, igualmente, los nuevos elementos a conceptos. Con el objetivo de mantener la integridad y coherencia del recurso, estas incorporaciones serán supervisadas por los expertos del dominio siguiendo el sistema de decisión descrito anteriormente.

Periódicamente, una vez realizada la comprobación de la integridad de las ontologías de dominio, se comprobará la integridad de las asignaciones con los conceptos de la ontología genérica. Esta comprobación se automatizará en lo posible, recurriéndose al sistema de decisión expuesto anteriormente, en caso de conflicto³.

5.2.5. Se incorporará información relativa a los distintos esquemas o metadatos, en concreto:

5.2.5.1. Grado de implantación: Organismos que lo utilizan, número de documentos XML que lo referencian (este cálculo se realizará en un primer momento mediante búsquedas de mención de la URI asociada), proyectos en los que se ha utilizado, etc.

5.2.5.2. Materia y dominio asociado.

5.2.5.3. Autor y organismo que respalda la autoría

5.2.5.4. Fecha de creación y de última actualización

5.2.5.5. Prefijos asociados a los Espacios de Nombres.

5.2.5.6. Nivel de cesión del recurso: licencia a la que está sujeta el recurso.

5.2.5.7. Hipervínculos a documentos XML basados en los *esquemas* del repositorio. Esta inclusión permitiría buscar el contenido por determinada etiqueta

Junto con las ontologías de esquemas se desarrollará la funcionalidad necesaria para su mantenimiento. Esta funcionalidad incluirá la consulta y recuperación de *esquemas* y etiquetas.

³ El proceso de comprobación de la integridad del recurso es responsabilidad del Ingeniero del dominio. En su caso, podrá requerir de la evaluación de asignaciones elemento-concepto, por parte del grupo de expertos en el dominio.

Para evitar la obsolescencia de los recursos y la consiguiente proliferación de enlaces erróneos o recursos desaparecidos, se incorporará una comprobación automatizada de la integridad de la información almacenada, que será ejecutada periódicamente.

5.2.6. Incorporar la funcionalidad que facilite la recuperación conceptual con la posibilidad de utilizar operadores booleanos, evitando la necesidad de un conocimiento previo del recurso por parte del usuario. Se proporcionarán, en principio, las siguientes consultas:

5.2.6.1. Recuperación por conceptos.

5.2.6.2. Esquemas en los que se representa un concepto.

5.2.6.3. Recuperación conceptual de documentos. Mediante la desambiguación del dominio y del campo semántico en el que se encuentra un término dado en un documento concreto.

5.2.6.4. Documentos que hacen uso de alguno de los esquemas que definen el concepto de búsqueda, permitiendo la asignación de valores a dicho concepto, es decir, permitiendo la recuperación de documentos por un concepto, indicando la instancia concreta del mismo.

5.2.6.5. Documentos en los que aparezca un concepto, con o sin valor, dado un esquema concreto.

5.2.6.6. Recuperación por la información adicional incorporada a los esquemas.

Se aplicará un algoritmo de posicionamiento para los resultados de las búsquedas realizadas. Este posicionamiento estará basado en criterios de popularidad, localización de los términos de búsqueda, número de documentos aproximado que los hacen referencia, etc., almacenados como metadatos de los esquemas.

5.2.7. Incorporar un motor de búsqueda que permita la recuperación de los documentos XML asociados a los esquemas almacenados en las ontologías de dominio. El sistema incorporará los siguientes elementos:

5.2.7.1. Una araña que recorra la Web localizando documentos XML que hagan mención a los Esquemas, a través de las URIs presentes en los documentos, y que recolecte aquellos que satisfagan dicha condición.

5.2.7.2. Un analizador de documentos que permita su extraer su información, (título, URI, fecha de modificación, etc.) así como la relativa a sus elementos y valores concretos. La información obtenida será relacionada con los esquemas en los que se basan los documentos, de modo similar a las soluciones que suelen adoptar los ficheros inversos para indización de documentos.

5.3. Especificación de requisitos software: Partiendo del conjunto de requisitos de usuario elicitado en la tarea anterior, se procederá a su refinamiento y formalización. El objetivo es obtener un conjunto completo, formal, consistente y no ambiguo de requisitos que represente las capacidades y restricciones que debe soportar la aplicación.

Se incluirán en esta tarea los aspectos relativos al interfaz de usuario y diseño de pantallas.

5.4. Diseño arquitectónico: Una vez definido el dominio del problema, en esta tarea se diseñará la arquitectura hardware y software que mejor sirva al desarrollo del proyecto. Se hará uso de estilos y patrones de diseño con el fin de maximizar la calidad del proyecto en cuanto a su mantenibilidad, esto es, modificabilidad y adaptabilidad. Se evaluarán en esta tarea las tecnologías existentes que faciliten la consecución del proyecto, esto es: lenguajes, marcos de desarrollo, librerías o entornos integrados de desarrollo.

5.5. Diseño detallado, codificación y pruebas: Atendiendo a la experimentalidad del proyecto y las tecnologías a utilizar, las fases de diseño, codificación y pruebas se realizarán de modo iterativo. De este modo se posibilita el control y corrección de las posibles desviaciones durante las últimas fases del desarrollo. Partiendo de la arquitectura propuesta, cada iteración abordará el diseño de componentes específicos, su codificación y sus pruebas, tanto unitarias como de integración.

Estas fases quedan fuera del ámbito de esta propuesta, no obstante, desde octubre de 2007 se está desarrollando un proyecto, dentro del Plan Nacional de I+D+i, titulado “*Desarrollo de un sistema de Recuperación Conceptual mediante Niveles Semánticos en la representación de Esquemas de Metadatos*”, basado en el planteamiento propuesto en esta tesis.

Para finalizar la presentación de la metodología, destacar que en la elaboración del presente documento se ha optado por los siguientes convenios en cuanto al uso de citas:

- Se utilizarán llamadas a la bibliografía final porque procura informaciones más completas, sistematiza el material consultado y permite extraer con facilidad informaciones globales de la bibliografía sobre el tema. Además, evita al lector buscar los textos en las notas, página por página.
- Se utilizarán notas al pie para incluir comentarios breves con aclaraciones sobre el texto, de interés para el lector.

1.6 Fuentes

Debido al carácter tecnológico y la aplicación dentro del espacio Web de la presente investigación, la mayor parte de la información se encuentra disponible en Internet. La Web es uno de los medios más importantes para la publicación de artículos y resultados dentro de este campo de investigación.

Las organizaciones de estandarización también utilizan la Web para poner a disposición de los investigadores e industrias los estándares que desarrollan. De este modo se ha tenido acceso a recomendaciones y estándares de la *International Standardization Organization (ISO)*, del *World Wide Web Consortium (W3C)* y del *Internet Engineering Task Force (IETF)*, entre otros.

La gran mayoría de los congresos mantienen recursos Web para publicar sus actas,

como medio para mejorar su accesibilidad. Sólo en algunos casos ha sido necesario consultar las actas impresas.

Las revistas especializadas han aportado información sobre revisiones del estado de la cuestión en ámbitos concretos, así como información detallada sobre las últimas herramientas y métodos. Muchas de ellas son accesibles desde Web como *Novatica* o *Science Direct*, otras han sido consultadas a partir del catálogo de la Biblioteca de la Universidad Carlos III de Madrid, como *IEEE Intelligent Systems* o *Computer Networks and ISDN Systems*. También en este catálogo se encuentran disponibles otros recursos electrónicos utilizados, como *ACM Digital Library*, *IEEE Xplore*, *ISI Web of Knowledge*, *CiteSeer* o *Google Scholar*, entre otros.

Las monografías, destinadas a albergar información con una vida media más larga, han sido consultadas en aquellos aspectos de la investigación con mayor estabilidad, por ejemplo, metodologías de desarrollo.

Como apoyo a la investigación, se han hecho uso de otras fuentes como diccionarios, manuales o enciclopedias.

Finalmente, destacar que la investigación se ha realizado entre los años 2005 y 2010, principalmente en los idiomas inglés y español.

Capítulo II: Fundamentos Teóricos

2.1 La Web Semántica

2.1.1 Introducción

Desde 1989, año en el que nació la *World Wide Web* (WWW) en los laboratorios del CERN, la Web se ha convertido en un espacio de información global formado por miles de millones de páginas. Curiosamente, el gran número de páginas Web existentes garantiza la existencia de información sobre casi cualquier tema, al tiempo que complica cada vez más encontrar la información que se busca. La principal causa de este problema proviene del hecho de que la información incluida en la Web está, primordialmente, dirigida a usuarios humanos, lo que dificulta el acceso a la misma y su semántica, a usuarios no humanos (máquinas).

En 1998, Tim Berners-Lee publicó en la página principal del *World Wide Web Consortium* (World Wide Web Consortium, 2010a) (W3C en adelante) su conocido artículo “*A Roadmap to the Semantic Web*” (Berners-Lee, T., 1998). En dicho artículo se introducía por primera vez el término Web Semántica (*Semantic Web*) destacándose la necesidad de expresar la información de forma que ésta fuera procesable por máquinas. El artículo presentaba un conjunto de pasos hasta lograr una Web en la que el razonamiento fuera automático, llevado a cabo por máquinas, y distribuido.

En 2001, Berners-Lee publica un nuevo artículo “*The Semantic Web*” (Berners-Lee, T.; Hendler, J.; Lassila, O., 2001), en el que presenta las principales características que tendrá la futura Web y afirma que las máquinas facilitarán nuevas prestaciones al mejorar su capacidad de procesar y comprender la información dispersa por la Web. Como solución a las limitaciones semánticas de la actual Web, propone hacer procesable de forma automática el contenido de la Web, definiendo la Web Semántica como:

“The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation...”

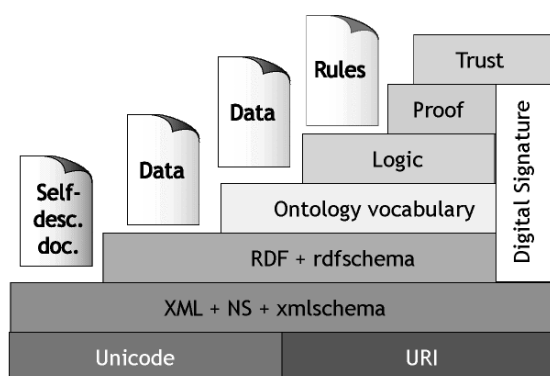
De este modo, la Web Semántica no sería una nueva Web sino la extensión de la existente, mediante la adición de metadatos que describan la semántica de las páginas de forma procesable por máquinas.

Desde entonces la Web Semántica se ha convertido en un importante tema de investigación con multitud de grupos de investigadores, congresos y grupos de trabajo. La semántica de la información es un aspecto clave para encontrar la forma de expandir

un espacio Web donde, actualmente, la mayoría de los recursos sólo pueden ser encontrados a través de búsquedas sintácticas.

2.1.2 Arquitectura

Para lograr una Web Semántica basada en metadatos procesables por máquinas, Berners-Lee propone un conjunto de capas que definen distintos niveles de representación semántica. De este modo, se progresa en la representación desde el nivel inferior, meramente sintáctico, hasta el nivel superior, donde es posible discernir si la información es fiable. La estructura que muestra la *Figura II-1* fue presentada por Tim Berners-Lee en una conferencia sobre el *Extensible Markup Language (XML)* en el año 2000.



*Figura II-1: Estructura y lenguajes de la Web Semántica*⁴

La estructura propuesta establece el siguiente conjunto de capas, en función de su capacidad de representación semántica, donde cada una de ellas presta un servicio a los niveles superiores.

2.1.2.1 Capa sintáctica

Se encuentra situada en el nivel inferior de la pirámide y tiene como objetivo facilitar el intercambio de datos a través de una sintaxis de serialización concreta. En el nivel superior de la capa, suele situarse XML por ser el lenguaje de intercambio de datos más utilizado por las capas superiores, no obstante, no es obligatorio y podrían utilizarse otros lenguajes. XML y XML Schema proporcionan la sintaxis necesaria para el intercambio y persistencia de la información. En el apartado 2.1.3.1 *Extensible Markup Language (XML)* y *XML Schema* se describen las principales características de este lenguaje.

El nivel XML hace uso del nivel inferior, donde se sitúan *Unicode* y los *URIs*. Unicode (Unicode, 2010) es una codificación de caracteres internacionales donde cada carácter recibe un identificador unívoco, de este modo, las páginas XML pueden representarse sin importar el idioma, plataforma o programa que las gestione. Dentro de este nivel también se incluyen los Identificadores Uniformes de Recursos (*Universal Resource*

⁴ Revisiones recientes de este gráfico como (Hewett Research, 2009), incluyen una última capa, encima de la de confianza, que representa el interfaz de usuario y las aplicaciones.

Identifier: URI). Los URIs (Berners-Lee, T., 1994) permiten la identificación unívoca de recursos dentro de la Web Semántica. En el caso de una página Web, el URI puede ser el URL (*Uniform Resource Locator*) de la página. En general, un URI no tiene por qué permitir el acceso a un recurso a través de Web, sino que su propósito es identificar de forma no ambigua el recurso.

2.1.2.2 Capa RDF(S)

La siguiente capa dentro de la pirámide tiene como objetivo proporcionar la suficiente flexibilidad para representar conceptos y reglas lógicas relacionadas, generando así valor añadido. El Marco de Descripción de Recursos (*Resource Description Framework: RDF*) (World Wide Web Consortium, 1999a) es una recomendación del W3C que define un lenguaje simple para expresar relaciones en forma de terna⁵ Recurso-Propiedad-Valor. La especificación de *RDF-Schema* (World Wide Web Consortium, 2004a) es también una recomendación del W3C cuyo objetivo es facilitar la interpretación semántica de RDF. De este modo, permite la representación de modelos semánticos, orientados a objetos, en forma de redes de conceptos. Entre las primitivas⁶ que incluye, se encuentran aquellas que permiten la definición de: Clases, propiedades, subpropiedades, restricciones de dominio y rango, etiquetas y comentarios. En el apartado 2.1.3.2 *Resource Definition Framework (RDF)* y *RDF-Schema* se describen las principales características de ambas recomendaciones.

2.1.2.3 Capa ontológica

La siguiente capa dentro de la estructura tiene como objetivo proporcionar mayor expresividad que la expresada por la capa inferior. Concretamente, es necesario poder representar primitivas complejas tales como: propiedades transitivas, restricciones de cardinalidad en las relaciones, etc. Existen distintos lenguajes que proporcionan la expresividad semántica necesaria en esta capa, entre otros: OIL (Lamarca, M.J., 2002), DAML+OIL (Defense Advanced Research Projects Agency, 2002) y OWL (Bechhofer, S. et al., 2004). Este último lenguaje *OWL Web Ontology Language* es una recomendación del W3C desarrollada para facilitar el procesamiento automático de contenido Web. Posee tres versiones distintas que van de la menos expresiva, *OWL Lite*, pasando por una intermedia y decidible, *OWL DL (DL: Description Logic)*, hasta llegar a la versión de máxima expresividad semántica *OWL Full*. En el apartado 2.1.3.3 *OWL Web Ontology Language* se incluyen las principales características de este lenguaje.

⁵ Se ha utilizado el término *terna* como traducción del término inglés *triple*. En la literatura se pueden encontrar traducciones como *tripleta* o *tripla*, también utilizadas con frecuencia. Según el diccionario de la RAE, tanto *terna* como *tripleta* tienen un significado similar a trío, como conjunto de tres cosas. *Tripla*, por el contrario, tiene un significado próximo a *triple*, como conjunto de tres cosas iguales. Se ha seleccionado el término *terna*, aunque podría haber sido también *tripleta*, por aproximarse mejor a la semántica del concepto. Del mismo modo, no se aconseja el uso del término *tripla*.

⁶ Ver *Capítulo VIII:Glosario*

2.1.2.4 Capa lógica

La capa lógica tiene como objetivo permitir la inferencia necesaria para proporcionar explicaciones a las respuestas obtenidas. Para ello deben definirse las reglas lógicas que permitan inferir nuevo conocimiento mediante un proceso de razonamiento. Para poder comprobar que los resultados son correctos es preciso traducir el razonamiento interno a un lenguaje de representación de pruebas (*proof*).

2.1.2.5 Capa de prueba y confianza

La capa más alta de la pirámide tiene como objetivo proporcionar los mecanismos para comprobar que la información recibida es coherente desde un punto de vista lógico. Apoyándose en el conjunto de reglas de la capa inferior, será capaz de mostrar dicho razonamiento de forma inteligible para los humanos de forma que pueda ser validable y, por tanto, confiable. Junto con la necesidad de validar la información desde el punto de vista lógico, existe la necesidad de validarla desde el punto de vista de la fiabilidad. De este modo, para confiar en la información debemos poder comprobar que es válida lógicamente y que es fiable, en cuanto a que: proviene de la fuente correcta, no ha sido alterada, no puede ser repudiada por su emisor, etc. La Firma Electrónica (*Digital Signature*) incluida en la estructura de la pirámide, incorpora los mecanismos de seguridad que garantizan la fiabilidad de la información. De este modo, el objetivo final es lograr una Web de Confianza (*Web of Trust*) donde la información intercambiada sea confiable a través de su certificación.

El W3C es el organismo encargado de dirigir los procesos de estandarización de los lenguajes para la Web Semántica. A través de los distintos grupos de trabajo, especializados en distintas áreas de investigación, desarrolla recomendaciones en conjunción con entidades académicas, empresas y particulares. La pirámide propuesta por Tim Berners-Lee, incluyendo el estado actual de los distintos estándares, está representada en la *Figura II-2*:

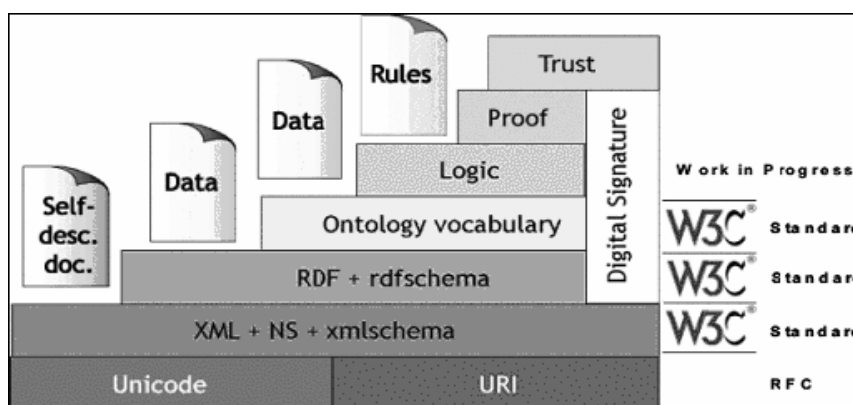


Figura II-2: Estructura y estándares de la Web Semántica. Extraído de (Sure, Y.; Studer, R., 2005)

Tal y como se representa, actualmente existen recomendaciones para las capas sintáctica a la ontológica inclusive. Actualmente se está trabajando en las capas más altas de la pirámide: lógica, prueba y confianza.

2.1.3 Lenguajes

2.1.3.1 Extensible Markup Language (XML) y XML Schema

XML es el estándar del W3C (World Wide Web Consortium, 2008) predominante para codificación e intercambio de datos entre aplicaciones. En XML los datos están agrupados en elementos delimitados por etiquetas. Estos elementos pueden estar anidados y contener atributos. De este modo, XML permite incluir *metadatos* (datos que describen datos) dentro de los documentos, de modo que la información incluida pueda ser procesada de forma automática.

La estructura de elementos, subelementos y atributos, así como la cardinalidad viene frecuentemente expresada en un documento denominado DTD (*Document Type Definition*) mediante una gramática formalizada. Las DTD, aunque fáciles de interpretar y muy utilizadas, tienen una serie de desventajas como son el no estar escritas con sintaxis XML o la escasa definición de tipos de datos, por este motivo se definió el Esquema XML (*XML Schema*) (World Wide Web Consortium, 2004b).

Los esquemas son documentos escritos en XML haciendo uso de espacios de nombres (*namespaces*) asociados mediante su URI y, que como los DTDs, expresan la estructura de los documentos XML. De forma similar a los DTD pero con mayor capacidad de representación sintáctica, permiten definir qué elementos puede contener un documento XML, cómo están estructurados, qué atributos y de qué tipo pueden tener esos elementos. Por lo anterior podemos decir que XMLS es un *metalenguaje* puesto que permite establecer la estructura y los elementos para generar documentos XML, es decir, permite definir nuevos lenguajes XML para su uso en dominios concretos, de forma flexible y extensible. Esquemas con elementos definidos en un alto nivel de abstracción, como OWL, expresan un marco común para la incorporación de ontologías (Mihalcea, R.; Mihalcea, S.I., 2001).

Según el W3C (World Wide Web Consortium, 2009a) los espacios de nombres, mediante la inclusión de un prefijo asociado al URI, constituyen un método sencillo para cualificar elementos y atributos en XML, permitiendo así su posterior reutilización. Los espacios de nombres se crearon para resolver conflictos en las denominaciones de elementos procedentes de distintas fuentes. En los documentos XML, para hacer referencia a un término del espacio de nombres se utiliza el Nombre Cualificado (*Qualified Name: QName*). El QName está formado por un prefijo seguido de dos puntos y el nombre del término. En la cabecera del documento, el prefijo estará declarado como un espacio de nombres junto con su URL. El conjunto de espacios de nombres no está acotado, ventaja que suele provocar la utilización en los documentos de forma no homogénea.

Las principales ventajas que proporciona XML pueden resumirse en:

- Proporciona la estructura sintáctica necesaria para que la información contenida en documentos Web pueda ser procesada tanto por humanos como por máquinas.
- Permite separar la información de un documento de la presentación de la misma,

a diferencia de otros lenguajes de marcado, por ejemplo HTML (*Hipertext Markup Language*).

- Establece una representación estándar para información proveniente de distintas fuentes, lo que facilita la intercomunicación entre aplicaciones.

Las limitaciones de XML se encuentran al intentar especificar la semántica del documento, es decir, al rebasar la especificación sintáctica del documento: estructura y elementos permitidos. XML no permite especificar la semántica formal del documento, si bien permite definir elementos como *Autor* o *FechaNacimiento*, estas expresiones están carentes de significado desde el punto de vista de una máquina, al igual que si se hubieran nombrado como *Elemento1* y *Elemento2*. Para el usuario humano ambas etiquetas tienen significado, aportado por la interpretación personal de la etiqueta.

Ha de observarse la capa en la que se encuentra XML según la pirámide de lenguajes de la Web Semántica descrita en el apartado anterior, la capa sintáctica. Teniendo en cuenta cuál es su función en dicha estructura, la limitación semántica no es tal, dado que la especificación semántica está delegada en capas superiores. De este modo otros lenguajes, como veremos a continuación, son los encargados de definir aspectos semánticos de los documentos.

Para finalizar esta breve introducción a XML es preciso comentar otras tecnologías relacionadas con XML, a saber:

- XLink 1.0 (World Wide Web Consortium, 2001): Especificación que define elementos insertables en documentos XML para crear o describir enlaces entre recursos.
- XHTML 1.0 (World Wide Web Consortium, 2002): Especificación en la que se reformula HTML 4 como aplicación XML 1.0. Proporciona compatibilidad con HTML y las bases para futuras ampliaciones de XHTML.
- DOM 1.0 (World Wide Web Consortium, 1998): Especificación del modelo de objetos de un documento (*Document Object Model*) consistente en una plataforma y un interfaz independiente de lenguajes que permite el acceso y manipulación del contenido, estructura y estilo de los documentos XML.
- CSS 2 (World Wide Web Consortium, 2009b): Primera revisión de la especificación de hojas de estilo en cascada (*Cascading Style Sheets*), nivel 2. Compatible con CSS nivel 1, permite la especificación de estilos a documentos estructurados (HTML, XML, etc.).
- RFC 1630 (Berners-Lee, T., 1994): Especificación de los Identificadores Universales de Recursos (*Universal Resource Identifier: URI*).
- XML Namespaces (World Wide Web Consortium, 2009a): Especificación que proporciona un método de cualificación de los nombres utilizados en documentos XML, mediante la asociación de cada espacio de nombres con un URI.

- XML Infoset (World Wide Web Consortium, 2004c): Especificación que proporciona un conjunto abstracto de datos para referenciar información dentro de un documento XML.
- XSL 1.1 (World Wide Web Consortium, 2006a): Lenguaje para la especificación de hojas de estilo para documentos XML.
- XSLT 1.0 (World Wide Web Consortium, 1999b): Especificación de la sintaxis y semántica del lenguaje de transformación de documentos XML en otros documentos XML. Forma parte de XSL e incluye etiquetas para especificar el formato de documentos XML.
- XPointer (World Wide Web Consortium, 2003): Especificación que define el lenguaje a utilizar como base para localizar un fragmento de información identificado por un URI.
- XPath 1.0 (World Wide Web Consortium, 1999c): Lenguaje para acceder a partes de un documento XML. Utilizado por XSLT y XPointer.

2.1.3.2 Resource Definition Framework (RDF) y RDF-Schema

El Marco de Descripción de Recursos (*Resource Description Framework: RDF*) (World Wide Web Consortium, 1999a) es el modelo de datos para la representación de metadatos en la Web Semántica. El modelo de datos se soporta sobre tres conceptos fundamentales:

- **Recurso (*Resource*):** Un recurso es *algo* sobre lo que podemos hacer una *sentencia*. Un recurso puede ser cualquier entidad: una persona, un libro, un lugar, etc. Todo recurso está identificado por un URI que lo identifica de forma unívoca aunque que no tiene por qué proporcionar acceso vía Web al recurso. En el caso de una página Web el URI puede ser el URL de dicha página.
- **Propiedad (*Property*):** Una propiedad define un aspecto, característica, atributo o relación de un recurso. Las propiedades poseen un significado y pueden tener relaciones con otras propiedades. La definición de la propiedad incluye los valores que puede tomar así como los tipos de recursos a los que puede asociarse.
- **Sentencia (*Statement*):** Una sentencia es una terna formada por un sujeto, predicado y objeto. El sujeto es el recurso a describir, el predicado es la propiedad que define una característica del objeto y el objeto es el valor que toma la propiedad para el recurso que define. Además, una propiedad es un recurso y el valor de la sentencia puede ser tanto un valor simple, como otro recurso, por ejemplo especificado mediante un URI.

Las sentencias son la estructura fundamental en la que se basa el modelo de datos RDF. Éstas pueden ser expresadas de forma textual, típicamente mediante ternas recurso-propiedad-valor (ver *Ejemplo II-1*), o de forma gráfica, mediante un grafo dirigido. En la representación gráfica los recursos y los valores se representan mediante nodos,

mientras que las propiedades se representan mediante arcos dirigidos desde el recurso al valor correspondiente (ver *Figura II-3*).

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:exterms="http://www.example.org/terms/">
  <rdf:Description rdf:about="http://www.example.org/index.html">
    <exterms:creation-date>August 16, 1999</exterms:creation-date>
  </rdf:Description>
</rdf:RDF>
```

Ejemplo II-1: Representación textual de una sentencia RDF. Extraído de (World Wide Web Consortium, 2004d)

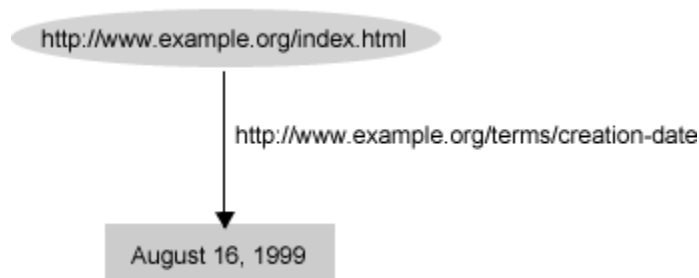


Figura II-3: Representación gráfica de una sentencia RDF. Extraído de (World Wide Web Consortium, 2004d).

Tanto en el ejemplo como en la figura, se muestra la representación de la sentencia que especifica, para el recurso “<http://www.example.org/index.html>”, la propiedad fecha de creación (*creation-date*) y le asigna el valor “August 16, 1999”. En el caso de la representación textual se ha incluido un ejemplo de RDF serializado con XML. Como se puede observar, y con el fin de facilitar su interpretación, en la representación gráfica suele representarse con un óvalo los recursos y con un cuadrado los valores.

La serialización de RDF con XML, consiste en la posibilidad de expresar documentos RDF mediante XML. En gran medida, esto ha contribuido al éxito en cuanto a uso y difusión de RDF en la Web. No obstante, a diferencia de los documentos XML, los documentos RDF no son serializados por un esquema XML, es decir, no son restringidos por un esquema, sino que conforman un grafo donde los elementos que pueden representarse, el vocabulario, son definidos en un esquema representado mediante el lenguaje Esquema RDF (*RDF Schema: RDFS*) (World Wide Web Consortium, 2004a).

RDFS proporciona un conjunto de tipos básicos y un método para definir nuevos tipos, asociados a un espacio de nombres, que permiten a los usuarios definir y compartir sus propios esquemas.

Un esquema RDF define los términos que puede utilizar un documento RDF. Además, para cada término, define un significado específico. Para facilitar la reutilización de términos, RDFS hace uso de los Espacios de Nombres, pudiendo utilizarse términos de distintos esquemas de forma no ambigua. En general, una descripción de un recurso puede incluir sentencias con propiedades de distintos esquemas. El esquema RDFS

puede contener también restricciones sobre los tipos de Recursos y Valores dentro de las sentencias.

Las principales primitivas de RDFS pueden agruparse en Clases, Propiedades y Restricciones, a continuación se incluye una breve descripción de cada una de ellas.

2.1.3.2.1 Clases

RDFS permite la definición Recursos, Clases y Propiedades. El Recurso es una clase genérica de la cual heredan todos los objetos descritos en un documento RDF. Las Clases y Propiedades son similares a las definidas en la orientación a objetos aunque, principalmente las últimas, no son definidas como integrantes de la clase sino que se definen en función del Recurso y el Valor que asocian. De este modo, algo impensable en orientación a objetos, como es el que un atributo pueda no pertenecer a una clase (o pertenecer a más de una), en RDF(S) sí puede darse. Para este propósito, las principales primitivas que ofrece RDFS son:

- ***rdfs:Resource***: Representa la clase genérica en el modelo RDF Schema. Todo objeto descrito por expresiones RDF es un recurso.
- ***rdfs:Class***: Es subclase de *rdfs:Resource* y representa el concepto genérico de tipo o categoría, similar a la noción de clase en orientación a objetos.
- ***rdf:Property***: Es subclase de *rdfs:Resource* y representa un aspecto del recurso que se está describiendo, similar a la noción de atributo en orientación a objetos.

2.1.3.2.2 Propiedades

Las propiedades posibilitan expresar relaciones entre instancias y sus clases, o clases y sus superclases. Las relaciones entre propiedades también son permitidas, obteniéndose así una jerarquía de propiedades. Destacar que además es posible la implementación de herencia múltiple, lo que proporciona gran flexibilidad al modelo RDF, al permitirse que una clase pueda heredar propiedades de varias clases *padre*. La herencia facilita en gran medida la reutilización y especialización de los términos definidos en otros esquemas. Las principales propiedades disponibles en RDFS son:

- ***rdf:type***: Es una subclase de *rdf:Property* y denota que un recurso es instancia de una clase, poseyendo todas sus características. Un recurso puede ser instancia de más de una clase.
- ***rdfs:subClassOf***: Es una subclase de *rdf:Property* y denota la relación de subclase/superclase entre dos clases. Esta propiedad es la base para la especificación de la herencia múltiple, y tiene característica de transitividad.
- ***rdfs:subPropertyOf***: Es una subclase de *rdf:Property* y denota la relación de especialización entre dos propiedades, posibilitando la definición de una jerarquía de propiedades.

2.1.3.2.3 Restricciones

Mediante estas primitivas es posible restringir las propiedades de un recurso, concretamente, estableciendo cuál es el recurso al que se puede aplicar la propiedad y sobre qué rango de datos puede tomar valores dicha propiedad. Las principales primitivas de restricciones son:

- ***rdfs:domain***: Es una instancia de la clase *rdfs:ConstraintProperty* y especifica a qué recurso se aplica una propiedad.
- ***rdfs:range***: Es una instancia de la clase *rdfs:ConstraintProperty* y restringe los valores (u objetos) que una propiedad puede asumir.

Tal y como se ha descrito, RDF(S) tiene capacidad para representar, en resumen, predicados binarios, jerarquías de clases y propiedades, así como restricciones de dominio y de rango para propiedades. No obstante, la expresividad de RDF(S) está limitada a la hora de modelar ontologías, dado que carece de capacidad para representar axiomas, propiedades de propiedades (por ejemplo, transitividad) o condiciones de pertenencia a una clase. Para poder representar los elementos necesarios para la representación de ontologías es necesario utilizar lenguajes con mayor expresividad. Entre ellos el recomendado por el W3C es OWL, descrito en el siguiente apartado.

2.1.3.3 OWL Web Ontology Language

El Lenguaje de Ontologías Web OWL (*OWL Web Ontology Language*) es una especificación del W3C (Bechhofer, S. et al., 2004) desarrollada para la definición de ontologías en la Web. OWL está especificado sobre RDF(S) y, por tanto, puede ser representado como un documento RDF(S)

Para poder representar ontologías, OWL incorpora un conjunto de primitivas que proporcionan mayor expresividad que RDF(S). Las principales mejoras de OWL frente a RDF pueden resumirse en:

- **Ámbito local de propiedades**: Permite definir una propiedad con un ámbito local a una clase, a diferencia de RDF(S) donde el ámbito es global.
- **Clases relacionadas**: Permite definir clases donde la pertenencia, *instancia*, a una de ellas implica una relación con respecto a otra/s, por ejemplo: *igualdad de clases*, *clases disjuntas* o *clases complemento*.
- **Combinación booleana de clases**: Permite definir clases como la unión, intersección o complemento de otras clases. El conjunto de instancias de la clase agregada será el resultado de la aplicación de la regla al conjunto de instancias de cada una de las clases combinadas.
- **Restricciones de cardinalidad**: Posee mayor capacidad para expresar restricciones de cardinalidad en los valores de una propiedad, por ejemplo: *al menos uno* o *exactamente dos*.

- Características de propiedades: Proporciona un mayor conjunto de tipos de propiedades, así como características especiales, por ejemplo: *transitividad*, *unicidad* o *invertibilidad*.
- Clases enumeradas: Permite la definición de clases como la enumeración de los individuos que la componen.

Un lenguaje para expresar ontologías debe atender a dos objetivos fundamentales: máxima expresividad y soporte a razonamiento. El primero proporciona la capacidad para representar axiomas, conceptos y relaciones complejas; el segundo la capacidad de razonamiento eficiente, es decir, poder desarrollar un razonamiento completo y decidible⁷. Como puede intuirse, la conjunción de ambos objetivos es difícil lograr, por este motivo, el grupo de trabajo del W3C decidió definir tres sublenguajes⁸ de OWL con distintas capacidades de representación.

- **OWL Lite:** Es el sublenguaje con menor capacidad expresiva. Básicamente está pensado para usuarios que necesiten jerarquías de clasificación y restricciones simples. Es una buena opción para la representación de tesauros y taxonomías.
- **OWL DL:** Es el sublenguaje con capacidad expresiva intermedia. Proporciona la mayor capacidad expresiva garantizando que el razonamiento es computacionalmente completo y decidible. Básicamente, OWL DL posee todas las primitivas que OWL Full aunque con algunas restricciones en la combinación con RDF(S), básicamente, los constructores de OWL y RDF(S) no pueden aplicarse entre ellos, por ejemplo: no se permite que una clase sea instancia de otra. El acrónimo *DL* de OWL DL, proviene de su correspondencia con la Lógica Descriptiva (*Description Logic*), lógica en la que se basan los fundamentos de OWL.
- **OWL Full:** Es el sublenguaje con mayor capacidad expresiva. Está diseñado para usuarios que precisen la máxima expresividad y la libertad sintáctica de RDF, a cambio, no se garantiza que el razonamiento sea completo ni decidible. Básicamente, elimina las restricciones de OWL DL, permitiendo cualquier combinación de las primitivas OWL con RDF(S).

Dentro de la estructura de la Web Semántica, el lenguaje OWL es el estándar propuesto para implementar ontologías en la Web y especificar axiomas de la capa lógica, puesto que permite la descripción semántica del conocimiento y proporciona soporte a razonamiento. Cabe destacar que, desde octubre de 2009 existe una nueva versión de OWL, denominada OWL 2 (World Wide Web Consortium, 2009c). En esta nueva

⁷ El razonamiento completo garantiza que todas las conclusiones serán computables. El razonamiento decidible garantiza que el razonamiento finalizará en tiempo finito. No obstante, no se garantiza que el razonamiento sea decidible a tiempo. En función del dominio del problema, es posible que el tiempo necesario para llevar a cabo un razonamiento exceda del tiempo en el que los resultados son válidos. Por ejemplo, si llevar a cabo una predicción del tiempo para la próxima semana implica un tiempo de procesamiento de un mes, evidentemente el razonamiento no es válido.

⁸ También llamados *especies*, del término anglosajón *species*.

versión se revisan algunas carencias del lenguaje como limitaciones en la expresividad, problemas sintácticos o problemas semánticos (Grau, B. et al., 2008), detectadas en la aplicación del lenguaje al modelado de distintos dominios del conocimiento. En la presente propuesta se ha preferido el uso de OWL 1, dado que el soporte a OWL 2 aún no está lo suficientemente extendido, debido a su relativa novedad.

2.1.4 Situación actual de la Web Semántica

Desde su creación, la Web ha obtenido un gran éxito permitiendo el intercambio mundial de información. No obstante, a día de hoy, aún posee importantes limitaciones a mejorar. Los usuarios humanos requieren de la Web una conceptualización que soporte interacción semántica, potenciando la usabilidad de los sitios, satisfaciendo así la necesidad de convertir la información en conocimiento. Los sistemas informáticos, por su parte, precisan de una mayor inteligencia y autonomía (Codina, L.; Rovira, C., 2006). Dotar de significado a la información, es un requisito fundamental para dar respuesta a las necesidades de ambos usuarios y para lograr una Web Semántica real.

Según Sanchez y Fernández (Sánchez, L.; Fernández, N., 2005), los sistemas basados en conocimiento serán los encargados de proporcionar servicios en la futura Web Semántica. Estos sistemas ya han sido estudiados por los investigadores en Inteligencia Artificial y se basan en dos componentes fundamentales: una base de conocimientos y un motor de inferencia. El primero establece el conjunto de hechos conocidos por el sistema, el segundo es el encargado de realizar el razonamiento sobre la base de hechos y la inferencia de nuevo conocimiento. En el ámbito de la Web Semántica, la base de hechos de los futuros sistemas será generada mediante anotaciones que establezcan el significado de la información contenida en las páginas semánticas. De este modo, las aplicaciones podrán hacer uso de este conocimiento para dar solución a problemas complejos, por ejemplo combinar información de fuentes heterogéneas. Para llevar a cabo el razonamiento sobre esta base de hechos, será necesaria la existencia motores de inferencia que operen sobre un modelo del dominio que incluya: el vocabulario de los conceptos, las propiedades que los relacionan y las reglas que lo gobiernan. En este punto entran en juego las ontologías.

Por lo expuesto hasta ahora, la anotación de los recursos Web juega un papel fundamental en la consecución de la Web Semántica. De este modo, es preciso agregar metadatos semánticos a los recursos, lo que comúnmente se conoce como *anotación semántica*. Este tema conforma un campo importante de investigación en la Web Semántica, encontrándose multitud de enfoques para su realización que, atendiendo al grado de automatización (Uren, V. et al., 2006), pueden clasificarse en:

- **Sistemas de Anotación Manuales:** Donde, típicamente, se proporciona al usuario un interfaz que le permite agregar anotaciones a recursos Web, incluyendo referencias a conceptos modelados en ontologías. Como ejemplos de este tipo de sistemas pueden destacarse: *Annotea* (Kahan, J. et al., 2001), *SHOE Knowler Annotator* (SHOE, 2000), *SMORE (Semantic Markup, Ontology and RDF Editor)* (Kalyanpur, A. et al., 2005) o *BIOSEM* (Fuentes-Lorenzo, D.; Morato, J.; Gómez, J.M., 2009).

- Sistemas de Anotación Automáticos y Semi-Automáticos: Haciendo uso de técnicas de procesamiento de lenguaje natural (PLN), extraen referencias en el texto que coincidan con conceptos modelados en ontologías, para ello suelen requerir corpus de entrenamiento para el establecimiento de los patrones de reconocimiento. Como ejemplos de este tipo de sistemas, se encuentran: *AeroDAML* (Kogut, P.; Holmes, W., 2001), *SemTag* (Dill, S. et al., 2003), *SCREAM* (Handschuh, S.; Staab, S.; Ciravegna, F., 2002), *MELITA* (Ciravegna, F. et al., 2002) o *PANKOW* (Cimiano, P.; Handschuh, S.; Staab, S., 2004).

No obstante, existen aún inconvenientes importantes (Gómez-Pérez, A. et al., 2002), como:

- La falta de acuerdo, en el ámbito de la semántica, sobre qué características de las palabras se deben anotar.
- La falta de un conjunto exhaustivo de categorías semántico-cognitivas básicas.
- La falta de herramientas con interfaces usables.
- Las limitaciones tecnológicas actuales, por ejemplo, la dificultad de obtener automáticamente páginas compactas, legibles y verificables.

A estos inconvenientes, se añaden (Bonilla, S., 2007):

- El peligro de obsolescencia que sufre la tecnología de anotación, provocado por la continua evolución de los estándares.
- El enorme reto que supone la anotación semántica de la totalidad de la Web, dada su magnitud y mutabilidad.

Así como los expuestos recientemente por Tim Berners-Lee (Jackson, J., 2009):

- Falta de formación tecnológica por parte de los administradores de sitios, concretamente en el lenguaje RDF.
- Carencias en la definición de recursos. Al no especificarse las URIs de los recursos, por parte los administradores de sitios, se imposibilita la recuperación y procesamiento de información, de forma automática.

La importancia de estos inconvenientes explica el porqué, aún hoy, la Web Semántica no es una realidad y, lo que es peor, hacen dudar sobre si algún día llegará a serlo. Lo que sí se presenta como requisito fundamental es el desarrollo de sistemas de anotación más potentes y usables, que faciliten la anotación semántica, la reutilización de esquemas y potencien el desarrollo de un mayor número de recursos semánticos. Actualmente, existen dos nuevos enfoques para solucionar algunos de los inconvenientes anteriores:

1. Microformatos: Es la gran apuesta de *Technorati*, y básicamente consiste en

prescindir de las propuestas de Tim Berners-Lee y del W3C para incorporar semántica específica en pequeños fragmentos de código XHTML. Es una propuesta similar a lo que fue en los 90 las etiquetas Meta de HTML pero algo más ambicioso. Lógicamente su gran popularidad actual se debe a su simplicidad, y esta misma simplicidad limita su aplicabilidad futura. De cualquier modo, dejan claro que la Web Semántica tiene que encontrar métodos para simplificar su aplicación.

2. La Web 2.0: A finales de 2004, Dale Dougherty utiliza por primera vez el término *Web 2.0* durante una tormenta de ideas con Craig Cline, para desarrollar ideas para una conferencia. Al analizar las empresas que habían sobrevivido al derrumbamiento provocado por el reventón de la burbuja *puntocom* a finales de 2001, Dougherty observó que, en lugar de resentirse, la Web era más popular que nunca y que aquel suceso había sido superado por un conjunto de empresas que parecían tener ciertas características comunes. Estas características son las que establecen los principios de la Web 2.0, definidos por Tim O'Reilly en 2005 (O'Reilly, T., 2005), y que pueden resumirse en:

- El uso de la Web como plataforma.
- El aprovechamiento de la inteligencia colectiva, haciendo de la participación de los usuarios un aspecto clave. La importancia de los blogs, aprovechando la inteligencia colectiva a modo de filtro de contenidos, juegan un papel importante en los resultados de los buscadores amplificando la visibilidad de ciertos recursos dentro de la comunidad bloguera.
- La posesión de los datos y los derechos de uso son un aspecto fundamental en competición por ganar una masa crítica de usuarios, valga como ejemplo *Google Maps*. Además, existe una preocupación creciente de los usuarios por la privacidad de sus datos personales. Ambos aspectos llevan a pensar en un futuro movimiento de *datos libres*, similar al producido con el software privativo y la aparición del *software libre*.
- Un cambio en el ciclo de desarrollo de software tradicional, caracterizado por la entrega de versiones estables y “duraderas” en cuanto a funcionalidad, necesario para poder atender requisitos con una estabilidad muy baja (días). Este nuevo tipo de software se caracteriza por entregarse en forma de servicio en línea, en lugar del “clásico” producto. Como aspecto clave para afrontar este nuevo tipo de desarrollo, es necesario tratar a los usuarios como co-desarrolladores, al igual que se realiza en el desarrollo de aplicaciones de código abierto.
- La potenciación de los modelos de programación ligeros, que permiten reducir el acoplamiento entre sistemas y tienden más a un enfoque sindicado de la información, frente al enfoque coordinado de la misma. También destaca el hecho de diseñar las aplicaciones para facilitar las posibilidades de reutilización, lo que no es de código abierto está débilmente protegido, facilitándose así el enriquecimiento de los servicios con enfoques provenientes de los usuarios.
- El desarrollo de software multiplataforma, en lugar de centrado en el PC, con el objetivo de llevar la Web también a dispositivos portátiles como iPod's,

móviles, etc.

- La mejora en la calidad de los servicios para usuarios incorpora mejoras en la facilidad de uso, como aspecto fundamental, así como nuevas funcionalidades dirigidas al aprovechamiento de la portabilidad de los dispositivos móviles.

Si bien ambas perspectivas, la Web Semántica y la Web 2.0, han sido vistas como competidoras en la futura evolución de la Web, lo cierto es que cada vez cobra más fuerza la visión de enfoques complementarios (Ankolekar, A. et al., 2007), (Fumero, A.; Roca, G., 2007) y (Morato, J. et al., 2008). Ambas tecnologías podrán salvar sus limitaciones cooperando y complementándose mutuamente. De este modo en los últimos años se están generando servicios con un punto de vista centrado en usuarios y comunidades, basados en lenguajes y tecnologías ontológicos. Como ejemplos de esta complementariedad están las aplicaciones *Tabulator* (World Wide Web Consortium, 2007), o *Piggy Bank* (Massachusetts Institute of Technology, 2008) del proyecto *Simile* (Massachusetts Institute of Technology Libraries, 2008a); esta última, mediante un plugin para *Firefox*, permite poner etiquetas en lenguaje libre (Web 2.0) y con versión en RDF (Web Semántica), e incluso compartir estos documentos mediante el *PiggyBank*. Las RSS o el proyecto *CoolWikNews* (Fuentes, D.; Gómez, J.; García, C., 2008) representan apuestas similares. El plan de actuación para las herramientas de la Web 2.0 de la Universidad de Edimburgo (Adie, C., 2007), es una de las propuestas pioneras en el ámbito de las universidades, enfocada a la docencia, el aprendizaje y la investigación.

2.2 Metadatos

2.2.1 Esquemas de metadatos

Los metadatos son descripciones que facilitan la recuperación, utilización o gestión de recursos de información (Hodge, G., 2001). Así, los metadatos pueden ser utilizados para organizar recursos electrónicos o facilitar la interoperatividad. Según Ruixin (Ruixin, Y. et al., 2001), creador del buscador de metadatos *DIMES*, es necesario utilizar documentos XML para asegurar esta interoperatividad.

Los metadatos suelen estar agrupados en conjuntos de elementos denominados esquemas o vocabularios de metadatos. Cada esquema de metadatos se diseña con una finalidad concreta y se define dentro de un contexto o espacio de nombres concreto. Dentro de dicho contexto, cada uno de los elementos del esquema posee una definición única. Es habitual el uso de varios espacios de nombres dentro de un mismo esquema, lo que permite el uso de los elementos definidos en otros esquemas, identificados mediante un nombre cualificado (*QName*), esto es: un identificador del esquema, seguido de dos puntos y el nombre del elemento en dicho esquema. Esta característica hace posible la reutilización y difusión de esquemas en la Web.

Según Duval (Duval, E. et al., 2002), los principios que deben regir el diseño de metadatos son: Modularidad, Extensibilidad, Refinamiento y Plurilingüismo. Los metadatos suelen llevar una semántica asociada (semántica del esquema), unas reglas de contenido y una sintaxis. En la práctica, una especificación de metadatos puede estar completada con una serie de pautas de cómo utilizar ese vocabulario para describir un

corpus concreto, normalmente especificada en un documento independiente al esquema en sí y pensado para ser interpretado por el usuario. Debe destacarse también que, si bien se especifica una semántica del esquema, se trata de un conjunto de definiciones, para cada uno de los elementos, establecidas por la entidad responsable de la creación y mantenimiento del esquema. De este modo, se puede establecer que dicha semántica no es compartida, en el sentido de que es establecida para el esquema concreto, lo cual como veremos posteriormente, dificulta su reutilización, causa problemas de interoperatividad y correspondencia entre elementos de distintos esquemas.

Sobre cómo almacenar los metadatos hay tres grandes posibilidades (Yau, H.S.; Hawker, J.S., 2004):

- En el mismo documento, por ejemplo añadiendo espacios de nombres en etiquetas META o en RDF.
- En un documento independiente que enlace a los documentos pertinentes, por ejemplo con RDF o con el estándar Topic Maps (International Organization for Standardization, 2002).
- Mediante un repositorio de datos.

Algunos vocabularios de metadatos especialmente relevantes por su popularidad, en función del número de documentos obtenidos en Swoogle (Swoogle, 2007b), son:

- Dublin Core (DC) (Dublin Core Metadata Initiative, 2003). Desarrollado y mantenido por el DCMI (Dublin Core Metadata Initiative, 2010) (*Dublin Core Metadata Initiative*), es un vocabulario de 15 elementos desarrollado para la descripción de recursos. Por su importancia en relación con el presente trabajo, en el apartado 2.2.2 *Esquemas de metadatos seleccionados* se incluye una descripción más detallada del esquema.
- Friend of a Friend (FOAF) (FOAF-project, 2010). Vocabulario, procesable por computadoras, sobre información personal y relaciones interpersonales. Los objetivos de FOAF son facilitar: el uso compartido de información sobre personas y sus actividades, la transferencia de información entre servidores Web, y la extensibilidad, combinación y reutilización automáticas vía Web. Por su importancia en relación con el presente trabajo, en el apartado 2.2.2 *Esquemas de metadatos seleccionados* se incluye una descripción más detallada del esquema.

Un problema de este espacio de nombres es que carece de un vocabulario estable unido al hecho de que permite cualquier extensión. El enorme número de documentos FOAF se explica porque, en su mayoría, son generados automáticamente en sitios Web que trabajan con blogs.

- Really Simple Syndication (RSS) (Harvard University, 2003). Es un vocabulario que establece el formato para syndicar contenidos de servidores Web. Define un conjunto de metadatos multipropósito que permiten la descripción de *canales* de información (noticias, opinión, etc.). Mediante los llamados *agregadores* un

usuario puede suscribirse a canales y recibir las noticias que éstos incluyan.

- Text Encoding Initiative (TEI) (The Text Encoding Initiative, 2010). Estándar que proporciona las pautas para la representación en línea de textos literarios y lingüísticos, con fines docentes, de investigación y preservación.
- Encoded Archival Description (EAD) (EAD, 2009). Vocabulario desarrollado para facilitar la descripción de inventarios, registros y catálogos de entidades como archivos, bibliotecas o museos; haciéndolos procesables para computadoras y permitiendo su uso en línea.
- VRA Core Categories (Visual Resources Association, 2010). Vocabulario que proporciona una organización categorizada para la descripción de material gráfico.
- VCard (Dawson, F.; Howes, T., 1998a). Vocabulario para la gestión de información de contactos. Además de la información básica como nombre, teléfonos, dirección de correo, etc., proporciona soporte para información binaria, por ejemplo, fotografías. Por su importancia en relación con el presente trabajo, en el apartado 2.2.2 *Esquemas de metadatos seleccionados* se incluye una descripción más detallada del esquema en su versión desarrollada por el W3C (World Wide Web Consortium, 2010b).
- Meta Vocabulay (MVCB) (Semantic Web Abreement Group, 2001). Vocabulario utilizado en blogs para la representación de metadatos relacionados con aspectos de administración, generación y gestión de las noticias.
- Simple Knowledge Organization System (SKOS) (World Wide Web Consortium, 2005). Vocabulario para la representación de Esquemas Conceptuales y sus contenidos, entre otros: tesauros, esquemas de clasificación, taxonomías, etc.
- Metadata Object Description Schema (MODS) (Library of Congress, 2005a). Vocabulario para la representación de registros MARC 21 y para la generación de registros de descripción.
- Metadata Authority Description Schema (MADS) (Library of Congress, 2005b). Relacionado con el anterior, representa el formato de Autoridad de MARC 21. De este modo, permite incluir información sobre agentes (personas y organizaciones), eventos y términos (conceptos, géneros, etc.).
- Zthes (Taylor, M., 2006). Define un perfil para tesauros según la especificación del protocolo Z39.50. El objetivo es permitir la navegación y consulta de tesauros en formato XML.
- Multimedia Content Description Interface (MPEG7) (International Organization for Standarization, 2004). Vocabulario para la descripción de contenido multimedia, dotándolo de semántica mediante un proceso de anotación. Entre la información que se puede añadir al contenido está la descripción del proceso de

creación, el uso del contenido, características de su almacenamiento, estructura, realidad representada, etc.

Para analizar el uso de los vocabularios de metadatos en la Web, en junio de 2004, Ding (Ding, L. et al., 2005), mediante Swoogle (Swoogle, 2007b), determinó el conjunto de espacios de nombres, asociados a vocabularios de metadatos, más utilizados. Los datos en junio de 2007 eran: DC 1.559.153, MCVB 420.971, RSS 812.524, FOAF 700.046 y vCard 188.752 documentos, respectivamente. Debido al crecimiento en su uso y en relación con la propuesta, se incluyeron en la evaluación OWL con 141.181 y RDF con 1.553.250 documentos, respectivamente. Los datos actualizados en mayo de 2010 muestran un crecimiento en el número de documentos, más del doble en algún caso, que hacen uso de los anteriores espacios de nombres, concretamente: DC 2.727.822, MCVB 515.217, RSS 1.242.925, FOAF 1.148.709, vCard 196.931, OWL 359.384 y RDF 2.381.832.

En relación al presente trabajo, son especialmente relevantes los siguientes conceptos relacionados con los esquemas de metadatos:

- Perfiles de aplicación (Heery, R.M.; Patel, M., 2000). Los perfiles de aplicación se definen como esquemas definidos a partir de elementos obtenidos a partir de otros esquemas. Con este objetivo, los elementos son incluidos mediante referencias al espacio de nombres en los que se definen. Los perfiles de aplicación son combinados y optimizados por los desarrolladores de una determinada aplicación, de este modo permiten el uso de esquemas estándar adaptado al contexto específico de la aplicación.

Los perfiles de aplicación poseen un conjunto de características distintivas:

- Los elementos incluidos en el perfil de aplicación pueden provenir de uno o varios espacios de nombres. La principal restricción es que los espacios de nombres tienen que existir.
- El perfil de aplicación no puede incluir elementos nuevos. En relación con la característica anterior, los elementos pueden provenir de cualquier esquema siempre que exista, es decir, no pueden definirse nuevos elementos que no pertenezcan a un espacio de nombres existente. Si el desarrollador desea crear nuevos elementos, deberá definir un nuevo espacio de nombres y su correspondiente esquema, haciéndose responsable de su declaración y mantenimiento.
- En los perfiles de aplicación se pueden especificar los esquemas y valores permitidos. En ocasiones puede tener interés la definición de los valores permitidos para un determinado elemento. Del mismo modo, pueden definirse los vocabularios sobre los que los elementos deben tomar sus valores, por ejemplo: formatos de fecha.
- Los perfiles de aplicación pueden refinar las definiciones de los elementos originales, de este modo, es posible adaptar el elemento para su uso específico en el contexto de la aplicación para la que se desarrolla el perfil.

La principal restricción a esta característica es que el refinamiento debe mantener la coherencia con la semántica original del elemento pudiéndose, únicamente, redefinir la semántica de forma más específica.

Los perfiles de aplicación proporcionan, en resumen, un modo de compartir y reutilizar esquemas a nivel de elemento. Evitan la necesidad de desarrollar esquemas ad-hoc para aplicaciones, permitiendo la definición de esquemas a partir de términos heterogéneos. Además, posibilitan la especialización del uso de esquemas mediante la definición de los elementos incluidos y los valores que pueden tomar, y, aunque con limitaciones, a través de la especialización de la semántica de los elementos originales.

- Registros de metadatos. Son sistemas basados en repositorios, que tienen el fin de gestionar, recuperar, referenciar y reutilizar vocabularios de metadatos existentes. Entre la información que estos sistemas proporcionan sobre los vocabularios que gestionan se incluye: la definición, origen y localización del recurso.

El estándar *ISO 11179: Metadata Registries* (International Organization for Standardization, 2005) define un conjunto de recomendaciones para la especificación y estandarización de elementos de datos. Como objetivos, trata la semántica de los datos, la representación de los elementos y el registro de las descripciones de dichos datos. Además, cubre aspectos de gestión de los vocabularios almacenados así como una arquitectura de referencia para el desarrollo de los sistemas que implementan los registros de metadatos.

Algunos proyectos a destacar sobre registros de metadatos son: el proyecto *Schemas*, para RDF(S) y espacios de nombres relacionados con proyectos de la UE (SCHEMAS, 2003); el *Open Metadata Registry*, que recopila metadatos relacionados con el *Dublin Core Metadata Initiative* (Dublin Core Metadata Initiative, 2008a); otros registros son el *Department Defense Metadata Registry* (Defense Information Systems Agency, 2010) o el *US Environmental Data Registry* (U. S. Environmental Protection Agency, 2010).

Debe destacarse que, si bien los registros de metadatos pueden incorporar la semántica de los elementos que gestiona, dicha semántica se define y mantiene de forma local. Es decir, el modelo de datos propuesto dentro del marco de la ISO 11179 soporta el almacenamiento de la semántica de los elementos y esquemas, pero la implementación del modelo se realiza en el ámbito del sistema. Si bien puede consultarse vía Web, su gestión está limitada dentro del sistema que proporciona el registro de metadatos.

- Por último, y también en relación con los esquemas de metadatos, hay que hacer notar las carencias que se producen en cuanto a la reutilización de términos. Como prueba de ello, basta con fijarse en términos que son definidos de forma reiterativa en múltiples esquemas, en lugar de reutilizar los definidos con anterioridad en otros esquemas. Un ejemplo de estos términos puede ser el término *narrower* que permite relacionar un término con otros más específicos en cuanto a significado. Puede encontrarse la definición de este término en

multitud de esquemas cuando, una vez definido, podría haberse reutilizado en los vocabularios posteriores. Es más, en la mayoría de los casos, la semántica del término común a los distintos esquemas no coincide exactamente, cuando deberían ser definiciones no ambiguas y equivalentes. Como ejemplos de esquemas con términos comunes puede analizarse el solapamiento entre *SKOS* (World Wide Web Consortium, 2005) y *PSI-thesaurus* (PSI Metadata, 2006). La carencia en la reutilización evidencia la necesidad de recursos que permitan la búsqueda de esquemas y faciliten su uso. La falta de consenso en la definición de los términos, evidencia la necesidad de establecer una semántica común y consensuada, a través de recursos que permitan su comprobación y reutilización, por ejemplo repositorios semánticos compartidos vía Web.

2.2.2 Esquemas de metadatos seleccionados

Se incluye en este apartado el conjunto de esquemas seleccionado para ejemplificar la propuesta. Concretamente, para cada uno de ellos, se incluye una descripción y resumen de sus principales características.

Destacar que el conjunto de esquemas ha sido generado en base a los criterios de selección, y según el proceso, descritos en el apartado 3.1.2 *Búsqueda y selección de esquemas*. A partir de estos requisitos, el conjunto de esquemas seleccionados ha sido: *DCMI Dublin Core Elements Set* (dcelements) (Dublin Core Metadata Initiative, 2008b), *W3C vCard Ontology* (vcard-rdf) (World Wide Web Consortium, 2010b), *Friend of a Friend* (foaf) (Brickley, D.; Miller, L., 2010), *Description of a Career* (doac) (Parada, R., 2008), *Description of a Project* (doap) (Dumbill, E., 2008) y *Personal Information Markup* (pim) (Berners-Lee, T., 2007).

A continuación se muestra una breve descripción de cada esquema seleccionado.

Nombre	Dublin Core Element Set v1.1
Descripción	El esquema dcelements proporciona URIs para un conjunto de 15 elementos definidos para descripción de recursos bibliográficos
Espacio de nombres	http://purl.org/dc/elements/1.1/
Localización	http://purl.org/dc/elements/1.1/
Servidor	http://dublincore.org/documents/dcmi-terms/#H2
Entidad responsable	DCMI Usage Board
Dirección de contacto	dc-usage@jiscmail.ac.uk
Número de elementos	15

Número de documentos 2.727.822
(fuente: swoogle)

Tabla II-1: Resumen del esquema de metadatos Dublin Core Element Set v1.1

Nombre	vCard
Descripción	Vocabulario expresado en RDF que se corresponde con el perfil de tarjeta de negocio electrónica, definida por la RFC 2426 en su versión 3.0 (Dawson, F.; Howes, T., 1998b).
Espacio de nombres	http://www.w3.org/2001/vcard-rdf/3.0#
Localización	http://www.w3.org/2001/vcard-rdf/3.0
Servidor	http://www.w3.org/TR/vcard-rdf
Entidad responsable	Semantic Web Interest Group
Dirección de contacto	semantic-web@w3.org
Número de elementos	59
Número de documentos (fuente: swoogle)	196.931

Tabla II-2: Resumen del esquema de metadatos vCard

Nombre	Friend Of A Friend (FOAF)
Descripción	El proyecto Friend of a Friend (FOAF) consiste en la creación de páginas digitales iniciales con la descripción de personas, los enlaces entre las mismas y sus actividades.
Espacio de nombres	http://xmlns.com/foaf/0.1/
Localización	http://xmlns.com/foaf/0.1/index.rdf
Servidor	http://www.foaf-project.org/
Entidad responsable	Dan Brickley
Dirección de contacto	danbri@rdfweb.org
Número de elementos	74
Número de documentos (fuente: swoogle)	1.148.709

Tabla II-3: Resumen del esquema de metadatos FOAF

Nombre	Description Of A Career (DOAC)
Descripción	DOAC es un vocabulario de metadatos desarrollado para la descripción de currículos personales.
Espacio de nombres	http://ramonantonio.net/doac/0.1/
Localización	http://ramonantonio.net/doac/0.1/doac.rdfs
Servidor	http://ramonantonio.net/doac/
Entidad responsable	Ramon Antonio Parada
Dirección de contacto	rap@ramonantonio.net
Número de elementos	33
Número de documentos (fuente: swoogle)	19

Tabla II-4: Resumen del esquema de metadatos DOAC

Nombre	Description Of A Project (DOAP)
Descripción	DOAP es un vocabulario de metadatos desarrollado para la descripción de proyectos software de código abierto.
Espacio de nombres	http://usefulinc.com/ns/doap#
Localización	http://usefulinc.com/ns/doap
Servidor	http://usefulinc.com/ns/doap
Entidad responsable	Edd Dumbill
Dirección de contacto	edd-web@usefulinc.com
Número de elementos	50
Número de documentos (fuente: swoogle)	3.289

Tabla II-5: Resumen del esquema de metadatos DOAP

Nombre	Personal Information Markup (PIM)
Descripción	PIM es un vocabulario de metadatos desarrollado para la descripción de información personal.
Espacio de nombres	http://www.w3.org/2000/10/swap/pim/contact#
Localización	http://www.w3.org/2000/10/swap/pim/contact
Servidor	http://www.w3.org/2000/10/swap/
Entidad responsable	Dan Conolly
Dirección de contacto	connolly@w3.org
Número de elementos	58
Número de documentos (fuente: swoogle)	35.390

Tabla II-6: Resumen del esquema de metadatos PIM

2.2.3 Interoperatividad

El término interoperatividad⁹ ha sido definido como la capacidad que tienen algunos sistemas para intercambiar y utilizar información procedente de otro sistema diferente (American Library Association, 2010). La cuestión no afecta solamente a las diferencias entre estructuras de datos, sino también a la heterogeneidad de las semánticas. Así, sólo es posible la interoperatividad cuando existen gramáticas y significados compartidos entre distintos vocabularios (Díaz Ortuño, P.M., 2003). El significado compartido entre diferentes sistemas suele implicar una correspondencia entre los vocabularios de cada sistema, lo cual puede llevar aparejado un alto grado de subjetividad y puede ocasionar una pérdida de información (SCHEMAS, 2003).

Según Hodge (Hodge, G., 2001) la interoperatividad de metadatos es la capacidad de intercambio de datos entre diferentes plataformas, estructuras de datos e interfaces, con una mínima pérdida de contenido y funcionalidad. Los mecanismos más comunes para abordar la interoperatividad son los esquemas de metadatos, el uso de protocolos de transferencias comunes y los sistemas de correspondencia (*crosswalks*) entre vocabularios.

⁹ Frecuentemente se traduce el término interoperability por el calco interoperabilidad en lugar de interoperatividad. Ambos términos son frecuentemente utilizados en la documentación técnica relativa al dominio. El segundo tiene en español el significado de 'capacidad de varios elementos para funcionar bien juntos'. En esta propuesta se ha preferido el uso del segundo término al provenir del término *operatividad* que, a diferencia de *operabilidad*, si tiene acepción en el diccionario de la Real Academia de la Lengua.

Hunter (Hunter, J., 2001) argumenta que los principales escenarios que precisan de interoperatividad son: los motores de búsqueda que recuperan metadatos heterogéneos; la integración y fusión de descripciones complementarias y parcialmente redundantes; y la adaptación a distintos perfiles de usuarios. Según esta autora, intentar relacionar manualmente vocabularios de metadatos uno a uno no es eficiente. Un indicador de la importancia de esta materia en relación con los esquemas de metadatos son las dos propuestas ISO que están en redacción: ISO 19763: *Framework for Metamodel Interoperability* (International Organization for Standardization, 2010) e ISO 20944: *Metadata Registry Interoperability and Bindings (MDRIB)* (International Organization for Standardization, 2009).

Algunos ejemplos de interoperatividad son:

- El protocolo Z39.50 (American National Standards Institute. National Information Standards Organization, 1992) para búsquedas distribuidas.
- *Open Archives Initiative* (Open Archives, 2008) protocolo con arquitectura cliente-servidor para recuperar documentos mediante metadatos (Hodge, G., 2001).
- *Cooperative Online Resource Catalog* (Online Computer Library Center, 2010a) para catalogación .
- *Interoperability of Data in E-commerce systems* (Indecs, 2009) correspondencia entre vocabularios para la facilitar la Interoperatividad entre aplicaciones de comercio electrónico.

Destaca especialmente el proyecto *Harmony* (Hunter, J., 2001). En este proyecto se propone la utilización de un modelo de datos (modelo ABC) para relacionar metadatos entre diferentes dominios. Las clases contenidas en Harmony son: recursos, sucesos, entradas y salidas, actos (agentes y roles), contexto (tiempo y lugar) y relaciones de sucesos. *MetaNet* (CES, 2003) es el recurso que complementa al anterior y que contiene relaciones entre metadatos de diferentes dominios.

2.2.4 Correspondencia entre Esquemas

Como se puede deducir del apartado anterior, uno de los sistemas para asegurar la interoperatividad es la correspondencia, bien sea entre vocabularios de metadatos o entre cualquier otro tipo de esquema.

La correspondencia de esquemas de datos (*schema matching*) consiste en relacionar los elementos de dos esquemas para que se correspondan semánticamente. Se adopta, frecuentemente, esta solución cuando se trabaja en integración de datos y esquemas, intercambio de información con bases de datos, resolución de consultas, *data warehouse* o comercio electrónico (Rahm, E.; Bernstein, P.A., 2001). Usualmente esta correspondencia se hace manualmente, pero existen abundantes iniciativas para su automatización. Rahm (Rahm, E.; Bernstein, P.A., 2001) argumenta que hacerlo manualmente es costoso, laborioso y está sujeto a posibles errores. En la práctica, este autor considera que las heurísticas por las que un elemento o metadato se traduce en

otro resultan difícilmente automatizables.

Un listado indicando estas correspondencias uno contra uno se puede encontrar en Day (Day, M., 2002). Uno de los mayores problemas a los que se enfrentan las correspondencias, según Howarth (Howarth, L.C., 2000), es que el 68,60% de los metadatos están representados en un único esquema, mientras que el 1,17% de los metadatos tienen una equivalencia exacta en otros esquemas. Es decir, que la mayoría de los elementos definidos en los esquemas de metadatos no poseen correspondencia en otros esquemas.

Antes de realizar la correspondencia, es necesario decidir si se relacionan elementos contra elementos o si también se relacionan elementos contra combinaciones de elementos, por ejemplo un elemento de un esquema contra un elemento con sus atributos y subelementos del otro esquema.

Además, la correspondencia se puede basar en un enfoque lingüístico, por ejemplo gracias al nombre del elemento, sus sinónimos, su descripción o el espacio de nombres. También puede seguirse un criterio sintáctico (estructural), por ejemplo utilizando las relaciones (con otros conceptos similares), multiplicidades o tipos de datos. En el caso del enfoque lingüístico se suelen utilizar diccionarios o tesauros para desambiguar los elementos, también se suele recurrir a lematizadores u otro tipo de normalizaciones terminológicas, por ejemplo para relacionar “NOJugadores” con “Número jugadores”.

Según Buscaldi (Buscaldi, D. et al., 2003), basando la correspondencia entre elementos en su similitud semántica, la similitud entre dos documentos XML depende de la similitud de las etiquetas que contienen, ya que las etiquetas son, en esencia, elementos semánticos. Es decir, calculando la similitud semántica de las etiquetas pueden recuperarse documentos XML. Con una idea similar Green (Green, S.J., 1998) propuso crear enlaces semánticos automáticamente bajo la premisa de que recursos que tratan sobre un mismo tema tienden a usar términos similares, y que esta similitud se podría computar gracias a un recurso léxico llamado *WordNet* (Princeton University, 2010).

WordNet (WN) es una herramienta libre de código abierto que contiene una red léxica del idioma inglés y que integra, entre otras relaciones, las de meronimia, generalización, sinonimia entre los distintos sentidos (*synsets*) de cada conjunto de sinónimos. En su versión 2.0 WN tiene 152.059 términos organizados en 115.424 *synsets* y 203.145 sentidos.

WN ha sido utilizado frecuentemente como recurso de desambiguación de conceptos vía densidad semántica o selección de conceptos (Morato, J. et al., 2004). Esta herramienta tiene una extensión denominada *EuroWordNet* (Amsterdam University, 1999) con una base de datos multilingüe que interrelaciona distintos idiomas. En cada uno de los idiomas se ha creado un WordNet independiente. Todos los WordNets se relacionan a través del *Inter-lingual Index* mediante sus *synsets*. El InterLingual Index es una interlingua independiente del idioma. Además, EuroWordNet utiliza una ontología de alto nivel y una ontología de dominio, ambas independientes del idioma e interrelacionadas con la interlingua. El proyecto, iniciado en el año 1999, tomó como referencia a WN 1.5, pero corrigiendo algunas de las deficiencias de esa versión, como la proliferación excesiva de sentidos, tratamiento de instancias y algunas inconsistencias

semánticas. Aunque en principio se trataba de un proyecto restringido a unas pocas lenguas europeas, actualmente se desarrollan proyectos para interconectar WordNets en otros idiomas. Dentro de los idiomas peninsulares, EuroWordNet está disponible en catalán, castellano, vasco y portugués (Universidad Politécnica de Cataluña, 2008). Minidir, es un léxico confeccionado para Senseval a partir de WN, pero intentando resolver la excesiva granularidad de sentidos de WN, es decir, sólo tiene los significados básicos (Taulé, M.; Martí, M.A., 2003).

En otra revisión sobre interoperatividad Zeng (Zeng, M.L.; Chan, L.M., 2004), diferenció entre:

- Interoperatividad entre idiomas, en los que se afronta los problemas descritos en la norma ISO de tesauros multilingües (International Organization for Standardization, 1976): equivalencias parciales, inexistencia del concepto, etc.
- Interoperatividad entre diferentes sistemas de organización del conocimiento: el mayor problema suele consistir en que distintos sistemas no suelen tener estructuras similares. Zeng (Zeng, M.L.; Chan, L.M., 2004) señala la gran presencia en este apartado de correspondencias entre sistemas de clasificación (por ejemplo, DDC (Online Computer Library Center, 2010a) o UDC (UDCC, 2009)) contra vocabularios controlados.

Entre las técnicas para relacionar distingos esquemas Zeng (Zeng, M.L.; Chan, L.M., 2004) identifica:

- Derivación/modelado: Donde uno de los vocabularios se genera a partir de la especialización o simplificación de otro más general. Un tipo de correspondencia entre vocabularios de metadatos por derivación es el denominado *crosswalk*. Consiste en la identificación de dos elementos mediante un conjunto de transformaciones y reglas de resolución (St. Pierre, M.; LaPlant Jr., W.P., 1998). En este tipo de correspondencia se tiene en cuenta no solamente los elementos, sino también la semántica y la sintaxis entre los dos vocabularios (Hodge, G., 2001). La dificultad para establecer la correspondencia entre los elementos estará asociada al grado de similitud entre los vocabularios, su granularidad y las reglas de aplicación de cada metadato (Peig Olivé, E., 2004). Existen recopilaciones de estas correspondencias (Massachusetts Institute of Technology Libraries, 2010), (UKOLN, 2002), modelos para su gestión en repositorios (Godby, C.J.; Young, J.A.; Childress, E., 2004) y servicios específicos que facilitan la traducción entre esquemas (Schwartz, C., 2009).
- Traducción/adaptación: Donde los términos del vocabulario provienen de la traducción de un vocabulario expresado en una lengua diferente.
- Satélites: Donde los vocabularios más específicos son tratados como satélites de una estructura más general. Se suelen utilizar para gestionar áreas de conocimiento especializadas.
- Correspondencia Directa: Consistente en establecer la equivalencia entre términos de distintos vocabularios controlados. La correspondencia directa

supone una labor intelectual complicada, por lo que se ha intentado proponer sistemas semiautomáticos. Uno de los grandes problemas de los enfoques automáticos es que no suelen tener en cuenta el contexto, como la estructura de los elementos o su interrelación con otros elementos.

- Concurrencia de Términos de diferentes vocabularios: Se establece a nivel de aplicación, por ejemplo en registros de metadatos, cuando se incluyen términos de distintos esquemas. En este caso, en lugar de establecerse la correspondencia a nivel de esquemas, se hace uso de estos conjuntos heterogéneos como relación de concurrencia, estableciéndose así una correspondencia entre esquemas.
- Lenguaje de Intercambio (*Switching*): Se utiliza para establecer la equivalencia entre términos de distintos vocabularios. Puede tratarse de un Lenguaje de intercambio o de un Esquema que es utilizado como intermediario.
- Enlace a través de una lista de unión temporal: Permite relacionar términos que no son conceptualmente equivalentes, pero que sí están relacionados lingüísticamente. Las relaciones no son persistentes sino que se establecen como respuesta, por ejemplo a una consulta, en base a criterios como correspondencias entre palabras (todo o parte).
- Enlace a través de un protocolo servidor de tesauros: Establece relaciones no persistentes entre elementos según el resultado de una consulta realizada a un tesoro.
- Mediación: Permite hacer uso del conocimiento representado en los vocabularios integrando dicho conocimiento mediante el uso de una ontología común. De este modo es posible interactuar con la vista integrada, siendo el mediado el componente encargado de descomponer y especializar las operaciones para cada fuente específica (Wache, H. Vögele, T. Visser, U. Stuckenschmidt, H. Schuster, G. Neumann, H. Hübner, S., 2001).

A continuación se incluyen ejemplos de iniciativas de interés, clasificados por el grado de automatización del proceso de correspondencia:

- Correspondencia manual: Özel (Özel, S.A. et al., 2004), propone enlazar manualmente la semántica, elementos de DTD's contra conceptos (*topics*) de un Mapa Conceptual (*Topic Map*). Según esta propuesta, el mapa conceptual actuaría como índice semántico, identificando los conceptos y relaciones en un conjunto de recursos. Los enlaces a las ocurrencias, elementos de DTD's permitirían acceder a las ocurrencias de los conceptos en dichos recursos.
- Correspondencia semiautomática: Se han propuesto diferentes correspondencias semiautomáticas entre esquemas XML o basados en RDF, por ejemplo: *LSD* (*Learning Source Descriptions*) (Doan, A. et al., 2001) con validación por experto; *SKAT* (*Semantic Knowledge Articulation Tool*) (Mitra, P.; Wiederhold, G.; Kersten, M.L., 2000) para ontologías y con validación por expertos; *CUPID* (Madhavan, J. et al., 2001), utiliza recursos lingüísticos buscando términos con una grafía similar, salvando el tipo de datos y el dominio. Además existen

herramientas interactivas para alinear ontologías como *Chimaera* (Stanford KSL Network Services, 2001a) o *Prompt* (Musen, M.A.; Noy, N.F., 2001). En este último, el grado de acierto en los experimentos oscila entre el 60 y el 100%. Una metodología para mejorar la correspondencia entre ontologías se ha propuesto en *Semantic enrichment for improving systems interoperability* (Xiaomeng, S.; Hakkarainen, S.; Brasethvik, T., 2004). Para mejorar la semántica asociada a cada concepto, se analizan las extensiones de la ontología y los documentos asociados a cada concepto, lo que redundará en correspondencias semánticamente más precisas. El proyecto KRAFT utiliza una búsqueda de coincidencias mediante *WordNet* (Hakimpour, F.; Geppert, A., 2001).

- Correspondencia automática: La correspondencia automática tiene problemas comunes con la desambiguación semántica. La desambiguación (*WSD: Word Sense Disambiguation*) es uno de los principales retos de los sistemas de Procesamiento de Lenguaje Natural. En el certamen Senseval que evalúa objetivamente técnicas, métodos y sistemas de WSD, las evaluaciones, en las que se compara una anotación intelectual con una automática, no han conseguido superar el 80% de aciertos (Taulé, M.; Martí, M.A., 2003).

2.3 Ontologías

2.3.1 Introducción

Las ontologías responden a la necesidad de compartir, reutilizar, interpretar y aplicar la información uniformemente en distintos ámbitos. Para lograrlo, las ontologías proveen, no sólo una representación del conocimiento del dominio interpretable por personas y computadoras, también una definición formal y consensuada de conceptos que garantiza la interpretación correcta de dicho conocimiento. Por último, y en relación con los esquemas vistos hasta ahora, la ontología proporciona un vocabulario común para el intercambio de información dentro de un dominio.

Son varias las definiciones del término *ontología* propuestas en la literatura científica. Gruber (Gruber, T., 1993) definió ontología como:

“[...] especificación explícita de una conceptualización.”

La cual, se convirtió en la más aceptada por la comunidad científica hasta que fue completada por Studer (Sure, Y.; Studer, R., 2005) como:

“[...] especificación explícita y formal de una conceptualización compartida.”

Donde:

- **Conceptualización** se refiere a una representación abstracta o modelo, de algún fenómeno en el mundo, perteneciente al Universo del Discurso. En dicho modelo estarán representados los conceptos y relaciones relevantes de dicho

fenómeno.

- **Explícita** se refiere a definición explícita que, para su uso, es necesario hacer de los conceptos, relaciones y restricciones.
- **Formal** se refiere al hecho de emplear un formalismo de representación, que permita a la ontología ser legible o interpretable por una computadora.
- **Compartida** expresa la noción de conocimiento consensuado, es decir, el conocimiento compartido no es privado de un individuo, sino que ha sido consensuado por un grupo o comunidad.

Nicola Guarino (Guarino, N., 1998) complementó la propuesta de Studer, definiendo ontología como:

“[...] teoría lógica que da cuenta del significado intencional de un vocabulario formal, es decir, de su compromiso ontológico hacia una conceptualización particular del mundo”.

Guarino define, por tanto, ontología como teoría lógica que aporta la semántica a un conjunto de términos del Universo del Discurso, según la interpretación que su creador haga de la realidad. De este modo, es posible tener distintas representaciones de una misma realidad, dado que éstas dependen de la visión que el desarrollador tenga de esa realidad.

Aun existiendo consenso sobre esta última definición del término, (Corcho, O.; Fernández-López, M.; Gómez-Pérez, A., 2003) pone de manifiesto cómo, en la práctica, existen distintos significados atribuidos al término ontología. Según sus autores, distintas comunidades científicas investigan en áreas tan diferentes como la gestión del conocimiento, procesamiento del lenguaje natural, comercio electrónico, Web, etc., la heterogeneidad de objetivos y uso en estas áreas, ha propiciado las distintas acepciones aplicadas al término ontología.

La propuesta de Jasper y Uschold (Jasper, R.; Uschold, M., 1999) intentan paliar este problema, mediante la definición de los requisitos mínimos que un modelo debe cumplir para ser considerado una ontología. Según estos autores:

“Una ontología puede tomar diversas formas, pero necesariamente incluirá un vocabulario de términos y alguna especificación de sus significados. Esto incluye definiciones y una indicación de cómo los conceptos están interrelacionados, que colectivamente imponen una estructura al dominio y restringen las posibles interpretaciones de los términos”.

Daconta et al. (Daconta, M.C.; Obrst, L.J.; Smith, K.T., 2003) proponen una clasificación de las distintas representaciones, ordenándolas en base a la capacidad de representación semántica. Así, el orden propuesto sería: taxonomía, tesoro, modelo conceptual y teoría lógica. Actualmente, se aplica el concepto *ontología ligera* (*lightweight ontology*) a las representaciones anteriores con menor contenido semántico:

taxonomías y tesauros. Los modelos conceptuales y las teorías lógicas son considerados *ontologías pesadas* (*heavyweight ontologies*) en base a su capacidad de representación.

Para establecer los tipos de ontologías, existen múltiples propuestas que las clasifican en base a distintos criterios: según el contenido (Mizoguchi, R.; Vanwelkenhuysen, J.; Ikeda, M., 1995); según el volumen y el tipo de estructura, pero también por la conceptualización específica del conocimiento (Heijst, G.v.; Schreiber, A.T.; Wielinga, B.J., 1997); según su grado de dependencia: ontologías de alto nivel, de dominios genéricas, de dominio, de aplicación, y de tareas (Guarino, N., 1998); o por la riqueza semántica de su estructura interna (Lassila, O.; McGuinness, D., 2001).

Atendiendo al ámbito de la conceptualización específica del conocimiento (Heijst, G.v.; Schreiber, A.T.; Wielinga, B.J., 1997), las ontologías pueden clasificarse en:

- **Ontologías de representación:** proporcionan conceptos subyacentes a los paradigmas o formalismos de representación del conocimiento, es decir, proporcionan el vocabulario necesario para modelar otras ontologías, utilizando determinado paradigma de representación del conocimiento. Es el caso de *Frame Ontology* disponible en el servidor de Ontolingua.
- **Ontologías genéricas o meta-ontologías:** proporcionan términos genéricos reutilizables en diferentes dominios, como por ejemplo, los términos: estado, evento, acción, componente, etc.
- **Ontologías de dominio:** que expresan conceptos que son específicos de un dominio determinado. Los conceptos en este tipo de ontologías son definidos usualmente como especializaciones de conceptos existentes en ontologías genéricas.
- **Ontologías de aplicación:** contienen todas las definiciones que son necesarias para modelar los conocimientos requeridos por una aplicación particular. Incluyen conceptos tomados de ontologías de dominio y genéricas, a menudo definidas utilizando el vocabulario indicado en ontologías de representación. Pueden contener extensiones de métodos y tareas específicas. Las ontologías de aplicación tienden a ser menos que las anteriores, pues son especificaciones concretas del dominio que se necesita para realizar una tarea particular en ese dominio determinado.

2.3.2 Componentes de una ontología

Una ontología está formada por un conjunto de conceptos, relaciones y restricciones, expresados mediante definiciones aceptadas por los miembros de una comunidad para expresar su área de conocimiento y en un formato procesable por la máquina. Concretamente, los elementos que forman una ontología se resumen en:

- **Conceptos:** Un concepto puede ser cualquier cosa acerca de la cual se pueda aseverar dentro del Universo del Discurso. De este modo, un concepto puede ser un objeto tangible (físico) o un objeto intangible, por ejemplo la descripción de una tarea, función, acción, estrategia, etc. Cada concepto tiene un nombre que lo

identifica y puede poseer un conjunto de atributos.

- **Relaciones:** Las relaciones representan el tipo de interacción entre los conceptos de un dominio, son formalmente definidas como subconjuntos del producto cartesiano de n conjuntos, esto es $R: C1 \times C2 \times \dots \times Cn$, donde cada conjunto está formado por el agregado de instancias de las clases relacionadas. Existen, además, relaciones con un significado especial, concretamente: la relación de especialización expresa una asociación *es_un* entre dos conceptos y la relación de composición expresa una asociación *es_parte_de* entre dos conceptos.
- **Funciones:** Son un caso especial de relaciones donde el enésimo elemento de la relación es único para los $n-1$ anteriores. Formalmente las funciones se definen como $F: C1 \times C2 \times \dots \times Cn-1 \times Cn$, donde la relación entre los valores de los $n-1$ conceptos, determina el valor del concepto *enésimo*.
- **Axiomas:** Los axiomas se usan para definir afirmaciones siempre ciertas en la realidad modelada. Los axiomas definidos en una ontología pueden ser estructurales o no estructurales. Un axioma estructural establece condiciones relacionadas a las jerarquías de la ontología, conceptos y atributos definidos. Un axioma no estructural se establece entre atributos de un concepto.
- **Instancias:** Representan elementos específicos del dominio de la ontología y son instancias concretas de los conceptos modelados.

Las ontologías difieren según su grado de aplicación como sugiere Steve (Steve, G.; Gangemi, A.; Pisanelli, D.M., 1997) y Guarino (Guarino, N., 1998). Sobre las metodologías para su creación existe una gran variedad de propuestas, todas ellas muy relacionadas con las clásicas para análisis de dominios, creación de tesauros y desarrollo de bases de conocimiento, entre las mejor formalizadas se encuentran *Methontology* (Gómez-Pérez, A.; Fernández-López, M.; Corcho, O., 2004) y *Ontology Development 101* (Noy, N.; McGuinness, D.L., 2001).

2.3.3 Áreas de aplicación

Dentro de las áreas de aplicación de las ontologías, destacan:

- **Ingeniería del Conocimiento.** Aunque íntimamente relacionada con la Ingeniería del Software, la ingeniería del conocimiento ha surgido como una disciplina diferente. Entre sus principales características se encuentran técnicas de *elicitación* o extracción de conceptos y relaciones de un dominio, modelado del conocimiento, formalismos para la representación del conocimiento y técnicas para la automatización del razonamiento.
- **Procesamiento de lenguaje natural.** El uso de ontologías en esta área permite la definición de elementos gramaticales del lenguaje y las relaciones entre ellos. De este modo es posible, por ejemplo, la realización del análisis sintáctico de un texto.

- **Interoperatividad entre sistemas de información heterogéneos:** Las ontologías proporcionan una importante solución en la integración de información heterogénea proveniente de múltiples fuentes. En este caso, las ontologías realizarían la labor de mediador, actuando como puente al establecer relaciones entre términos de distintas fuentes.
- **Búsqueda semántica en sitios Web:** El uso de ontologías asociadas a motores de búsqueda permite la realización de búsquedas semánticas, a diferencia de las actuales, meramente sintácticas. De este modo, y mediante el procesamiento conceptual de la consulta, es posible obtener un conjunto de resultados cuyo contenido es semánticamente similar, aunque difiera sintácticamente.
- **Modelado de empresas:** La aplicación de ontologías en esta área permite, principalmente, gestionar el conocimiento organizacional, facilitando el uso de un lenguaje y reglas de negocio, únicos para los distintos niveles de la empresa.
- **Aplicaciones en la Web Semántica:** Las ontologías se muestran ya como un elemento fundamental para especificar la semántica de los contenidos en la Web Semántica. A través de un lenguaje de marcado sintáctico y semánticamente consensuado, posibilitan el procesamiento y la inferencia automática de conocimiento.

Para más información acerca del uso actual de las ontologías: en (Palacios, V. et al., 2006d) se presenta un estudio bibliométrico sobre el uso de las ontologías, incluyendo aspectos como el ámbito de aplicación, ubicación geográfica o aspectos socio-económicos; en (Fuentes-Lorenzo, D.; Morato, J.; Gómez, J.M., 2009) se muestra un estudio comparativo del uso de ontologías de alto nivel (*top ontologies*) en organizaciones, aplicaciones, industria e investigación.

2.3.4 Ontologías de acceso libre

A continuación se presentan, de forma resumida, algunas de las ontologías más representativas publicadas en Web.

- **Cyc (Cycorp, 2010):** Es un proyecto de Inteligencia Artificial que intenta aunar una representación ontológica formal con una gran base de hechos sobre conocimiento común. El objetivo es servir como soporte a aplicaciones que realicen razonamiento humano simulado. El proyecto se inició en 1984 y, aunque la base de conocimiento es propietaria, existe una versión libre llamada *OpenCyc* (Cycorp, 2009a). Actualmente Cyc se ha puesto a disposición de investigadores en Inteligencia Artificial mediante una licencia de investigación, la versión se ha denominado *ResearchCyc* (Cycorp, 2009b). Tanto *OpenCyc* como *ResearchCyc* pueden ser descargadas para su uso.
- **Web Knowledge Base (WebKB) (WebKB, 2003a).** Es un Sistema de Gestión de Conocimiento accesible vía Web. WebKB permite a los usuarios almacenar, organizar y recuperar conocimiento en una gran base de conocimiento. Los usuarios pueden hacer uso del sistema de forma privada, su conocimiento no es compartido por el resto de usuarios, o de forma compartida. WebKB, en sus dos

versiones WebKB-1 y WebKB-2, no está disponible para su descarga pero es accesible vía Web.

- *Suggested Upper Merged Ontology* (SUMO) (Suggested Upper Merged Ontology, 2010). Es una ontología que incluye conceptos generales con el objetivo de servir de base para el desarrollo de ontologías más específicas. Este tipo de ontologías se conocen como: de alto nivel o fundacionales. SUMO proporciona también un conjunto de ontologías de dominio y una ontología de nivel intermedio (*Mid-Level Ontology: MILO*) que permite enlazarlas con la ontología general. Las ontologías proporcionadas por SUMO están desarrolladas bajo licencia y pueden ser descargadas para su uso.
- *Generalized Upper Model* (GUM) (Bremen University, 2010). Es una ontología general e independiente del dominio, diseñada para aplicaciones lingüísticas, que representa el conocimiento de modo compatible con técnicas de procesamiento del lenguaje natural. Actualmente posee una versión desarrollada sobre el lenguaje OWL-DL. GUM versión 3.0 está disponible para su descarga y uso.
- *WordNet* (WN) (Princeton University, 2010). Es un sistema léxico de referencia que contiene una red léxica del idioma inglés y que integra, entre otras relaciones, las de meronimia, generalización, sinonimia entre los distintos sentidos (*synsets*) de cada conjunto de sinónimos. En su versión 3.0 WN tiene 155.287 términos, 117.659 sentidos y 206.941 pares palabra-sentido. WordNet es libre y puede ser descargado para uso.
- *Descriptive Ontology for Linguistic and Cognitive Engineering* (DOLCE) (Laboratory for Applied Ontology, 2006). Es una ontología cuyo objetivo es capturar, mediante un conjunto de categorías y primitivas, las categorías ontológicas subyacentes en el lenguaje natural y en el sentido común. De este modo, DOLCE no tiene como objetivo ser una ontología universal sino una ontología fundacional, esto es, un punto de partida o base para la construcción de ontologías mediante el establecimiento de relaciones e integración de ontologías existentes. La importancia de DOLCE reside en su planteamiento, de hecho, DOLCE ha sido reutilizada en varias ontologías predecesoras como PROTON, SUMO o GUM.
- *Semantically-Interlinked Online Communities* (SIOC) (Digital Digital Enterprise Research Institute, 2010). Es una ontología RDF diseñada para el entorno Web, cuyo objetivo es la integración de información proveniente de comunidades online, esto es, información de la Web social. Recientemente ha tenido una gran popularidad a través de su uso en varias aplicaciones comerciales y de código abierto. Es utilizada habitualmente en conjunción con el vocabulario FOAF para expresar información personal complementaria a la información social.
- *Upper-level Mapping and Binding Exchange Layer* (UMBEL) (UMBEL.org, 2009). Es una ontología ligera cuyo objetivo es relacionar ontologías externas, mediante el alineamiento de sus clases con conceptos de UMBEL. Los conceptos de UMBEL conforman, de este modo, la espina dorsal en la

interrelación de ontologías en el entorno Web. Los conceptos de UMBEL provienen de un proceso de destilado a partir de OpenCyc y están interrelacionados haciendo uso de los lenguajes SKOS y OWL-Full. En la propuesta ha sido de especial interés en cuanto a su estructura y al tratamiento de aspectos como la sinonimia o el plurilingüismo.

- *PROTo ONtology* (PROTON) (Terziev, I.; Kiryakov, A.; Manov, D., 2005). Es una ontología genérica que proporciona cobertura a conceptos generales en ámbitos como la anotación semántica, indización o recuperación de documentos. Sus principios de diseño son: independencia de dominio, definiciones lógicas ligeras, alineamiento con los estándares más populares y cobertura para dominios concretos, permitiendo su modelado a través de la especialización de los conceptos incluidos, tarea para la cual ha sido específicamente diseñada. Especificada mediante el lenguaje OWL-Lite, posee 300 clases y 100 propiedades desarrolladas a partir de OpenCyc, WordNet 2.0, DOLCE y EuroWordNet, entre otras.
- Otros ejemplos de ontologías son, en el dominio biomédico: *Gene Ontology* (The Gene Ontology, 1999) proporciona un vocabulario controlado para la descripción de los genes de cualquier organismo; *Protein Ontology* (Protein Information Resource, 2009) para la descripción y razonamiento del conocimiento sobre proteínas; *Foundational Model of Anatomy* (Washington University, 2010) para el modelado de la estructura del cuerpo humano. En el dominio de las Ciencias Naturales: *Systems Biology Ontology* (European Bioinformatics Institute, 2010) para el modelado de sistemas biológicos; *Plant Ontology* (Plant Ontology Consortium, 2010) para el modelado de estructuras, crecimiento y estados de desarrollo de plantas. En el ámbito de la Lingüística: *General Ontology for Linguistic Description (GOLD)* (GOLD Community, 2009) para la descripción del lenguaje humano; *ThoughtTreasure* (Mueller, E.T., 2003) base de conocimientos y arquitectura para el procesamiento del lenguaje natural. En relación con el Patrimonio Cultural: *CIDOC CRM* (International Council of Museums, 2006) es un Modelo de Referencia Conceptual desarrollado por el *International Committee on Documentation of the International Council of Museums (ICOM-CIDOC)* para la representación de información de herencia cultural.

2.3.5 Repositorios semánticos

Dentro de este apartado dedicado a las ontologías, es necesario mencionar el software de soporte o *middleware* que permite el almacenamiento y gestión de ontologías. La posibilidad de gestión de la persistencia de ontologías, esto es, almacenamiento y gestión en Bases de Datos, es lo que ha llevado a denominar a estos sistemas como *Repositorios Semánticos*.

Obtenida del estudio realizado en (Corbera, S., 2007), a continuación se enumera una selección de los repositorios semánticos más utilizados actualmente:

2.3.5.1 OWLIM

OWLIM (Ontotext, 2010a) es un repositorio semántico de alto rendimiento desarrollado en Java. Está disponible en dos versiones:

- *SwiftOWLIM* ejecuta razonamiento y evaluación de consultas en memoria, mientras que una estrategia de persistencia asegura la preservación, consistencia e integridad de los datos. No incluye optimización de consultas, lo que genera un rendimiento menor que *BigOWLIM*, pero está sujeto a una licencia libre.
- *BigOWLIM* opera directamente con ficheros binarios persistentes, los cuales permiten alcanzar billones de sentencias. *BigOWLIM* es el único motor que ofrece inferencia OWL a más de 1 billón de triplas. Incluye optimización en las consultas pero su licencia es comercial.

El desarrollo de OWLIM está parcialmente soportado por SEKT (SEKT Project, 2006) y otros proyectos FP6 (Sexto Programa Marco).

2.3.5.2 Plataforma KIM

La plataforma KIM (*Knowledge and Information Management*) (Ontotext, 2010b) proporciona la infraestructura y los servicios para anotación semántica automática, indexación y recuperación de contenido no estructurado y semi-estructurado.

Las aplicaciones directas de KIM son:

- Generación de metadatos para la Web Semántica, que permite la conexión mediante hiperenlaces y visualización y navegación avanzadas.
- Gestión del conocimiento, incremento de la eficiencia de los indexadores existentes, recuperación, clasificación y aplicaciones de filtrado.

KIM incluye:

- PROTON, KIMSO, KIMLO y KIM World KB.
- KIM Server – con API para acceso remoto e integración.
- *Front-ends* (componentes o interfaces de usuario): KIM Web UI y plugin para Internet Explorer.

Como funcionalidad base, KIM analiza textos y reconoce referencias a entidades (como personas, organizaciones, localizaciones, fechas). Una vez identificadas éstas, intenta hacer corresponder la referencia con una entidad conocida, haciendo uso de la URI y descripción de la entidad conocida. Alternativamente, se genera una nueva entidad, con una nueva URI y descripción. Finalmente, la referencia en el documento es anotada con la URI de la entidad. A este proceso se le llama, también al resultado, anotación semántica. Este tipo de metadatos puede ser usado para indexación, recuperación, visualización y conexión automática mediante hiperenlaces de documentos.

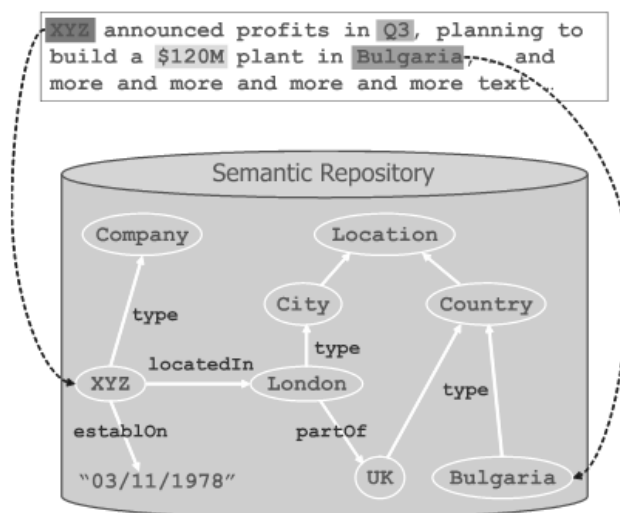


Figura II-4: Esquema de anotación semántica en KIM extraído de (Ontotext, 2010b).

Para permitir un proceso de arranque de aplicaciones sencillo, KIM está equipado con la ontología genérica PROTON, con unas 300 clases y 100 propiedades. Además, KIM viene con una base de conocimiento (KIM KB), precargada con unas 200.000 descripciones de entidades. Su rol es el de proporcionar, como conocimiento de fondo, una cobertura lo más exhaustiva posible de las entidades generales. Estas entidades son aquellas que son consideradas “well-known”, motivo por el cual es complicado extraer de manera automática sus descripciones.

Desde un punto de vista técnico, la arquitectura de la plataforma permite a las aplicaciones basadas en KIM, la gestión completa de la anotación semántica, esto es, anotación, recuperación de contenido, consulta y modificación, todo ello basado en restricciones semánticas.

2.3.5.3 Sesame

Sesame (OpenRDF.org, 2010) proporciona un marco de desarrollo para el almacenamiento, consulta y razonamiento con RDF y RDF Schema. Dentro de este marco, incorpora un conjunto de librerías para desarrollar aplicaciones semánticas, compatibles con el lenguaje Java (Sun Microsystems, 2010), que permiten analizar gramaticalmente, interpretar, consultar y almacenar información semántica. Además del conjunto de librerías, una de las principales aportaciones de Sesame es el repositorio que incorpora. El repositorio puede trabajar de distintos modos: en memoria principal, haciendo uso de ficheros, o utilizando una base de datos relacional. El desarrollador tiene la posibilidad de instalar un servidor de Sesame, que hace transparente el acceso y actualización de la información, al tiempo que facilita el acceso al repositorio, por ejemplo, vía Web. Respecto a los lenguajes de consulta, Sesame soporta SeRQL (leído “circle”), RDQL y RQL.

2.3.5.4 Jena

Jena (Hewlett-Packard Development Company, 2009a) es, junto con Sesame, el marco de desarrollo más utilizado en aplicaciones semánticas. De modo similar a Sesame,

proporciona un conjunto de librerías de código abierto para el desarrollo de aplicaciones semánticas en Java. Desarrollado y mantenido por los laboratorios HP (Hewlett-Packard Labs, 2009), para la investigación y desarrollo de la Web Semántica, Jena ofrece un entorno programático para RDF, RDFS y OWL. También incluye la posibilidad de gestionar la persistencia de ontologías mediante sistemas gestores de bases de datos relacionales. Los lenguajes de consulta soportados son RDQL y SPARQL. A diferencia de Sesame, su arquitectura es más simple, de modo que no proporciona un servidor que gestione el acceso y la persistencia de las ontologías, quedando estas tareas a cargo del desarrollador.

La versión actual de Jena es la 2.6.2 y el marco de trabajo incluye:

- Un analizador sintáctico de RDF.
- Un API de RDF que permite crear y manipular modelos RDF desde una aplicación Java. Además, proporciona clases Java para representar modelos, recursos, propiedades, literales, sentencias, etc.
- Lectura y escritura RDF en RDF/XML, N3 y N-Triples.
- Un API para la gestión de ontologías con soporte para OWL, DAML y RDFS. La gestión es independiente del lenguaje y los recursos no están ligados estáticamente a una clase Java particular. También permite la validación básica de OWL, ofrece soporte a inferencia y detecta la violación de restricciones e inconsistencias en la ontología.
- Motor de consultas SPARQL.
- Almacenamiento en memoria y persistente. Jena permite crear modelos persistentes de forma transparente en una base de datos relacional, concretamente, soporta MySQL, Oracle y PostgreSQL.
- Soporte a inferencia. Jena incluye el siguiente conjunto de razonadores básicos y ofrece un API para incluir nuevos:
 - Razonador OWL.
 - Razonador DAML.
 - Razonador de Reglas RDF.
 - Razonador de Reglas Genéricas.

2.3.5.5 Joseki

Joseki (Hewlett-Packard Development Company, 2009b) es un API Web para consultas y actualizaciones remotas de modelos RDF. El sistema se compone de un API cliente que ofrece el acceso necesario para realizar operaciones de consulta sobre el protocolo RDF WebAPI. Joseki no es una base de datos RDF propiamente dicha, pero sí es un

servidor que implementa una interfaz de consulta SPARQL para Jena.

El servidor RDF puede correr embebido en una aplicación como un programa independiente, o como una aplicación Web dentro del correspondiente servidor de aplicaciones. Es el encargado de realizar operaciones de consulta y actualización sobre los modelos que alberga.

Joseki se puede obtener de la página de descarga del sitio [20] en varios paquetes. Es un sistema completo, que incluye el servidor Web Jetty, usado para ejecuciones embebidas o independientes, y el fichero WAR como aplicación Web. También están incluidos el código fuente, documentación y Javadoc.

2.3.5.6 Karlsruhe Ontology Management Infrastructure (KAON)

KAON (Forschungszentrum Informatik - Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB), 2001) es también un marco de desarrollo para aplicaciones basadas en ontologías. También proporciona librerías para el desarrollo de aplicaciones semánticas en Java. De forma similar a Sesame, proporciona un servidor que permite el acceso vía Web a las ontologías y, a diferencia de los anteriores, incluye una herramienta gráfica para visualizar y modelar ontologías. Respecto a los lenguajes de consulta y de representación, posee el inconveniente de que ambos son propietarios. Actualmente se encuentra ha desarrollado una nueva versión, KAON 2, basada en OWL-DL y F-Logic (en lugar de RDFS, como su antecesor) y que incluye soporte a SPARQL como lenguaje de consulta. Desgraciadamente, la nueva versión ha dejado de ser software libre, lo que impide la adaptación y modificación de su funcionalidad.

2.3.5.7 Integrated Ontology Development Toolkit (IODT)

IODT (IBM, 2004) es un conjunto de herramientas, propiedad de IBM, para almacenamiento, manipulación, consulta, e inferencia de ontologías y sus instancias correspondientes. También se puede definir como un conjunto de herramientas que dirige el desarrollo de una ontología. Este conjunto de herramientas incluye EODM (*EMF Ontology Definition Metamodel*), el marco de trabajo de EODM, y un repositorio de ontologías OWL (llamado Minerva).

EODM procede del *Ontology Definition Metamodel (ODM)* (Object Management Group, 2009) del *Object Management Group* (Object Management Group, 2010) y es implementado en el *Eclipse Modeling Framework (EMF)* (The Eclipse Foundation, 2010). Para facilitar el desarrollo de aplicaciones semánticas, EODM incluye análisis sintáctico y serialización RDFS/OWL, transformación entre RDFS/OWL y otros lenguajes de modelado de datos, así como inferencia. Estas funciones se pueden invocar desde el Workbench EODM o Minerva y son llamadas por las aplicaciones que usan ontologías. EODM también es un proyecto de código abierto de Eclipse.

IODT incluye los siguientes componentes, que pueden descargarse y usarse de manera independiente y cooperativa:

- EODM es la librería de tiempo de ejecución que permite a la aplicación: leer y serializar una ontología RDFS/OWL en formato RDF/XML, manipular una

ontología usando objetos Java, llamar a un motor de inferencia y acceder a los resultados de la inferencia, así como transformar ontologías a otros modelos.

- El marco de trabajo de EODM es un entorno para el desarrollo de ontologías integrado en Eclipse, que soporta la creación, gestión y visualización de ontologías.
- El Repositorio de Ontologías OWL (llamado Minerva) es un sistema basado en RDBMS (*Relational Database Management Systems*) de alto rendimiento para el almacenamiento, inferencia y consulta de ontologías OWL. Soporta DLP (*Description Logic Program*), un subconjunto de OWL-DL, y consultas conjuntivas, un subconjunto del lenguaje SPARQL.

2.3.5.8 AllegroGraph

AllegroGraph (Franz Inc., 2010), de Franz Inc., es un sistema para cargar, almacenar en memoria y realizar consultas contra datos RDF. Incluye un interfaz SPARQL y razonamiento RDFS. Tiene un interfaz Java y otra en Prolog. Proporciona un sistema de alto rendimiento, persistente, con una base de datos gráfica basada en RDF (con soporte para SPARQL y RDFS), y que permite razonamiento Prolog desde aplicaciones Java.

Allegrograph está diseñado para conseguir altas velocidades en las operaciones de carga y consulta. La carga de tripletas, a través de sus analizadores sintácticos de RDF/XML y N-Triples, facilita el trabajo, principalmente, con grandes ficheros.

2.3.5.9 Ontolingua Server

Ontolingua Server (Knowledge System Laboratory - Stanford University, 2001) es un conjunto de herramientas y servicios que soporta la construcción de ontologías compartidas entre grupos distribuidos, desarrollado por el *Knowledge Systems Laboratory* (KSL) en la Universidad de Stanford (Stanford University, 2010a). La arquitectura del servidor de ontologías proporciona acceso a una librería de ontologías, traductores a lenguajes (Prolog, CORBA IDL, CLIPS, Loom, etc.) y un editor para crear y navegar entre las ontologías. Las aplicaciones remotas o locales pueden acceder a cualquiera de las ontologías de la librería de ontologías usando el protocolo OKBC (*Open Knowledge Based Connectivity*).

Ontolingua Server soporta la creación, evaluación, acceso, uso y mantenimiento de ontologías. Además, no sólo permite el desarrollo de ontologías con individuos, sino que también gestiona el proceso de lograr consenso sobre ontologías comunes entre grupos distribuidos. Hace uso de la World Wide Web para permitir un amplio acceso y provee a los usuarios con la habilidad de publicar, navegar, crear y editar ontologías almacenadas en el servidor. El servidor ofrece tres modos de interacción.

- Primero, grupos remotos distribuidos pueden usar sus navegadores Web para acceder, construir, y mantener ontologías almacenadas en el servidor. El servidor interactúa con ellos a través del protocolo HTTP y el lenguaje de hipertexto (HTML) usado en la World Wide Web. Esto hace al servidor accesible para un público muy amplio. Cualquier usuario con herramientas de

navegación Web habituales, puede navegar, construir y mantener ontologías almacenadas en el servidor. El servidor permite a varios usuarios trabajar simultáneamente sobre una misma ontología en una sesión compartida. El editor ofrece funcionalidades para soportar el trabajo colaborativo (por ejemplo, notificaciones, comparación o registros de cambio compartidos).

- Segundo, las aplicaciones remotas pueden consultar y modificar ontologías almacenadas en el servidor a través de Internet. Estas interacciones programáticas usan una interfaz de red que proporciona el protocolo *Generis Frame Protocol*. Esta interfaz de red soporta consultas y actualizaciones.
- Tercero, un usuario puede convertir una ontología a un formato usado por una aplicación específica. Se puede usar la conversión a diferentes lenguajes, por ejemplo, una conversión al lenguaje CLIPS produce un conjunto de definiciones de clase y reglas de inferencia que pueden ejecutarse directamente en una aplicación basada en CLIPS, mientras que una conversión a un Lenguaje de Definición de Interfaces (IDL) produce un fichero cabecera IDL que un programa CORBA (*Common Object Request Broker Architecture*) puede usar para interactuar con un ORB (*Object Request Broker*).

2.3.5.10 OntoSaurus

OntoSaurus (University of Southern California, 2007) ha sido desarrollado por el Instituto de Ciencias de la Información (ISI) en la Universidad de California del Sur. Está compuesto por dos módulos: un servidor de ontologías, que usa Loom como sistema de representación del conocimiento, y un navegador de ontologías del servidor, que empaqueta dinámicamente páginas HTML. Las páginas HTML incluyen la documentación textual e imágenes, y muestran las jerarquías de ontologías.

Está implementado usando las extensiones de Netscape2 para HTML2.0, concretamente utilizando marcos y JavaScript. Usa un servidor HTTP basado en Lisp (CL-HTTP) que fue desarrollado en el *MIT AI Laboratory*.

En OntoSaurus se pueden editar las ontologías con formas HTML y existen traductores de Loom a Ontolingua, KIF, KRSS y C++.

En el estudio realizado en (Corbera, S., 2007) se pueden consultar con mayor profundidad estos y otros repositorios semánticos, incluyendo también una evaluación de editores de ontologías. El estudio también aporta una comparativa entre los sistemas evaluados.

2.3.6 Métodos de correspondencia entre ontologías

Para finalizar de este recorrido por las ontologías, es preciso abordar su interrelación, como aspecto fundamental dentro de los objetivos de esta propuesta.

Como se señala en (Noy, N., 2004), existen multitud de ontologías utilizadas actualmente que se solapan en contenido. A esto hay que añadir que la mayoría de los dominios están representados en más de una ontología, motivado por la necesidad de

distintas vistas o representaciones de un mismo dominio. Incluso, intentando evitar la heterogeneidad a través del uso de ontologías genéricas, es necesario realizar la correspondencia entre los conceptos fundacionales y los componentes específicos que los extienden.

Para lograr que las aplicaciones semánticas puedan interactuar y hacer uso de distintas fuentes, es necesario integrar el conocimiento representado en distintos modelos de forma heterogénea, resolviendo problemas (Smart, P.R.; Engelbrecht, P.C., 2008) y (Marrero, M. et al., 2006), derivados de:

- Terminología: El hecho de que las ontologías a combinar estén definidas en diferentes lenguajes de representación, suele implicar tanto problemas de sintaxis como de semántica. Problemas sintácticos, en cuanto a que diferentes lenguajes ofrecen diversas sintaxis para representar los mismos conceptos. Problemas semánticos, puesto que la semántica puede variar de un lenguaje a otro, bien porque no esté claramente especificada o bien porque simplemente se trate de semánticas diferentes. Estos problemas son similares a los usuales de homonimia y sinonimia pero a nivel de concepto. Estos motivos explican que diversos métodos de correspondencia de ontologías contemplen una etapa previa de normalización, que consiste en convertir ambas ontologías a un mismo lenguaje de representación (Kalfoglou, Y.; Schorlemmer, M., 2003).
- Modelado: Un mismo conocimiento puede ser modelado de diversos modos, de forma que, aún utilizando el mismo lenguaje de representación y los mismos constructos, las ontologías resultantes difieran. Como en el caso de los problemas terminológicos, podemos clasificar los problemas del mismo modo, aunque en este caso referidos al nivel conceptual o de ontología, esto es: sintácticos y semánticos. Los problemas sintácticos incluyen los relacionados con: el paradigma o el modo en el que se van a representar los conceptos y que, dado que depende de nuestro punto de vista, puede variar de unas ontologías a otras; la cobertura, que determina la extensión del dominio representada; y la granularidad o nivel de detalle de las representaciones. Los problemas semánticos, relativos a los términos del lenguaje natural con los que anotamos un concepto, y que por tanto, son susceptibles de incluir ambigüedades tales como sinonimia u homonimia. Los problemas de modelado son problemas conceptuales, y por tanto, más difíciles de resolver que los problemas relacionados con el lenguaje de representación.

El modo de integrar los modelos implica encontrar correspondencias de calidad entre los conceptos que representan, esto es, resolviendo los problemas derivados del lenguaje y el modelado. Este proceso es lo que se denomina *correspondencia entre ontologías* (*ontology matching*¹⁰). La correspondencia entre ontologías es una de las áreas de investigación más activas de la Web Semántica, en especial, el establecimiento automático de correspondencias ajustadas a la semántica de ambas fuentes.

¹⁰ En la literatura se utiliza como sinónimo el término *ontology mapping* y su castellanización *mapeo de ontologías*. Se ha preferido el uso del término *correspondencia entre ontologías* al considerarse una mejor traducción del término según el diccionario de la RAE.

El proceso de correspondencia entre ontologías tiene como resultado un *alineamiento*, definido como un conjunto de relaciones entre conceptos de dos ontologías. Dichas relaciones pueden ser de equivalencia, generalización, reglas de transformación, etc. Más concretamente (Euzenat, J.; Shvaiko, P., 2007), dadas dos ontologías o y o' , su alineamiento comprende un conjunto de correspondencias (*correspondences*) o relaciones bidireccionales¹¹ entre entidades, simples o compuestas, que pertenecen a o y o' , respectivamente. Una correspondencia se describe como una cuádrupla $\langle e, e', r, n \rangle$ donde:

- e y e' son las entidades entre las que se define la relación.
- r es la relación que se declara ente las entidades, a través de la correspondencia.
- n es el grado de confianza en la correspondencia. De este modo se establece una forma de expresar en qué medida es correcta la correspondencia. Este valor opcional es establecido, típicamente, mediante la información de retroalimentación que proporcionan los usuarios o el análisis de los registros de sucesos.

Tal y como se plantean, los alineamientos son independientes de un lenguaje particular y requieren, por tanto, un lenguaje de definición que soporte la expresión de las relaciones que establecen.

Los estudios sobre los ámbitos de aplicación de la correspondencia entre ontologías (Euzenat, J.; Shvaiko, P., 2007) y (Shvaiko, P.; Euzenat, J., 2005), establecen los siguientes requisitos relativos al proceso:

- El tipo de entrada que procesa el sistema de correspondencia, que puede ser información del esquema o de las instancias. En ocasiones las instancias no están disponibles por seguridad o porque se obtienen como resultado de una consulta en tiempo de ejecución. Este tipo de sistemas procesan los esquemas para obtener los alineamientos.
- El comportamiento específico del proceso, relativo a:
 - Automático o manual: si el proceso se realiza de forma automática o de forma manual. En este caso, por automático debe entenderse sin intervención del usuario, incluyendo la información de retroalimentación. Un ejemplo de este ámbito son las comunicaciones entre agentes inteligentes.
 - Correcto o incorrecto: si devuelve o no correspondencias incorrectas. En algunos ámbitos se prima la recuperación de contenido frente a su precisión. De este modo se asume que parte de los resultados no sean correctos. Suele darse en ámbitos donde el usuario final puede discernir u ordenar el conjunto de resultados.

¹¹ Cuando la relación es unidireccional recibe el nombre de *mapping*. No recomendamos el uso de la castellanización *mapeo* puesto que el término más próximo existente es *mapear* cuya definición, según la RAE es “*Localizar y representar gráficamente la distribución relativa de las partes de un todo*”, cuya semántica tampoco se ajusta al término anglosajón. En su lugar proponemos el uso de *asociación unidireccional*, tomado de Ingeniería del Software, o *arista*, tomado de la Teoría de Grafos.

- Completo o incompleto: si devuelve o no todas las correspondencias. En algunos ámbitos se prima el rendimiento, en términos como tiempo de respuesta, frente a la completitud de los resultados. Un escenario típico incluiría consultas on-line sobre una base de conocimientos, con un tiempo máximo de respuesta establecido.
- Si se realiza en tiempo de ejecución o en tiempo de diseño. En este caso la diferencia estriba en el momento en el cual son necesarios los alineamientos: en diseño, por ejemplo, en el ámbito de la integración de esquemas, donde se puede generar el alineamiento y que éste quede almacenado para su uso posterior; o en tiempo de ejecución, donde el alineamiento se genera, por ejemplo, como resultado de una consulta o de una comunicación entre agentes.

Respecto al proceso de correspondencia, (Euzenat, J.; Shvaiko, P., 2007) define otros parámetros que pueden intervenir en él, concretamente:

- El uso de un alineamiento previo como entrada (A), que será completado como resultado del proceso.
- Parámetros de la correspondencia (p) como, por ejemplo, pesos, umbrales, etc.
- Recursos externos utilizados en el proceso de correspondencia (r), como conocimiento común, tesauros específicos del dominio, etc.

De este modo, el proceso de correspondencia puede ser visto como una función f , que devuelve el alineamiento A' entre dos ontologías o y o' , haciendo uso de un conocimiento común r , según los parámetros establecidos en p .

$$A' = f(o, o', A, p, r)$$

Figura II-5: Función de alineamiento, tomada de (Euzenat, J.; Shvaiko, P., 2007)

Existen multitud de algoritmos de correspondencia que articulan la función de alineamiento (Rahm, E.; Bernstein, P., 2001) y (Kalfoglou, Y.; Schorlemmer, M., 2003), en base a la combinación de una o varias de las siguientes técnicas (Van Hage, W.R.; Katrenko, S.; Schreiber, G., 2005), (Hu, W.; Qu, Y., 2008), (Tang, J. et al., 2006), (Euzenat, J.; Valtchev, P., 2004) y (Melnik, S.; Rahm, E.; Bernstein, P.A., 2003):

- Técnicas terminológicas: Se basan en el procesamiento de las etiquetas de las entidades, sus comentarios y otras propiedades textuales como las etiquetas relacionadas. Estas técnicas provienen del procesamiento del lenguaje natural y de la recuperación de información. Incluyen: el uso de las etiquetas, por ejemplo calculando distancias entre cadenas; el uso de la ontología como un corpus, por ejemplo mediante medidas estadísticas basadas en la frecuencia de la ocurrencia de un término; o el uso de recursos externos, por ejemplo diccionarios.
- Técnicas estructurales: Se centran en las relaciones entre las entidades. Estas relaciones pueden ser entre conceptos y sus atributos o relaciones entre entidades. Abarcan la comparación entre tipos de relación, así como otras

técnicas para la generación de grafos: árboles de distancias, etc.

- Técnicas extensionales: Realizan una comparación entre las relaciones que posee cada entidad, esto es, su extensión. La extensión puede estar definida en función de las instancias o de recursos relacionados como, por ejemplo, documentos indizados. En el caso de que dos entidades no indiquen el mismo conjunto de documentos, se establece una función de similitud entre sus extensiones. Estas técnicas provienen del análisis de datos y la estadística.
- Técnicas semánticas: Se fundamentan en la semántica proporcionada por las ontologías. Utilizan conocimiento formalizado, interno o externo, y validadores de teoremas para encontrar las inconsistencias de un determinado alineamiento. Pueden ser utilizadas para extender el alineamiento o para detectar correspondencias conflictivas. Dentro de esta categoría se encuentra el uso de ontologías externas, típicamente genéricas, que proporcionan conocimiento común, facilitando así el proceso de correspondencia (Marrero, M. et al., 2006).

Los alineamientos de calidad son recursos escasos, debido a la complejidad del proceso de correspondencia, pero necesarios para la integración de ontologías heterogéneas, como se comentaba anteriormente. Por este motivo, una vez generado el alineamiento es preciso definir su gestión. El objetivo es mantener y compartir el alineamiento, de modo que evolucione a lo largo del ciclo de vida de la ontología, al tiempo que queda disponible para su uso en distintas aplicaciones. Esta gestión implica proporcionar soporte a las fases del ciclo de vida de los alineamientos, que pueden resumirse en:

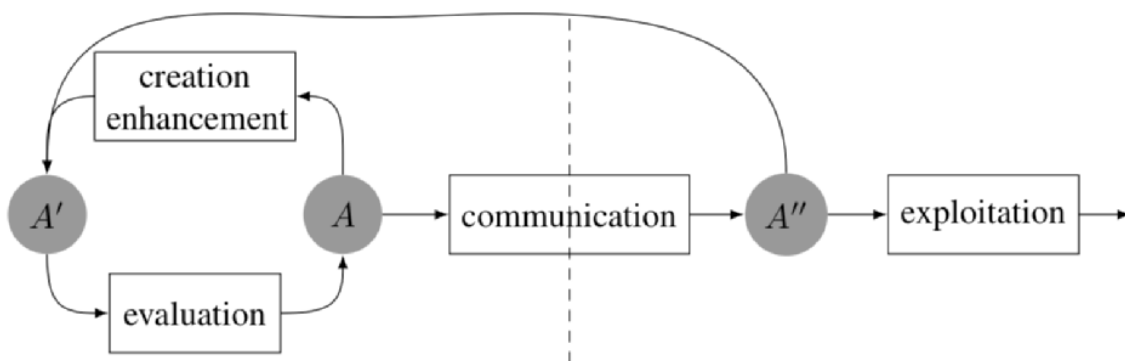


Figura II-6: Ciclo de vida de los alineamientos, tomado de (Euzenat, J.; Shvaiko, P., 2007).

Como representa la figura anterior, el alineamiento es creado (*creation*) a través de un proceso de correspondencia, que puede ser manual. A continuación entra en un bucle que incluye las tareas de evaluación (*evaluation*) y mejora (*enhancement*). La evaluación puede realizarse de forma manual o automática, y tiene como objetivo valorar la calidad del alineamiento obtenido. La mejora puede realizarse mediante cambios manuales en el alineamiento o procedimientos de refinamiento, por ejemplo, haciendo uso de umbrales y el grado de confianza de las correspondencias. Una vez estabilizado, el alineamiento debe ser almacenado, compartido y publicitado (*communication*) para su uso y reutilización. El alineamiento puede ser transformado o interpretado a través de acciones como mediación o fusión (*exploitation*). De este modo, es posible realizar la fusión o la mezcla de las ontologías alineadas (Kalfoglou, Y.;

Schorlemmer, M., 2003), esto es, la mínima unión de los vocabularios que permite: su uso combinado haciendo uso del alineamiento, por ejemplo en procesos de traducción; o la generación de una ontología global que incluya las ontologías originales y sus relaciones, obtenidas del alineamiento, por ejemplo para recuperación de información.

Junto con el ciclo de vida de los alineamientos, debe tenerse en cuenta que éstos evolucionan en paralelo con la ontología. De este modo, cuando una ontología sufre una modificación los alineamientos deben ser actualizados, conforme a los cambios realizados. Esto puede conseguirse mediante el registro (*traza*) de los cambios de la ontología y su aplicación a los alineamientos en los que participa. Así, la nueva versión de los alineamientos será generada a partir de la última versión, añadiendo el conjunto de cambios derivados de las modificaciones en la ontología.

Definido el ciclo de vida de los alineamientos, su gestión implica proporcionar soporte a cada una de las fases que lo componen (Institut National de Recherche en Informatique et Automatique, 2010). Las funciones requeridas para llevar a cabo la gestión pueden resumirse en: correspondencia entre dos ontologías, automática o manual; almacenamiento persistente; recuperación de un alineamiento mediante su identificador; recuperación de los metadatos de un alineamiento, por ejemplo, para comprobar su versión; eliminación de un alineamiento; recuperación de alineamientos entre dos ontologías; edición de un alineamiento; poda de un alineamiento o eliminación de correspondencias, por ejemplo, cuando su grado de confianza no supere un umbral establecido; o la generación de código que permita la transformación o traducción de datos en base a un alineamiento.

Respecto a las herramientas, existen múltiples clasificaciones y revisiones de los sistemas existentes actualmente (Shvaiko, P.; Euzenat, J., 2005), (Choi, N.; Song, I.; Han, H., 2006) o (Kalfoglou, Y.; Schorlemmer, M., 2003). Algunos de los más destacados son: *Similarity Flooding (SF)* (Melnik, S.; Garcia-Molina, H.; Rahm, E., 2002) que utiliza un algoritmo de correspondencia híbrido, terminológico y estructural, comenzando por un alineamiento basado en comparación de cadenas, que es refinado mediante un algoritmo de propagación de coeficientes de similitud; *Artemis (Analysis of Requirements: Tool Environment for Multiple Information Systems)* (Castano, S.; Antonellis, V.D., 2001) que realiza un análisis de afinidad en base a las etiquetas, la estructura y la proximidad con un tesoro común; *Cupid* (Madhavan, J.; Bernstein, P.A.; Rahm, E., 2001) también realiza la correspondencia de forma híbrida, incorporando técnicas lingüísticas y estructurales para, a continuación, calcular coeficientes de similitud con un tesoro de dominio específico; *COMA (COmbination of MAtching algorithms)* (Do, H.; Rahm, E., 2002) proporciona una librería extensible de algoritmos de correspondencia con seis algoritmos elementales, cinco híbridos y uno orientado a la reutilización, además, incluye un marco para la combinación de resultados y una plataforma para la evaluación de la efectividad; *NOM (Naive Ontology Mapping)* (Ehrig, M.; Sure, Y., 2004) proporciona un marco similar a COMA, pero incluye un conjunto de algoritmos basados en reglas y otro basado en técnicas extensionales; *QOM (Quick Ontology Mapping)* (Ehrig, M.; Staab, S., 2004) es un sucesor de NOM que prima la eficiencia en la generación de alineamientos frente a la calidad, fundamentado en que una pérdida marginal en la calidad implica una gran mejora en la eficiencia, lo que demuestra en comparación con sistemas similares; *OLA (OWL Lite Aligner)* (Euzenat, J.; Valtchev, P., 2004) utiliza un conjunto de algoritmos semánticos que

calculan y minimizan las distancias entre todos los elementos incluidos en las ontologías; *Anchor-PROMPT* (Musen, M.A.; Noy, N.F., 2001) es una extensión de PROMPT, también llamada SMART, que utiliza técnicas terminológicas en conjunción con las correspondencias establecidas por el usuario o por otro algoritmo; *S-Match* (Giunchiglia, F.; Shvaiko, P.; Yatskevich, M., 2004) utiliza técnicas estructurales generando las relaciones semánticas entre dos grafos generados a partir de las ontologías a relacionar. Otras herramientas como *Snoogle* (SemWebCentral, 2007) o el plugin *Prompt* de Protégé (Noy, N., 2009) permiten realizar correspondencia manual entre ontologías.

2.4 Recuperación de documentos XML

Existe un gran número de trabajos sobre recuperación de documentos XML (Christophides, V. et al., 2003). Algunos de estos trabajos tratan sobre lenguajes de recuperación propios de XML, como *XQL* (Derksen, E. et al., 1999). Una de sus versiones, *XIRQL* (Fuhr, N.; Großjohann, K., 2000), incluso permite asignar pesos a los términos, búsquedas por relevancia y tipos de datos. Este apartado se centra en métodos de recuperación de documentos XML.

- **Motores XML:** La mayoría de los programas son aplicaciones de pago y pensadas para empresas. Se puede consultar un listado pormenorizado en *XML Searching Resources* (SearchTools.com, 2009). En este caso, los productos se pueden clasificar en función de dos ámbitos: intranet y extranet. Los primeros tienen como objetivo el ámbito de la red interna a la empresa, redes corporativas. Los segundos se dirigen a la red global, la Web. Como ejemplo de los buscadores XML en el ámbito de la intranet están *Quizx* (XMLmind, 2010), *XML-Server* de *MarkLogic* (MarkLogic, 2010), *Xindice* (Apache Software Foundation, 2007) o *Repository* de *XMLMedia* (E-XMLMedia, 2002). En el ámbito de la extranet, algunos ejemplos representativos son: *IxiaSoft* (IXIASOFT, 2008), donde la indización se hace mediante un índice creado por el usuario, además puede seleccionar qué elementos o atributos serán indizados y qué etiquetas serán asociadas bajo un mismo concepto; *X3 XML Search Engine* (Docsoft, 2010), que realiza la indización basada en etiquetas y no en documentos, obteniendo así una consulta contextual; *XIRCUS* (Meyer, H. et al., 2002) trata de responder a consultas poco especificadas (sobre la estructura XML) y consultas exactas (texto en elementos, atributos o hiperenlaces) (Meyer, H. et al., 2002), aunque no puede recuperar espacios de nombres, atributos o Xlink), el algoritmo de posicionamiento está basado en tf-IDF (Morato, J.; Sánchez, S.; Valiente, M.C., 2005), (Bao, Z. et al., 2009), (Guo, L. et al., 2003), (Schlieder, T.; Meuss, H., 2002).
- **Directorios de DTDs/Esquemas:** No son frecuentes los directorios con este tipo de material, destacando *XML.ORG Registry* (XML.ORG, 2008) centrada en comercio electrónico, *XMLpitstop* (XML Pitstop, 2010) o *ZVON* (ZVON.org, 2010), uno de los primeros directorios de este tipo, relanzado ahora en 2010.
- **Motores de metadatos:** Aunque algo más frecuentes, suelen estar reducidos a un único vocabulario de metadatos, normalmente Dublin Core. Destacan en esta categoría *SemanticWebSearch* (Intellidimension, 2007) que busca en lenguajes

de ontologías y etiquetas META; *HotMeta* (DTSC, 1998) que realiza búsquedas en un conjunto restringido de metadatos; *Semantic Blogging Demonstrator* (Hewlett-Packard Labs, 2003) que busca por un conjunto de metadatos más las cuestiones qué, quién, por qué, dónde y *Learning About Learning Objects* (Botts, N., 2008). Además, también habría que mencionar los registros de metadatos citados en el apartado 2.2.1 *Esquemas de metadatos* y *NBII Clearinghouse Search* (National Biological Information Infrastructure, 2010), que permite realizar búsquedas por valores de un gran conjunto de metadatos.

- **Directorios de metadatos:** Se trata de recursos que recopilan instancias de metadatos con el fin de desambiguar o anotar documentos. Como ejemplo significativo se encuentra el *UKOLN Metadata Resources* (UKOLN, 2003). Aunque algo distintos en su concepción pero muy relacionados estarían los *Public Subject Indicators de Topic Maps* (Techquila PSI, 2003) y (Ontopia, 2003).
- **Motores de Ontologías:** Durante los años 2003 y 2004 irrumpieron un conjunto muy numeroso de este tipo de motores, como *OntoMap* (Linhalis, F.; Mattos Fortes, R.P.; Abreu Moreira, D., 2009) y *OntoWeb* (Hyun Hee, K., 2005). El fenómeno contrastó con el hecho de que la Web Semántica y los lenguajes que utilizaban eran de aparición reciente y en gran parte estaban en desarrollo. Su uso se encuentra actualmente muy centrado en la Web (Radhakrishnan, A., 2009), aunque la migración de estas tecnologías a las intranets abre perspectivas interesantes para su estudio. Los buscadores semánticos utilizan Procesamiento del Lenguaje Natural, Sistemas de Organización del Conocimiento y heurísticas basadas en el comportamiento del usuario para mejorar los resultados de las búsquedas (Sullivan, J., 2006). El resultado percibido en la Web ha sido el aumento de buscadores con un grado variable de usabilidad, según se acercan en mayor o menor medida al usuario. Algunos ejemplos son: *Ontaria* (World Wide Web Consortium, 2006b) de la W3C; *Semantic Web Search* (Intellidimension, 2007); *OwlSeek* (OwlSeek, 2009) incluye metadatos del LCC; *SchemaWeb* (SchemaWeb, 2008); *Swoogle* (Swoogle, 2007b) permite búsquedas por namespaces y lenguajes de ontologías, utiliza el *Swoogle Rank* para ordenar los resultados; en la línea de Swoogle se encuentran también *Falcons* (Cheng, G. et al., 2008), *Sindice* (Oren, E. et al., 2008) y *Watson* (Aquin, M. et al., 2007); *WebKB* (WebKB, 2003a) con una ontología propietaria para búsqueda conceptual basada en WN y accesible vía Web; *Simile* (Massachusetts Institute of Technology Libraries, 2008b) de W3C, HP y el MIT, está orientado a la recolección y publicación de metadatos en lenguajes de ontologías; *OWLIR* (EBiquity, 2010) permite la recuperación de documentos marcados semánticamente con DAML-OIL.

Como ejemplos de recuperadores que utilizan modelos de conocimiento, utilizando recursos lingüísticos estarían productos como *Swotti* (centrado en análisis de opinión) (Buzztrend, 2010), *Evri* (basado en entidades del nombre) (Evri Inc., 2010), *Powerset* (Microsoft, 2010a) o *Natural Finder* de Bitext (con PLN) (Bitext Innovations, 2010). En el último año, han aparecido tres buscadores con mayor potencia semántica *Wolfram-Alpha* (Wolfram Alpha

LLC, 2010), *Google Squared* (Google, 2010) y *MSN Bing* (Microsoft, 2010b).

- **Directorios de ontologías:** Algunos ejemplos de este tipo de recursos son: *SemWebCentral* (SemWebCentral, 2010); *DAML Ontology Library* (DAML.org, 2004); *Library of Ontologies* (Stanford KSL Network Services, 2001b); *WebOnto ontology library* (WebOnto, 1998) o *WebODE ontology library* (WebODE, 2002). Destacan, en el ámbito de la Biomedicina, *Open Biological and Biomedical Ontologies* (OBO Foundry, 2010) y *NCBO Bioportal* (The Board of Trustees of Leland Stanford Junior University, 2010), donde se están realizando importantes esfuerzos de normalización y correspondencia entre ontologías. Como resultado se está realizando un importante uso de las ontologías que albergan, en tareas como anotación y recuperación.

2.5 Marco teórico y herramientas del Proceso de Desarrollo

Tal y como la Ingeniería del Software establece, el diseño del sistema informático, que proporcionará soporte a la propuesta, deberá realizarse conforme a un proceso de desarrollo bien establecido. Este proceso define, tanto las fases y actividades a realizar para la consecución del proyecto, como el conjunto de documentos asociados al mismo. El objetivo es diseñar un producto de calidad, esto es: que satisfaga el conjunto de funcionalidades que se establezcan (*corrección*), incluyendo la documentación relacionada que facilite el mantenimiento y actualizaciones posteriores (*mantenibilidad*), con el mínimo posible de fallos (*fiabilidad*).

En esta sección se definirá y mostrarán las actividades del proceso de desarrollo software, así como su organización en modelos de ciclo de vida. Se describirá brevemente el lenguaje de modelado unificado y, por último, se incluirá un resumen de los estilos arquitectónicos y patrones de diseño utilizados en el diseño de la plataforma informática de soporte a la propuesta.

2.5.1 Fundamentos

La idea de aplicar el concepto de proceso al desarrollo software proviene del campo de la fabricación, donde los procesos o pasos en la fabricación están definidos y se controlan de manera continua. Este enfoque se puede aplicar al mundo del software para la gestión del proceso a nivel de proyecto y para mejorar las capacidades de los grupos de desarrollo.

Según Cuevas (Cuevas, G. et al., 2003), el proceso software establece el marco de trabajo tanto técnico como de gestión para poder aplicar los métodos, herramientas y personas a la tarea de desarrollo de software. La definición del proceso identifica los roles y las tareas específicas y establece medidas para el control de ejecución de cada paso.

La definición adecuada del proceso permitirá a una organización dedicada del desarrollo de software, asegurar que cada elemento de trabajo se asigna apropiadamente y que se conoce su estado en cada momento. A su vez, esta definición oficial del proceso en la organización, permitirá incorporar nuevos métodos que lo mejoren.

Un proceso definido permite que cada nuevo proyecto sea construido en base a la propia experiencia y la de sus predecesores. Los problemas recogidos a medida que se utiliza el proceso permiten identificar sus causas y corregirlos. De este modo, tanto el proceso, como su definición y la infraestructura de soporte evolucionarán con la experiencia.

En la *Figura II-7* se muestra el proceso donde se han incorporado, a las de desarrollo, las actividades de gestión: gestión de requisitos, planificación, gestión de la configuración y gestión de calidad.

2.5.2 Actividades de gestión del Proceso Software

Las actividades de gestión están orientadas a controlar el desarrollo del proceso software y a corregir las desviaciones con respecto a los parámetros de calidad establecidos. De este modo, se trata de tareas que se desarrollan a lo largo del ciclo de vida del software.

Una vez obtenida la especificación de requisitos, su gestión se ocupa de fijar el marco de referencia, de carácter contractual, que determinará el desarrollo. Incluirá todas las actividades de control que mantengan la integridad y exactitud del acuerdo sobre los requisitos, a medida que progrese el proyecto. Estas medidas incluirán la gestión de las modificaciones sobre los requisitos, surgidas durante el desarrollo, con el fin de garantizar el tratamiento y control de estas actualizaciones. La correcta gestión de las modificaciones es fundamental para poder realizar el seguimiento de los requisitos en las distintas fases del ciclo de vida. Esta característica, denominada *trazabilidad*, es fundamental para el desarrollo del proyecto y su posterior mantenimiento.

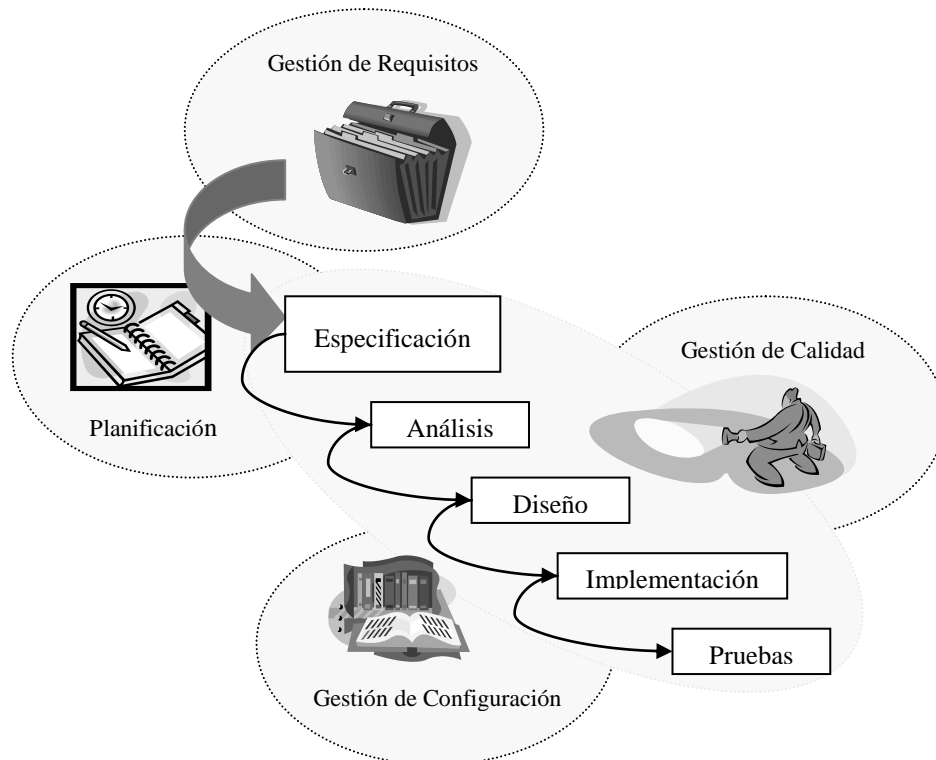


Figura II-7: Representación un proceso de desarrollo software.

La gestión de configuración del software es el medio que permite conocer, en todo momento, qué componentes y versiones, tanto de un producto como de sus elementos, son las correctas. De forma más concreta, la gestión de configuración es el proceso de identificar y definir los elementos de la configuración¹² para:

- Controlar la liberación de versiones y cambios durante todo el ciclo de vida.
- Registrar e informar de su estado y de las peticiones de cambios.
- Verificar la corrección y acabado de los elementos.

Una vez establecidos los objetivos de calidad de un proyecto software y especificados aquellos procesos operativos y recursos necesarios para satisfacer dichos objetivos, la gestión de la calidad será el conjunto de actividades diseñadas para evaluar el proceso por el cual se desarrollan los productos. Así, deberá garantizar que el sistema, componente o proceso cumple los requisitos especificados y cubre las necesidades o expectativas del cliente o usuario.

Por último, la planificación del proyecto consistirá en establecer la estructura temporal de las fases, actividades y tareas del proyecto, en función de los recursos de que se disponga. El seguimiento y control de evolución determinarán la necesidad de nuevas planificaciones que puedan alterar la fecha de finalización de proyecto.

2.5.3 Ciclo de vida del software

El ciclo de vida software es el periodo que comienza cuando un producto software es concebido y termina cuando deja de estar disponible. El ciclo de vida se divide, normalmente, en fases que estructuran y organizan las etapas de concepción, desarrollo y mantenimiento.

Un modelo de ciclo de vida es la descripción de las distintas formas de desarrollo de un proyecto, es decir, la orientación que debe seguirse para obtener, a partir de los requerimientos del cliente, sistemas que puedan ser utilizados por dicho cliente. A continuación se describirán los modelos de ciclo de vida en cascada y en espiral por su relación con el presente trabajo.

2.5.3.1 Ciclo de vida en cascada

El ciclo de vida en cascada es el ciclo más simple. Históricamente, fue el primer ciclo en aparecer. La vida de los sistemas pasa por una serie de fases consecutivas y separables. Las fases se desarrollan en secuencia y sólo una vez, aunque puede haber iteraciones dentro de cada fase. En la *Figura II-8* se esquematiza el ciclo de vida en cascada con cada una de sus fases.

¹² Elemento de configuración es un conjunto software y, opcionalmente hardware, tratado como una unidad por la gestión de configuración.

El modelo de ciclo de vida se escoge en la planificación inicial del proyecto. Cada proyecto se estructura en actividades, que serán previamente planificadas y agrupadas en fases. Además, los productos deben ser terminados y entregados, de acuerdo a la planificación, y funcionar correctamente.

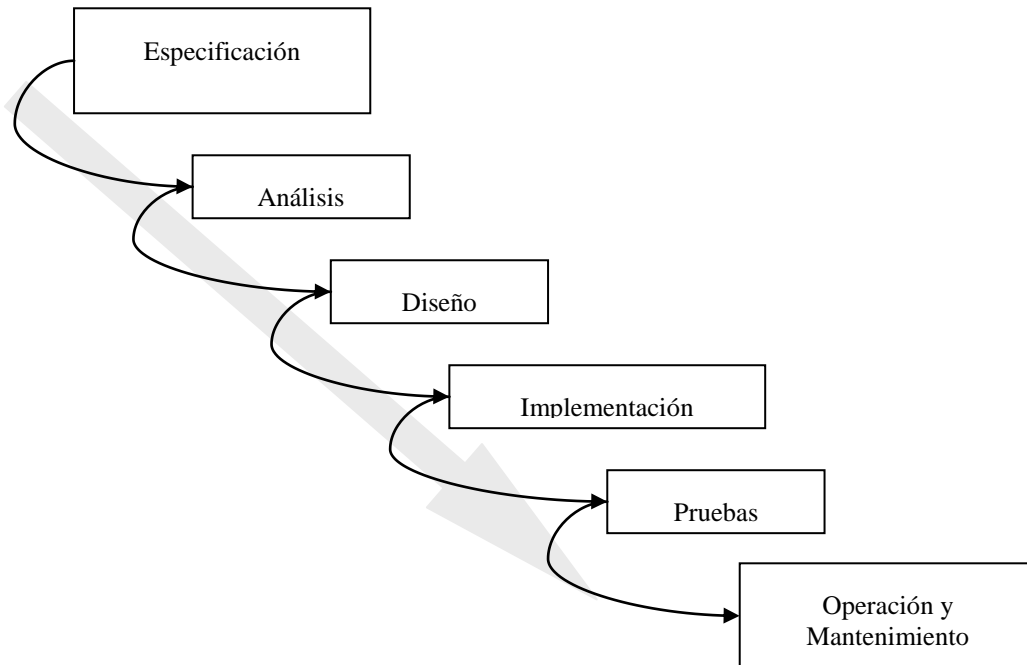


Figura II-8: Ciclo de vida en cascada.

Todo ciclo de vida debe incluir las fases incluidas en la siguiente tabla¹³.

FASE	DEFINICIÓN	DOCUMENTACIÓN ASOCIADA
UR	Definición de Requerimientos de Usuario	URD
SR	Definición de Requerimientos del Software	SRD
AD	Diseño de la Arquitectura	ADD
DD	Diseño Detallado y codificación	DDD, SUM, CODE
TR	Transferencia del software	STD
OM	Operación y mantenimiento	PHD

Tabla II-7: Fases de un ciclo de vida.

Las primeras cuatro fases finalizan con una revisión. Estas fases han de aparecer siempre, independientemente del tamaño, tipo, software empleado, o equipo de desarrollo, ya sea propio o externo.

¹³ Según el estándar PSS-05-0 Issue 2, definido por la Agencia Espacial Europea (European Space Agency, 2010)

El ciclo de vida del software comienza realmente con la entrega del Documento de Requisitos de Usuario (URD, *User Requirements Document*) a los usuarios. Por lo tanto, la revisión del URD es la primera actividad del ciclo de vida. Junto con esta revisión, ha de producirse la entrega de un plan de gestión de proyecto, incluyendo una planificación y una estimación de costes para el proyecto.

Los entregables de cada fase deben de ser revisados y aprobados por el usuario antes de proceder con la siguiente fase y pueden tener carácter contractual.

Hay seis hitos que indican el progreso del ciclo de vida:

- Aprobación del URD.
- Aprobación del Documento de Requisitos de Software (*Software Requirements Document: SRD*).
- Aprobación del Documento de Diseño Arquitectónico (*Architectural Design Document: ADD*).
- Aprobación del Documento de Diseño Detallado (*Detailed Design Document: DDD*), Manual de Usuario del Software (*Software User Manual: SUM*) y Código, así como confirmación de disponibilidad para las pruebas de aceptación provisional.
- Entrega del Documento de Transferencia del Software (*Software Transfer Document: STD*) y confirmación de aceptación provisional.
- Entrega del Documento Histórico del Proyecto (*Project History Document: PHD*) y confirmación de aceptación final.

A continuación se describen las fases del ciclo de vida del software.

- **Definición de requisitos del usuario:** En esta fase se trata de plantear el problema que deberá ser solucionado mediante un sistema software. Para ello, se realizan entrevistas con el cliente, de forma que se obtienen los requisitos que debe tener el sistema para resolver sus necesidades.
- **Definición de requisitos del software:** Esta es la fase de análisis del problema. A partir de las necesidades que han originado la puesta en marcha del proyecto, se trata de comprender qué es lo que tiene que realizar el sistema en su conjunto. Se analizará el entorno en el que se desenvolverá el sistema y la manera en que éste interactuará con él.
- **Diseño de la arquitectura:** Esta fase corresponde al planteamiento de la solución. El sistema se descompone en un conjunto de subsistemas, de manera que se simplifica el sistema global. Se definen las relaciones que tienen cada uno de los subsistemas con los demás.
- **Diseño detallado y codificación:** En esta fase se refina el diseño realizado en la fase anterior. Además se realiza la implementación del sistema, se valida y se

verifica, de manera que el sistema realizado cumpla con los requisitos definidos en las fases iniciales.

- **Transferencia del software:** En esta fase, el sistema es transferido a su entorno de explotación. Puede haber un periodo de enseñanza para que el usuario conozca la manera de utilizar el sistema.
- **Operación y mantenimiento:** Esta suele ser la fase más larga del ciclo de vida. Transcurre durante el uso del sistema en su entorno de explotación. Además, incluye las actuaciones de mantenimiento y mejora del sistema.

2.5.3.2 Ciclo de vida en espiral

En el modelo de proceso en espiral el proyecto se ataca en una serie de ciclos de vida cortos cada uno de los cuales finalizará con una revisión de software ejecutable. En la *Figura II-9* se muestra un ejemplo de desarrollo completado en cinco iteraciones del ciclo: Análisis, diseño, implementación y pruebas (Ariadne Training, 2005).

Este ciclo de vida es especialmente recomendable cuando se planea hacer múltiples entregas del software, es decir, existen fases OM independientes. Todas las fases del ciclo se cumplen para producir cada entrega, que incorpora las experiencias de las entregas anteriores, con sucesivas etapas de:

- Especificación.
- Construcción.
- Explotación.
- Revisión para obtener un refinamiento del producto de cara al siguiente ciclo.

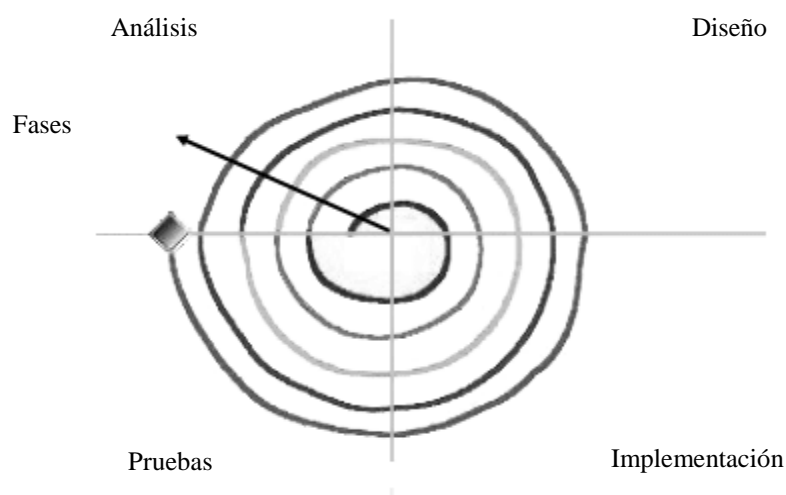


Figura II-9: Ciclo de vida en espiral.

Cada una de las iteraciones aporta un mayor grado de aproximación al resultado final. Este tipo de ciclo de vida se usa en las siguientes situaciones:

- En grandes proyectos en tamaño o duración, como varios meses de especificación, y varios años de desarrollo, que hacen que las organizaciones que los desarrollan o que los vayan a usar no sean estables.
- En proyectos especiales, como tecnologías muy avanzadas o poco habituales.
- Cuando se requiere la experiencia del usuario para refinar un diseño.
- Cuando hay impedimentos de disponibilidad tecnológica en el momento actual.
- Cuando se prevé que aparezcan nuevos requerimientos de usuario.
- Algunos requerimientos son más complejos que otros y pueden retrasar todo el desarrollo.

Generalmente, este modelo de ciclo de vida está basado en el uso de técnicas de prototipado rápido e incremental. Un prototipo implementa aspectos clave seleccionados del software a desarrollar, por ejemplo: una interfaz de usuario.

Aunque presenta numerosas ventajas, este ciclo de vida tiene implica algunos riesgos:

- Si las especificaciones iniciales no son completas, la estructura inicial del software puede no soportar el peso de la evolución posterior.
- Puede ser necesario reescribir, con gran coste, partes completas de código.
- Las soluciones temporales (o los prototipos) se convierten en definitivas, incluso embebidas dentro de segmentos de código mayor. Esta práctica es poco recomendable, teniendo en cuenta que el objetivo del prototipo es un desarrollo rápido con el menor coste posible. Incluir un prototipo en la solución definitiva implica menor calidad (diseño, código, documentación, etc.) en el producto desarrollado.

2.5.4 El Lenguaje de Modelado Unificado: UML

En este apartado se describirá de manera general el lenguaje de modelado UML. Aunque actualmente la versión del estándar es la 2.1.1 (Object Management Group, 2006) se ha seguido la notación de la versión 1.4.2 (International Organization for Standardization, 2001) al estar mejor soportado por las herramientas de modelado. Tras una breve introducción, se mostrarán los principales diagramas de modelado.

2.5.4.1 Introducción

El lenguaje de modelado unificado UML (*Unified Modeling Language*) es un lenguaje estándar, con el que es posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Un aspecto importante del modelo es que no pretende definir un

estándar de desarrollo, sino únicamente un lenguaje de modelado. Otros métodos de modelado, como *OMT (Object Modeling Technique)* (Rumbaugh, J.; Firesmith, D., 1996), sí definen procesos concretos. En UML, los procesos de desarrollo son diferentes según los distintos dominios de trabajo. UML recomienda utilizar los procesos que otras metodologías tengan definidos.

En la definición de UML se estableció, como objetivo principal, la consecución de un método que aunara los mejores aspectos de los lenguajes predecesores. Para ello, se plantearon las siguientes características:

- El método debía ser capaz de modelar no sólo sistemas de software, sino otro tipo de sistemas reales de la empresa, siempre utilizando los conceptos de la orientación a objetos (OO).
- El lenguaje de modelado que se pretendía definir, debía poder ser utilizado tanto por máquinas como por personas.
- Debía establecer un acoplamiento explícito de los conceptos y los artefactos ejecutables.
- Debía manejar los problemas típicos de los sistemas complejos de tiempo real.

Con estos requisitos se pretendía lograr que los lenguajes que se aplican, siguiendo los métodos más utilizados, sigan evolucionando en conjunto y no por separado. Además, se unificarían las perspectivas entre diferentes tipos de sistemas, no sólo software, al facilitar la comprensión de las fases de desarrollo, los requerimientos de análisis, el diseño, la implementación y los conceptos propios de la orientación a objetos.

El lenguaje UML es una técnica de modelado de objetos, y como tal, supone una abstracción de un sistema para llegar a construirlo en términos concretos. El modelado no es más que la construcción de un modelo a partir de una especificación. Un modelo es una abstracción de la realidad que se quiere representar, el cual se elabora para comprender la realidad antes de construir el sistema. El modelo omite detalles que no resultan esenciales para la comprensión del original, por lo que facilita la interpretación. Los modelos, además, permiten comprobar más fácilmente los sistemas que se modelan y detectar los posibles errores.

Para representar la realidad, se utilizan diversos diagramas que ofrecen distintos puntos de vista de esta realidad. Un diagrama es una representación gráfica de una colección de elementos del modelo.

En los siguientes apartados se describirán brevemente cada uno de los diagramas que componen UML.

2.5.4.2 Diagrama de Casos de Uso

Un *Caso de Uso* es una secuencia de transacciones, desarrolladas por un sistema, en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema,

mediante su interacción con los usuarios u otros sistemas. Esto quiere decir que estos diagramas muestran la relación entre los actores y los casos de uso. Una relación es una conexión entre los elementos del modelo. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema, al mostrar la reacción producida como respuesta a los eventos que se producen en el mismo. En la *Figura II-10* se muestra un ejemplo.

En este tipo de diagrama intervienen algunos conceptos nuevos. Un **actor** es una entidad externa al sistema, que se modela y que puede interactuar con él. Un ejemplo de actor, podría ser un usuario o cualquier otro sistema.

Las relaciones entre casos de uso y actores pueden ser las siguientes:

- Un actor se comunica con un caso de uso.
- Un caso de uso extiende otro caso de uso.
- Un caso de uso usa otro caso de uso.

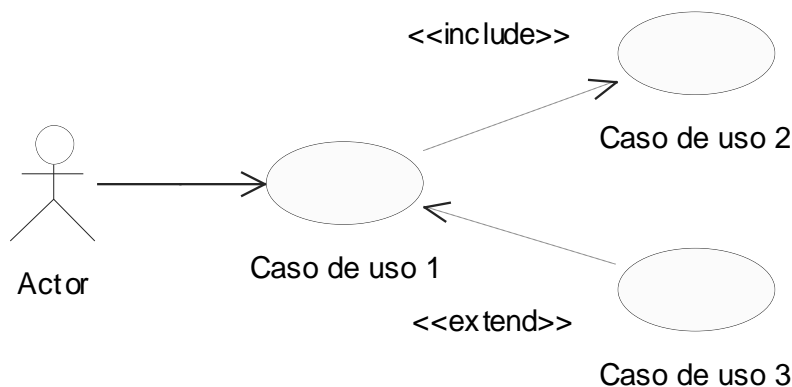


Figura II-10: Diagrama ejemplo de casos de uso.

2.5.4.3 Diagrama de Clases

Los diagramas de clases representan un conjunto de elementos del modelo que son estáticos, como las clases y los tipos, sus contenidos y las relaciones que se establecen entre ellos.

Algunos de los elementos que se pueden clasificar como estáticos en UML, son los siguientes:

- **Paquete:** Es el mecanismo de que dispone UML para organizar sus elementos en grupos.

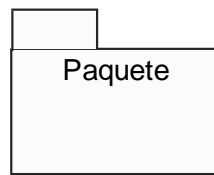


Figura II-11. Representación gráfica de un paquete.

- **Clase:** Describe un conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y significado. Los componentes de una clase son:
 - **Atributo:** Se corresponde con las propiedades de una clase.
 - **Operación:** También conocido como **método o función**, es un servicio proporcionado por la clase que puede ser solicitado por otras clases y que produce un comportamiento en ellas cuando se realiza.

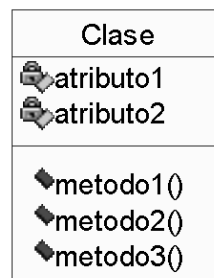


Figura II-12. Representación gráfica de clase.

- **Interfaz:** Representa la funcionalidad que proporciona uno o varios elementos del modelo al resto de elementos. Las interfaces no tienen asociado un comportamiento, por lo que tendrá que ser implementado por otro elemento.

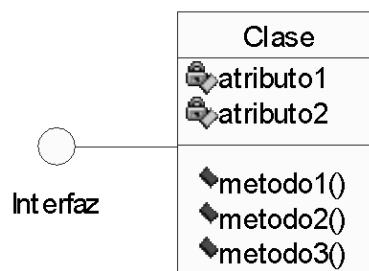


Figura II-13. Representación gráfica de una interfaz junto a la clase que la realiza.

En el diagrama de clases se muestra, además, las relaciones estructurales existentes entre los elementos descritos anteriormente. Un ejemplo de un diagrama de clases completo, podría ser el de la *Figura II-14*.

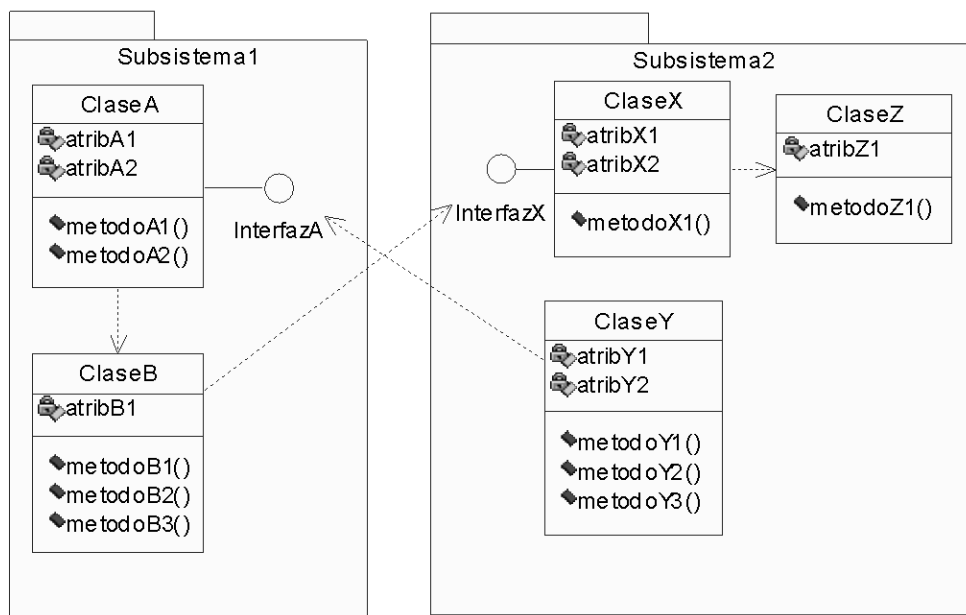


Figura II-14. Ejemplo de diagrama de clases.

2.5.4.4 Diagrama de Estado

Al igual de los descritos a continuación, forman parte de la representación dinámica del sistema, por lo que complementan a los anteriores, centrados en la descripción de la parte estática del mismo.

Los *diagramas de estado* representan la secuencia de los posibles estados por los que un objeto, o una interacción entre objetos, evolucionan durante su tiempo de vida, en respuesta a los eventos recibidos. De este modo, modela el sistema como una máquina de estados.

Un *estado* en UML, se puede definir como la situación en que se encuentra un objeto o una interacción cuando satisface una condición, desarrolla alguna acción o se encuentra esperando un evento.

Cuando un objeto, o una interacción, pasan de un estado a otro, debido a un evento, se dice que ha sufrido una *transición*. Puede suceder que, con la aparición de un evento, el estado de origen y el de destino sea el mismo, teniendo así una transición reflexiva.

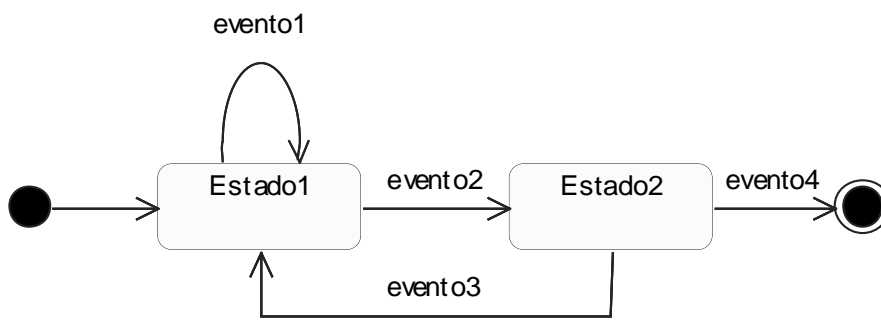


Figura II-15. Ejemplo de diagrama de estado.

2.5.4.5 Diagrama de Actividades

Son similares a los diagramas de flujo tradicionales. En realidad, se corresponden con un caso especial de los diagramas de estado, donde los estados son *estados de acción*, y las transiciones vienen provocadas por la finalización de las acciones que tienen lugar en los estados de origen. Es decir, estados con una acción interna y una o más transiciones que suceden al finalizar esta acción o, dicho de otro modo, un paso en la ejecución de lo que será un procedimiento.

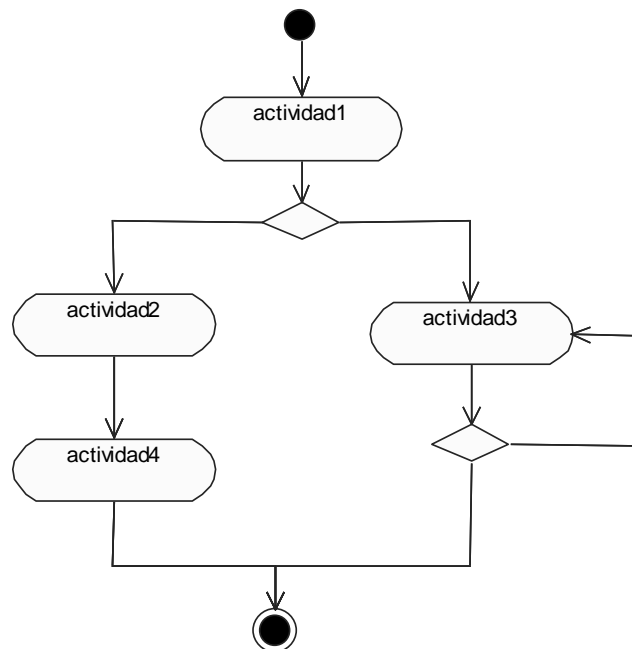


Figura II-16. Ejemplo de diagrama de actividades.

Generalmente, estos diagramas están asociados a la implementación de un caso de uso, o de un método. Los diagramas de actividad se utilizan para mostrar el flujo de

operaciones que se desencadenan en un procedimiento interno del sistema.

2.5.4.6 Diagrama de Secuencia

Estos diagramas al igual que los siguientes, los Diagramas de Colaboración, muestran las relaciones de interacción que tienen los objetos entre sí. También forman parte de la descripción dinámica del sistema incluyendo, además, las interacciones a lo largo del tiempo.

Ambos tipos de diagramas, secuencia y colaboración, son equivalentes. La diferencia entre ellos está en el mayor énfasis que el diagrama de secuencia pone en la representación del transcurso del tiempo, frente a la representación del diagrama de colaboración, donde se enfatiza más la representación de las relaciones de interacción.

Los diagramas de secuencia muestran las interacciones entre un conjunto de objetos, ordenadas según el momento del tiempo en que tienen lugar. El objeto puede existir sólo durante la ejecución de la interacción y puede ser creado o destruido durante la ejecución de la interacción. El diagrama de secuencia permite representar el periodo durante el que un objeto está desarrollando una acción directamente o a través de un procedimiento.

En este tipo de diagramas intervienen los *mensajes*, que son la forma en que se comunican los objetos. Los mensajes consisten en la solicitud, por parte de un objeto origen, de una operación de un objeto destino. Existen distintos tipos de mensajes según cómo se producen en el tiempo: simples, síncronos, y asíncronos.

Los diagramas de secuencia permiten indicar cuál es el momento en el que se envía o se completa un mensaje mediante el tiempo de transición, especificado en el diagrama.

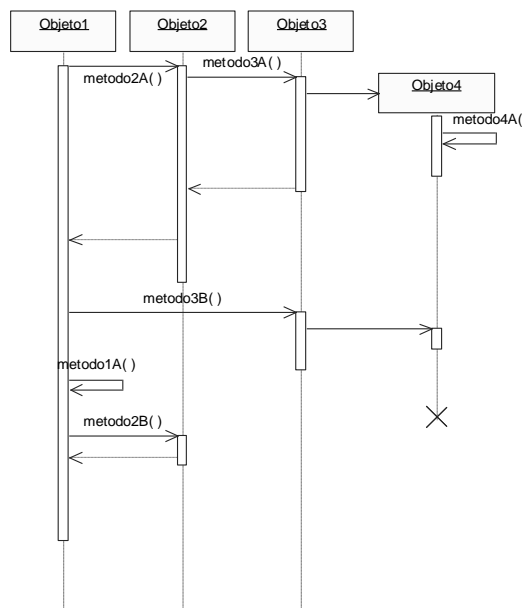


Figura II-17. Ejemplo de diagrama de secuencia.

2.5.4.7 Diagrama de colaboración.

Muestra la interacción entre varios objetos y los enlaces que existen entre ellos. Representa las interacciones entre objetos organizadas alrededor de los objetos y sus vinculaciones. A diferencia de un diagrama de secuencia, un diagrama de colaboraciones enfatiza las relaciones entre los objetos, no la secuencia en el tiempo en que se producen los mensajes.

Los componentes de los diagramas de colaboración son los objetos, enlaces y mensajes. Un *enlace* es una instancia de una asociación que conecta dos objetos de un diagrama de colaboración. El enlace puede ser reflexivo si conecta a un elemento consigo mismo. La existencia de un enlace entre dos objetos indica que puede existir un intercambio de mensajes entre los objetos conectados.

Los diagramas de interacción muestran el flujo de mensajes entre elementos del modelo. El flujo de mensajes representa el envío de un mensaje desde un objeto a otro, si entre ellos existe un enlace. Los mensajes que se envían entre objetos pueden ser de distintos tipos, también según como se produzcan en el tiempo. Los tipos de mensajes básicos son, al igual que en los diagramas de secuencia, simples, síncronos y asíncronos.

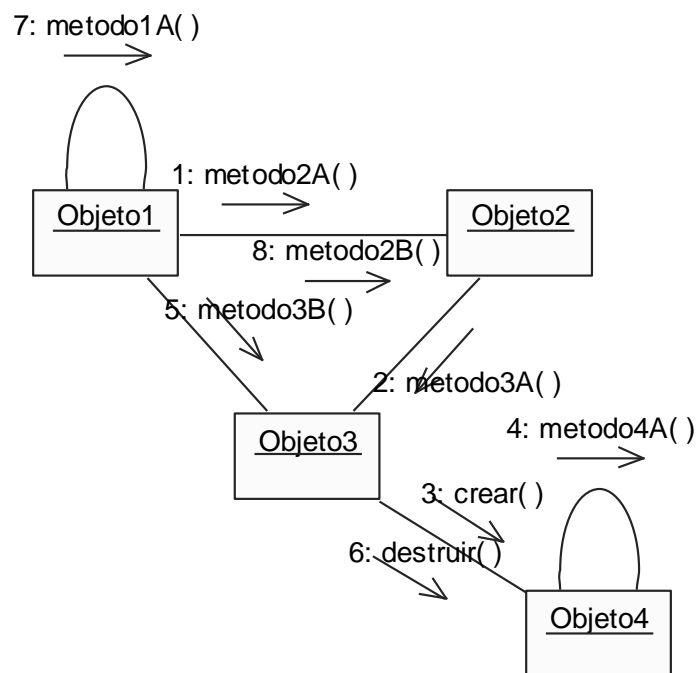


Figura II-18. Ejemplo de diagrama de colaboración.

2.5.4.8 Diagrama de Componentes

A diferencia del resto de diagramas, que expresan la lógica del sistema, el Diagrama de Componentes y el Diagrama de Despliegue modelan el sistema desde un punto de vista físico, entendiendo como tal: máquinas, redes, ficheros ejecutables y cualquier otro tipo de archivo que intervenga en el sistema.

Un componente es una pieza del sistema que se está representando. El diagrama de componentes muestra la dependencia entre los distintos componentes de software, incluyendo componentes de código fuente, binario y ejecutable.

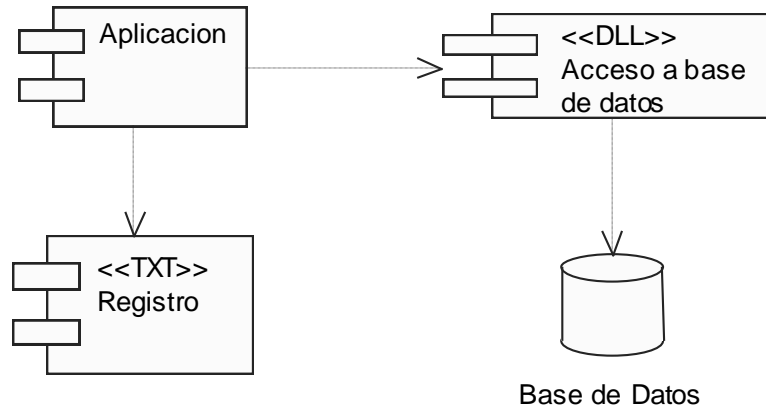


Figura II-19. Ejemplo de diagrama de componentes.

2.5.4.9 Diagrama de Despliegue

El Diagrama de Despliegue muestra la configuración de los componentes hardware, los procesos, los elementos de procesamiento en tiempo de ejecución y los objetos que existen en tiempo de ejecución. En este tipo de diagramas intervienen nodos, asociaciones de comunicación, componentes dentro de los nodos y objetos que se encuentran a su vez dentro de los componentes. Un *nodo* es un objeto físico en tiempo de ejecución, es decir, una máquina que se compone habitualmente de, al menos, memoria y capacidad de procesamiento. Un nodo puede, a su vez, puede estar formado por otros componentes.

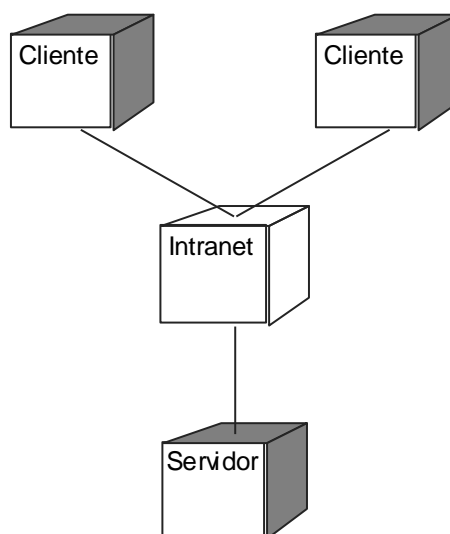


Figura II-20. Ejemplo de diagrama de despliegue.

2.5.5 Estilos arquitectónicos y patrones de diseño

Los estilos arquitectónicos y los patrones de diseño, aunque con distinto nivel de abstracción, son soluciones prácticas a un problema común y recurrente (Marinescu, F., 2002). Los estilos y patrones documentan y exponen un problema que puede suceder al diseñar o implementar una aplicación, proponiendo una solución práctica a dicho problema. En este apartado se describirán los principales estilos arquitectónicos y patrones de diseño aplicados en el planteamiento de la plataforma informática de soporte.

2.5.5.1 Modelo-Vista-Controlador (*Model-View-Controller*)

En general, una misma aplicación debe permitir acceso a múltiples usuarios utilizando diferentes tipos de interfaces. Por ejemplo, una vista HTML para clientes, una vista WML para clientes, una interfaz para aplicaciones *standalone* de administración y una vista en XML para servicios Web a proveedores, donde cada una de estas interfaces de acceso deben proporcionar los mismos datos a los clientes y ser parte de la misma lógica del negocio.

Aplicando el estilo arquitectónico Modelo-Vista-Controlador (MVC) se logra separar el modelo de negocio, de la presentación usada para ser visualizado y de la lógica utilizada para procesar acciones externas. Esta separación permite múltiples vistas sobre el mismo modelo de datos, lo que simplifica la implementación, al permitir separar roles de desarrollo, y el mantenimiento del sistema.

El patrón MVC está compuesto de tres elementos:

- **Modelo:** Representa los datos empresariales y las reglas para acceder a ellos y modificarlos.
- **Vista:** Especifica cómo deben ser mostrados los datos contenidos por el Modelo.
- **Controlador:** Transforma interacciones efectuadas en la Vista en acciones a ser ejecutadas por el Modelo.

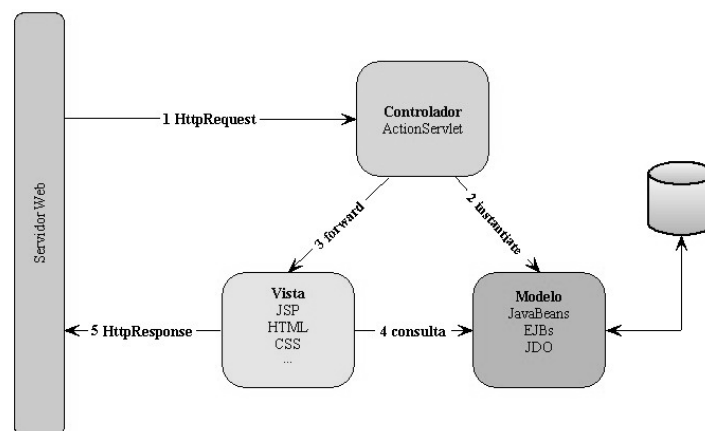


Figura II-21: Patrón arquitectónico MVC modelo 2.

2.5.5.2 Capas (*Layers*)

Una aplicación multicapa es una aplicación que ha sido dividida en múltiples componentes de aplicación. Las aplicaciones multicapa conllevan un gran número de ventajas significativas sobre las tradicionales arquitecturas cliente/servidor, incluyendo mejoras en: Escalabilidad, Fiabilidad, Usabilidad, Reutilización y Flexibilidad.

En una aplicación tradicional cliente/servidor, la aplicación cliente contiene la lógica de la presentación, la lógica de negocio, y la lógica de la manipulación de datos (conexiones y consultas SQL a bases de datos). El servidor es generalmente un gestor de bases de datos relacionales (que realmente no forma parte de la aplicación).

En cambio, en una arquitectura multicapa, el cliente de la aplicación contiene únicamente la lógica de la presentación. La lógica de los procesos y del acceso a datos se divide en componentes separados y son desplegados en uno o varios servidores.

Los componentes servidores pueden compartir recursos limitados, como procesos, conexiones a bases de datos, y sesiones de red. Si la demanda del sistema se ve incrementada, los componentes más utilizados pueden ser replicados y distribuidos a lo largo de múltiples sistemas. Los sistemas multicapa pueden construirse sin límites de escalabilidad. Si el diseño es eficiente, más o mayores servidores pueden añadirse al entorno para impulsar un mayor rendimiento y para soportar usuarios adicionales. Los sistemas multicapa pueden escalarse para soportar cientos de miles o millones de usuarios concurrentes.

Un entorno multicapa también es capaz de soportar varios niveles de redundancia. A través de la replicación y la distribución, una arquitectura multicapa elimina cualquier cuello de botella o puntos aislados de fallo. Este acercamiento multicapa proporciona una alta fiabilidad y una consistencia del sistema capaces de soportar operaciones críticas de negocio.

Una aplicación que utilice el estilo arquitectónico de capas es más sencilla de gestionar que las tradicionales aplicaciones cliente/servidor, ya que sólo una pequeña parte del código se despliega realmente en los sistemas del cliente. La mayoría de la lógica de la aplicación se despliega, gestiona y mantiene en los servidores.

La arquitectura de aplicaciones multicapa soporta sistemas de aplicaciones extremadamente flexibles. La mayoría de la lógica de la aplicación se implementa en pequeños componentes modulares. La lógica de procesos real en los componentes se encapsula detrás de una interfaz abstracta y bien definida. El código interno al componente puede ser modificado sin necesidad de realizar ningún cambio en la interfaz. Por lo tanto, puede modificarse un componente sin alterar al resto de componentes de la aplicación. Así, este tipo de aplicaciones pueden adaptarse con facilidad para reflejar los cambiantes requisitos de los procesos de negocio. Además, gracias a su interfaz bien definida, un componente es una pieza de software reutilizable. Cada componente realiza una serie de tareas específicas, que están publicadas y son accesibles desde cualquier otra aplicación a través de la interfaz. Una función cualquiera de negocio puede ser implementada una sola vez y ser reutilizada en otra aplicación que requiera esa misma función. Si una organización mantiene una adecuada librería de

componentes, el desarrollo de aplicaciones se convierte únicamente en una simple cuestión de ensamblar los componentes adecuados, de forma que realicen las funciones que son requeridas.

Un número cualquiera de entornos cliente puede acceder al mismo componente servidor a través de su interfaz. Un único sistema de aplicación multicapa puede dar soporte a una gran variedad de dispositivos clientes, incluyendo a las tradicionales estaciones de trabajo, clientes Web u otros clientes no tan usuales, como tarjetas inteligentes, asistentes personales de datos (*PDA*), etc.

Desafortunadamente, la construcción de aplicaciones multicapa no es tan sencilla como construir aplicaciones cliente/servidor. Las aplicaciones multicapa deben interactuar con una gran variedad de servicios *middleware*.

Dará soporte a la arquitectura MVC, cuya combinación se muestra en la siguiente figura:

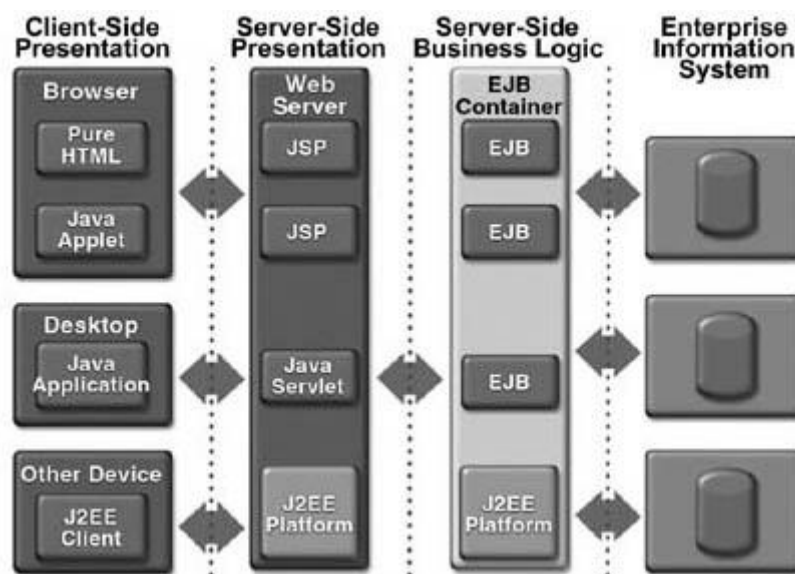


Figura II-22: Combinación de los estilos arquitectónicos Capas y MVC.

2.5.5.3 Value Object

Un patrón de diseño proporciona, como se comentaba al inicio de este apartado, una solución a un problema recurrente, concretamente, relacionado con el diseño detallado de componentes. El patrón de diseño *Value Object* (o *Transfer Object*) ofrece una solución compacta y organizada para transferir datos entre capas de una arquitectura.

Un Value Object no es más que un objeto que “empaqueta” datos para que puedan viajar entre las capas. Dicho objeto encapsulará datos procedentes de uno o varios objetos de negocio, que serán accesibles mediante métodos de acceso y asignación (*get* y *set*). Nótese que, aunque los Value Objects están directamente relacionados con los objetos del dominio, no se trata de los mismos objetos. Los objetos del dominio pueden contener lógica de negocio mientras que los Value Objects son meros almacenes de datos. Además, no tiene por qué haber una relación uno-a-uno entre objetos del dominio

y Value Objects. Se destacan algunos puntos importantes:

- Por cada objeto de negocio puede haber más de un Value Object. De hecho, una estrategia muy común es usar un Value Object distinto para cada caso de uso.
- Un problema importante es el de la sincronización entre los valores del Value Object y los del objeto del dominio que representa. Hay que asegurarse de que dichos valores están actualizados o que una falta de actualización de los mismos no conlleve consecuencias graves (caso típico de las operaciones de sólo lectura). En caso de usar los Value Object (VO) tanto para mostrar datos como para almacenarlos (VOs actualizables) habrá que tener sumo cuidado para sincronizar la información de los VOs que hayan cambiado.

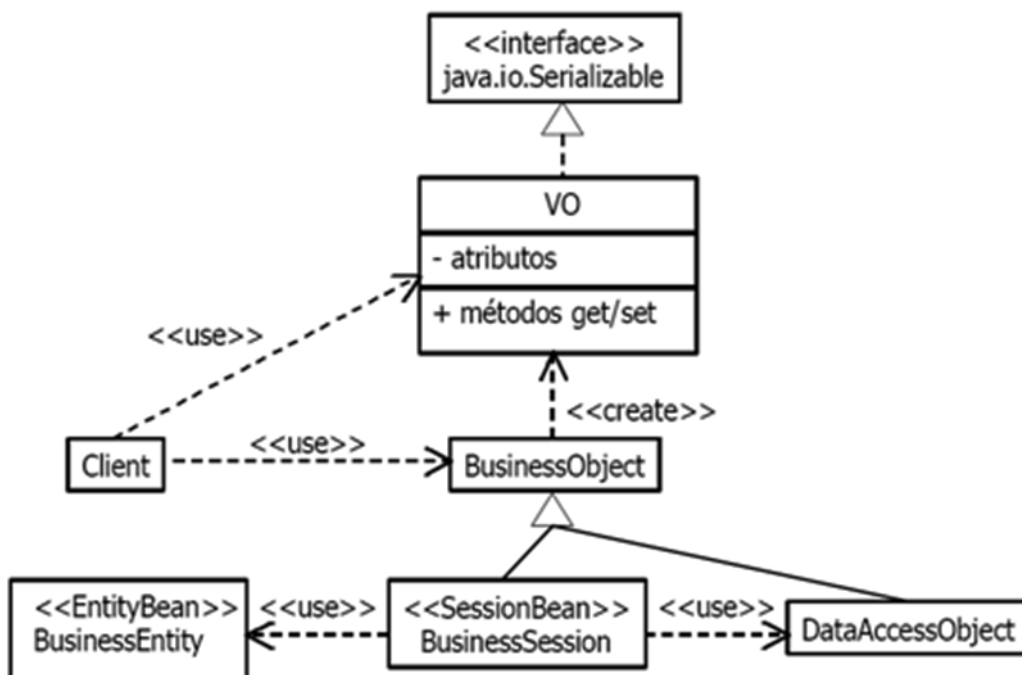


Figura II-23: Estructura del patrón de diseño Value Object.

2.5.5.4 Factory Method

El patrón de diseño *Factory Method* define una interfaz para crear objetos, pero deja que sean las subclasses quienes decidan qué clases instanciar. Así, permite que una clase delegue en sus subclasses la creación de objetos, de forma que la clase derivada toma la decisión sobre qué clase instanciar y cómo instanciarla. El patrón *Factory Method* permite escribir aplicaciones más flexibles respecto de los tipos a utilizar, delegando la creación de las instancias en el sistema a subclasses que pueden ser extendidas a medida que evoluciona el sistema. Permite, además, encapsular el conocimiento referente a la creación de objetos, lo que hace que el diseño sea más adaptable con un aumento mínimo en la complejidad del diseño.

La estructura del patrón *Factory Method*, está formada por los siguientes elementos:

- Producto: define la interfaz de los objetos que crea el método de fabricación.
- ProductoConcreto: implementa la interfaz Producto.
- Creador: declara el método de fabricación, el cual devuelve un objeto del tipo Producto. También puede definir una implementación predeterminada del método de fabricación que devuelve un objeto ProductoConcreto. Puede llamar al método de fabricación para crear un objeto Producto.
- CreadorConcreto: redefine el método de fabricación para devolver una instancia de ProductoConcreto.

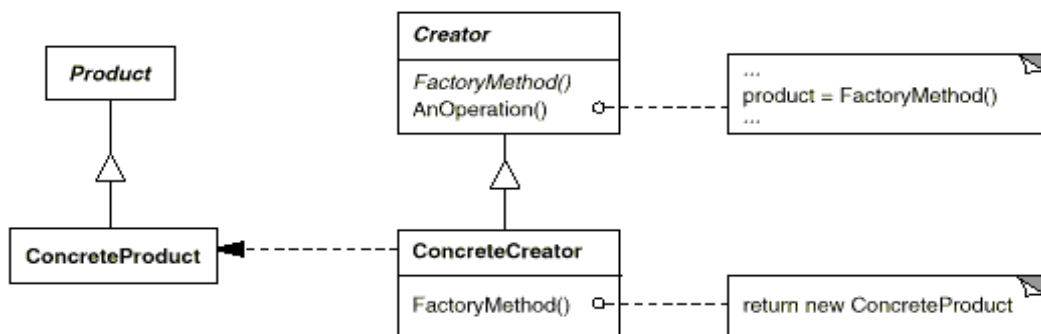


Figura II-24: Estructura del patrón de diseño Factory Method.

2.5.5.5 Facade

Este patrón de diseño proporciona una interfaz unificada o *fachada* (*facade*) para un conjunto de interfaces en un subsistema, es decir, define una interfaz de nivel más alto haciendo al subsistema más fácilmente utilizable y reusable. El patrón Facade oculta al cliente el conjunto de clases de servicio, separando así la funcionalidad, de las clases que la utilizan o clientes. De este modo, el uso de este patrón reduce al mínimo la comunicación y dependencias entre subsistemas.

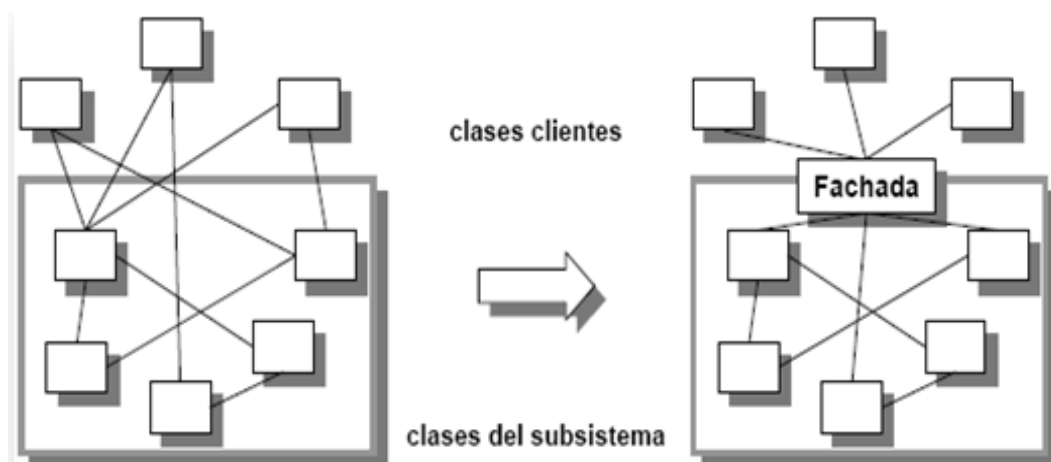


Figura II-25: Estructura del patrón de diseño Facade.

La estructura del patrón Facade, está formada por los siguientes elementos:

- Facade (fachada): conoce que clases del subsistema son responsables de una petición y delega las peticiones de los clientes en los objetos del subsistema.
- Clases del subsistema: implementan la funcionalidad del subsistema. Atienden las peticiones recibidas por el objeto Facade. No tienen conocimiento del Facade (no guardan referencia de éste).

Mediante esta estructura los clientes se comunican con el subsistema a través de la fachada, que reenvía las peticiones a los objetos del subsistema apropiados pudiendo realizar, además, funciones de traducción.

Capítulo III: Marco Experimental

Tal y como se planteaba en la introducción de este trabajo, el objeto de la presente la propuesta es añadir un plano semántico a la representación de Esquemas de forma que posibilite su recuperación conceptual basada en el significado y contenido de sus elementos.

Con este objeto, en el apartado *1.4 Objetivo*, se incluyeron los objetivos del presente trabajo y que se resumen a continuación:

- Mejorar la interoperatividad de esquemas de metadatos heterogéneos, permitiendo correspondencias más precisas, desambiguando la representación y la definición de los términos, así como las relaciones entre conceptos.
- Posibilitar la reutilización de esquemas dentro de un marco extensible que permita incluir definiciones desambiguadas, comprensibles por el usuario y accesibles vía Web.
- Posibilitar una recuperación con un árbol de exploración más amplio, lo que se traduce en un conjunto de resultados más completos, derivados de las relaciones entre conceptos presentes en el modelo conceptual.
- Eliminar la ambigüedad semántica tanto estructural como conceptual mediante una representación más completa de esquemas.
- Recopilar y almacenar un Corpus de referencias a los esquemas más populares, facilitando así su acceso y reutilización.
- Permitir la recuperación conceptual de documentos semánticos, incluyendo búsquedas por el contenido conceptual, así como búsquedas por esquemas y vocabularios de metadatos con un concepto determinado.
- Diseñar un sistema informático que proporcione soporte a la presente propuesta, maximizando su usabilidad.
- Incluir en el diseño del sistema informático de soporte, incorporar metadatos en la representación ontológica de los esquemas, que serán utilizados tanto en la búsqueda como por el algoritmo de posicionamiento.
- Incluir en el diseño del sistema informático la funcionalidad que permita al usuario incorporar sus propios esquemas, que en última instancia serán validados por un sistema de votación de expertos.

- Incluir en el diseño del sistema informático la funcionalidad que permita la recuperación semántica de documentos semánticos y Esquemas asociados, mediante una interfaz Web de acceso libre.

Junto con los objetivos planteados, es necesario establecer unos criterios de calidad que la solución proporcionada deberá cumplir, destinados a maximizar su extensibilidad, compatibilidad y reutilización. Con estas premisas, el conjunto de requisitos de la solución se puede resumir en:

- La solución debe de poderse aplicar a cualquier conjunto de esquemas.
- Debe garantizarse la compatibilidad con sistemas que hagan uso del esquema original, posibilitando la aplicación de principios de simplificación en sistemas no compatibles con el esquema semántico.
- Debido a su popularidad se seguirán las recomendaciones de nombrado de términos y espacios de nombres del DCMI y el W3C.
- La asignación semántica debe permitir la reutilización de elementos en el futuro mediante definiciones consensuadas, no ambiguas, comprensibles y accesibles.
- Se generará la documentación de los esquemas cualificados de forma que se facilite su interpretación, tanto por usuarios como por computadoras.
- Se debe proporcionar una solución que soporte los criterios de modularidad, extensibilidad, refinamiento y plurilingüismo.
- Los esquemas cualificados semánticamente constarán de dos documentos: uno formal dirigido a los usuarios y otro en RDF para el procesamiento automático.

Destacar que los anteriores requisitos no están resueltos en otras propuestas, entre las que destacan: los Perfiles de Aplicación, que sólo permiten un refinamiento local de las definiciones; los registros de metadatos que incluyen definiciones locales y sólo un conjunto de recomendaciones sobre cómo expresarlas; y por último, los *crosswalks* o correspondencias uno a uno entre elementos de dos esquemas, que presentan problemas de consenso en cuanto a la definición semántica de los elementos y a la escalabilidad de la solución. Además, ninguna de estas tres opciones está pensada para la reutilización futura de los metadatos.

A diferencia de las anteriores, la presente propuesta sí tiene como objetivo la reutilización de esquemas dentro de un marco extensible que permita incluir definiciones desambiguadas, comprensibles por el usuario y accesibles vía Web.

Una vez planteados los objetivos y requisitos de la solución, se ha desarrollado la metodología, ver apartado 1.5 *Metodología*, para la consecución de los mismos. Destacar que la metodología presentada ha sido validada, en el ámbito de la cualificación de esquemas, mediante diversas publicaciones, nacionales e internacionales como: la *Fifth International Semantic Web Conference* (Palacios, V. et al., 2006a), la *Conferência Ibérica de Sistemas e Tecnologias de Informação* (Palacios,

V. et al., 2006b) o la *International Conference on Dublin Core and Metadata Applications* (Palacios, V. et al., 2006c).

Los siguientes apartados de este capítulo se estructuran siguiendo los pasos establecidos en la metodología, resumidos en la *Figura III-1*, incluyéndose aspectos relevantes del proceso de resolución en relación a los requisitos establecidos, los problemas encontrados y las soluciones propuestas.

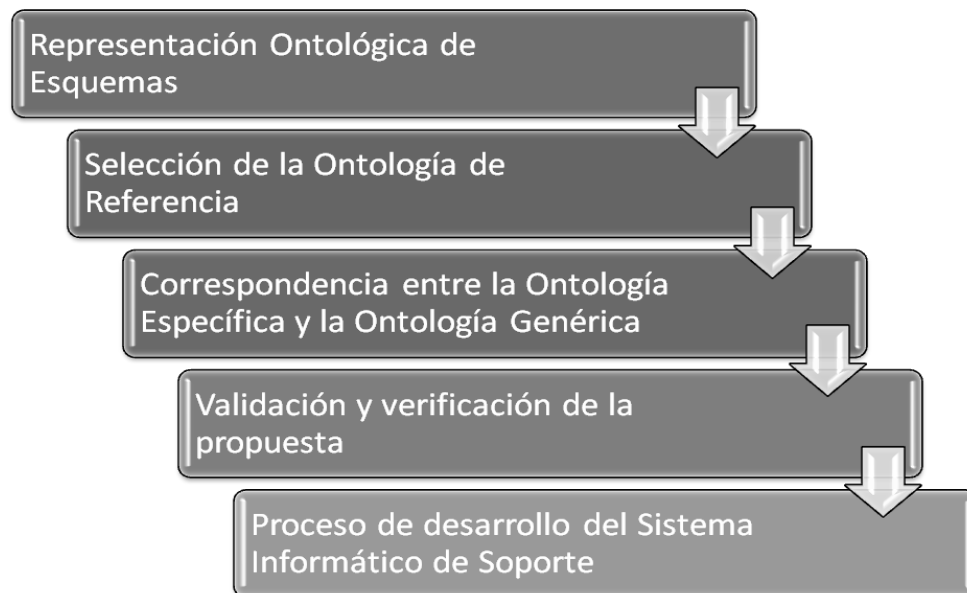


Figura III-1: Principales pasos de la metodología propuesta

3.1 Representación ontológica de esquemas

Son varios los objetivos que se abordan en este apartado y que se resumen a continuación, a modo de introducción.

En primer lugar es necesario establecer el modo de incluir la semántica de los esquemas. A continuación deberá seleccionarse el lenguaje de representación con la capacidad expresiva suficiente para representar dicha semántica y, finalmente, deberá plantearse una solución siguiendo los criterios de calidad propuestos: extensibilidad, compatibilidad y reutilización.

En segundo lugar, es necesario seleccionar un primer conjunto de esquemas sobre el cual poner en práctica la representación propuesta. La selección de esquemas deberá tener en cuenta criterios de popularidad, estabilidad, ámbito y estandarización.

En tercer lugar, deberá realizarse la representación semántica de los esquemas seleccionados, siguiendo el modelo de formalización propuesto y atendiendo a los criterios de calidad antes mencionados.

Finalmente, se procederá a la publicación de las representaciones semánticas de los esquemas para su uso y reutilización, tanto por los usuarios como por la plataforma informática de soporte.

La siguiente figura resume las principales tareas de esta fase:

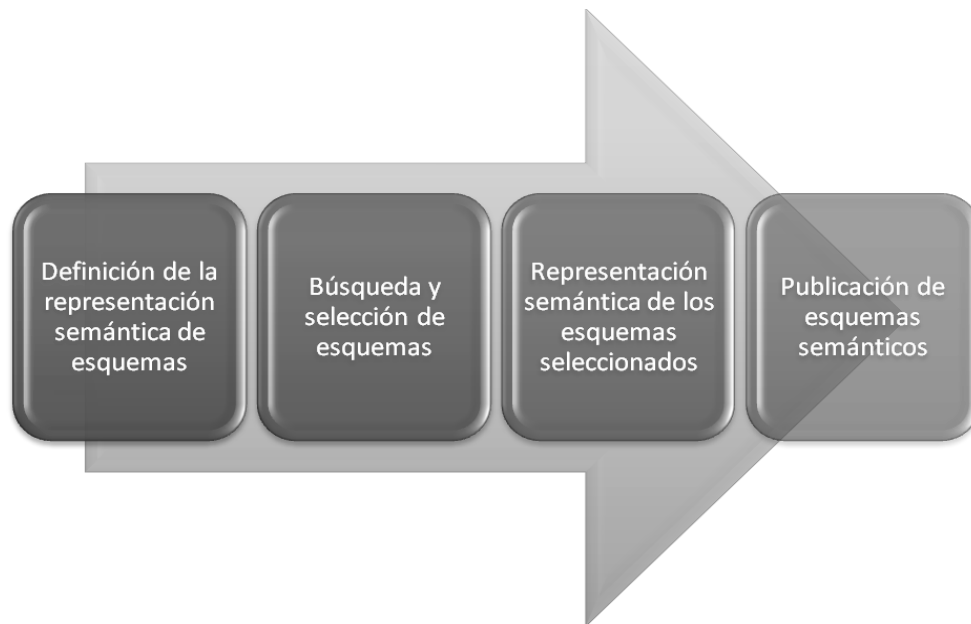


Figura III-2: Tareas de la fase de representación ontológica de esquemas

3.1.1 Definición de la representación semántica de esquemas

La representación semántica de esquemas aborda el problema que supone la ambigüedad sintáctica y semántica que poseen los vocabularios en su definición clásica. Con el fin de paliar dicha ambigüedad, es preciso establecer una representación de los esquemas que permita establecer la semántica de sus elementos, independientemente de su estructura, compatible con el esquema original y aplicable a cualquier esquema. Esta representación facilitará la interoperatividad entre esquemas así como su reutilización, basándose en la semántica común de sus elementos.

La representación propuesta, tal y como se describe a lo largo del presente apartado, se basa en el uso de ontologías como formalismo, dada la capacidad que representación estructural y semántica que poseen. Así, se proponen dos representaciones complementarias para cada esquema:

- La primera de ellas, la versión semánticamente cualificada del esquema, tiene como objetivo sustituir en su uso al esquema original. De este modo, esta representación maximiza la compatibilidad con el esquema de origen, al tiempo que permite incorporar la semántica de los elementos incluidos en el mismo. Tal y como se verá a continuación, estos objetivos se traducen en el uso del lenguaje de definición del esquema original, el mantenimiento del tipo de los elementos y en la cualificación de cada elemento a través de la incorporación de cualificadores semánticos, definidos exprofeso en el esquema de cualificadores semánticos, que permiten relacionar cada elemento con su semántica precisa.
- La segunda representación del esquema, denominada representación como *ontología específica*, es una ontología de dominio dirigida a capturar la

semántica concreta de cada elemento, tal y como se define en el esquema original, al tiempo que aborda aspectos como sinonimia y plurilingüismo. Centrada en la semántica de los elementos, incluye la definición de cada elemento como clase o propiedad, según corresponda, en base a una ontología desarrollada con este fin y denominada SEMSE (Semantic Metadata Search, 2010a).

En el resto del apartado se abordan las representaciones propuestas para cada esquema original, primeramente los *esquemas cualificados semánticamente* y, seguidamente, las *ontologías específicas*.

3.1.1.1 Esquemas cualificados semánticamente

Como primer paso para la representación semántica de esquemas, es necesario establecer el modo en el que puede ser incluida, partiendo de cualquier esquema de metadatos.

Con este objetivo, se inició un proceso de evaluación de esquemas basado en la capacidad de ampliación y grado de estandarización. Dentro de este proceso destacó el DCMI al proponer múltiples recomendaciones destinadas, no sólo al establecimiento de un esquema de metadatos, sino también a la ampliación de dicho esquema y, lo que es más importante, la estandarización en la definición de esquemas de metadatos.

En relación a la estandarización en la definición de esquemas de metadatos, el DCMI propone un modelo de referencia denominado Modelo Abstracto (*Abstract Model*) (Dublin Core Metadata Initiative, 2004). El objetivo de este documento es proporcionar un modelo de referencia independiente de la sintaxis de codificación y contra el cual puedan compararse distintas codificaciones.

Aunque desarrollado para los metadatos del DCMI, el modelo fue analizado desde un punto de vista genérico con el fin de encontrar aspectos comunes, relacionados con la definición y extensibilidad, aplicables a cualquier esquema. El resultado del análisis fue publicado en la Conferencia Internacional de Dublin Core (Palacios, V. et al., 2006e) a través de un artículo en el que se incluían recomendaciones para facilitar la interpretación y uso del Modelo Abstracto. Algunas de las recomendaciones propuestas han sido incluidas en la nueva versión del Modelo Abstracto (Powel, A. et al., 2007).

En relación con el modo de extender los términos para incluir la semántica, fue de especial interés el planteamiento que se propuso en (Palacios, V. et al., 2006e) para la extensión de las clases y propiedades, mediante la definición de subclases y subpropiedades. Haciendo uso de la relación de generalización, los términos, clases y propiedades, pueden ser especializados generando nuevos términos. El uso de la relación de generación es especialmente importante debido a las propiedades que implica:

- Un término puede tener cualquier número de subtérminos, lo que facilita la extensibilidad.
- El término *genérico* en la jerarquía, es independiente de los términos *específicos*

o *subtipos*. De nuevo se facilita la extensibilidad al no ser necesaria la modificación del término original.

- Los términos *específicos* son, desde un punto de vista genérico, del tipo del término padre. De este modo, un subtérmino conserva las líneas generales definidas por el término del cual proviene y se dice que es un *subtipo* del padre. Se trata de un aspecto de gran relevancia en lo concerniente a la compatibilidad, al mantener las características del término padre, se proporciona la posibilidad de utilizar dichas características en caso de no poderse procesar el término hijo. También tiene relación con la reutilización al heredarse la definición del padre por los hijos.
- En relación con el punto anterior, si bien un término hijo comparte por defecto las propiedades del término padre del cual proviene, el hijo tiene la capacidad de redefinir esas propiedades. Este aspecto tiene relevancia en relación con la definición de la semántica de los elementos, permitiendo al término hijo redefinir el significado heredado de su padre.

Una vez encontrada la forma de ampliar los elementos, mediante la relación de generalización, el siguiente paso consistió en analizar cómo se definían los subtérminos en el esquema del DCMI. Esto llevó al análisis de los cualificadores incluidos en Conjunto de Elementos de Metadatos de DC (*DCMES*). Los cualificadores pueden clasificarse en cualificadores de codificación de sintaxis y cualificadores de refinamiento. Los primeros permiten representar vocabularios controlados y formatos, lo cual no se prestaba a la representación semántica. Los cualificadores de refinamiento, por el contrario, se definen haciendo uso de la relación de generalización, son subtérminos de los originales, y permiten la redefinición de la semántica heredada del término padre. Esta redefinición, no obstante, tiene limitaciones, un hijo puede definir su semántica siempre que se encuentre dentro de la definición del término padre, en otras palabras, el término hijo sólo puede especializar o restringir la definición del término del cual proviene.

Viendo que era posible que los cualificadores de refinamiento fueran adecuados para representar la semántica, se inició una evaluación de los mismos. Como resultado se comprobó que ninguno de ellos se adecuaba a la representación de la semántica del elemento original. Del mismo modo se comprobó que tampoco los elementos del esquema básico se prestaban al objetivo planteado. Concretamente, de los elementos y cualificadores evaluados, únicamente el término *relation* se aproximaba a la representación semántica buscada. Partiendo de este término, las opciones de representación son:

- Refinamiento del término *relation*: Podría generarse un elemento de refinamiento del término para albergar la semántica. De este modo, la solución quedaría completamente integrada con términos de DC. Sin embargo, sería preciso modificar el conjunto de cualificadores de refinamiento definidos por el DCMI.
- Uso del término *relation*: El uso del término es posible y no precisa ninguna modificación en el conjunto de cualificadores del DCMI. Como principal

inconveniente está la escasa semántica del término, únicamente indica una relación con el término original. De modo similar al anterior, en este caso la solución también quedaría integrada con el esquema DC.

- Definición de un nuevo conjunto de cualificadores: La definición de un nuevo esquema de cualificación no implicaría la modificación de los esquemas del DCMI. El nuevo esquema cualificaría los elementos del esquema DC de forma distinta a los existentes, ya que incluiría un nuevo conjunto de términos que podríamos denominar *cualificadores semánticos*. De este modo, el nuevo esquema semántico incluiría un término derivado de *relation*, que expresaría la relación semántica entre el término original y su significado.

De las opciones planteadas ha sido seleccionada la tercera, las razones de esta elección son, básicamente: por tratarse de un modo de extensión propuesto por el DCMI; por compatibilidad con el esquema original y con los cualificadores del DCMI; por considerarse más *natural* la creación de un nuevo tipo de esquema que albergue cualificadores que aportan un matiz distinto a los términos originales y por tratarse de una solución más flexible, dado que en el esquema puede albergar futuros cualificadores derivados, por ejemplo, de otros esquemas de metadatos.

De este modo, la solución propuesta define un esquema de metadatos denominado *Esquema de Cualificadores Semánticos (Semantic Qualifiers Schema)*. El objetivo de este esquema es albergar un conjunto de *Cualificadores Semánticos (Semantic Qualifiers)* que permitan definir la semántica de los elementos de cualquier esquema, mediante la cualificación semántica de sus términos. El esquema será definido mediante los siguientes metadatos:

Esquema de Cualificadores Semánticos

Nombre: Esquema de Cualificadores Semánticos

Título: Cualificadores Semánticos para la desambiguación semántica de Esquemas a través de una Ontología de Referencia

Descripción: Los espacios de nombres de los términos incluidos en este esquema, proporcionan URIs del Esquema de Cualificadores Semánticos. Los Cualificadores Semánticos forman un vocabulario más rico, que simplifica la desambiguación semántica de términos y el descubrimiento de recursos

Localización: <http://purl.org/semse/sq/>

Publicado por: Vicente Palacios

Estado:	propuesto
Fecha de publicación:	25-01-2010

Tabla III-1: Metadatos para la definición de Esquema de Cualificadores Semánticos

Dentro de la definición, cabe destacar la necesidad de que el esquema esté accesible vía Web. Para ello, siguiendo las indicaciones del DCMI sobre Políticas de Espacios de Nombres (Dublin Core Metadata Initiative, 2001) se ha hecho uso de una URL persistente, concretamente, la proporcionada por el servicio *Persistent Uniform Resource Locator (PURL)* (Online Computer Library Center, 2010b). A través de este servicio se garantiza la accesibilidad del esquema a través de una URI que permanece constante, independientemente de cualquier posible cambio en la dirección real del esquema.

Una vez definido el esquema, es necesario incluir en él los cualificadores semánticos. Dado que son cualificadores, provendrán de un término más genérico a través de la relación de generalización. De este modo, redefinirán la semántica del elemento original, haciéndola más específica y aportando mayor semántica sobre la relación existente entre el término al que se aplica y el valor que toma el cualificador.

Después de evaluar distintos esquemas, se decide tomar el término *relation* del esquema Dublin Core como término padre para el nuevo cualificador semántico. Esta decisión se basa en su popularidad, grado de estandarización, estabilidad y legibilidad. La definición del término *relation*, según el esquema del DCMI es:

Term Name: relation	
URI:	http://purl.org/dc/elements/1.1/relation
Label:	Relation
Definition:	A reference to a related resource.
Comment:	Recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system.
Type of Term:	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
Date Issued:	1999-07-02
Modified	2008-01-14

Tabla III-2: Definición original del término 'relation' del esquema Dublin Core

Tal y como se observa en la definición del término, su significado es una referencia a un recurso relacionado y, como ámbito de valores posibles, se recomienda la identificación del recurso siguiendo un sistema de identificación formal.

La solución propuesta parte de esta definición para definir un nuevo cualificador basado en el término *relation* y denominado *hasSemantics* (*tieneSemantica*). Haciendo uso de este cualificador, es posible relacionar términos de esquemas con sus significados, relacionándolos mediante un sistema formal de identificación. Los significados de los términos y el recurso del cual serán extraídos, será un aspecto a abordar posteriormente, de momento, bastará con definir el rango de valores del cualificador dentro de la especificación de URIs [12], previendo la identificación de los significados mediante sus URIs. De este modo, el cualificador semántico quedará definido como:

Nombre del Término: hasSemantics	
URI:	http://purl.org/semse/sq/hasSemantics
Etiqueta:	Tiene Semántica
Definición:	Relación Semántica con una Ontología de Referencia.
Comentario:	Elemento de Refinamiento de DC.Relation para enlazar elementos de un esquema con su semántica mediante relaciones con Conceptos de una Ontología de Referencia
Refina	http://purl.org/dc/elements/1.1/relation
Tipo de Término:	http://dublincore.org/usage/documents/principles/#element-refinement
Estado:	propuesto
Fecha de Publicación:	25-01-2010

Tabla III-3: Metadatos para la definición del cualificador semántico ‘*tieneSemantica*’

A partir de la tabla anterior observamos como:

- El nombre del cualificador es ‘hasSemantics’ (*tieneSemantica*), asignado según las recomendaciones sobre nombrado del DCMI (Weibel, S.L.; Baker, T., 2004).
- El cualificador debe publicarse en Web para hacerlo accesible. De modo similar a la publicación del esquema, debe garantizarse su accesibilidad vía Web, para lo cual se hace uso de una URL persistente (Online Computer Library Center, 2010b).
- Se define y comenta el significado y uso del cualificador.
- Se indica el término original del cual proviene. Esto es, el término que refina

dentro del esquema de metadatos original, en este caso *relation* de Dublin Core. De este modo se amplía el conjunto de cualificadores de refinamiento con el elemento semántico recién definido.

Una vez definidos el esquema y los cualificadores semánticos, el siguiente paso es establecer cómo se van a incluir en los términos para poder aportarles la semántica que precisan. En este punto, de nuevo, volvemos a encontrarnos con distintas posibilidades:

- Modificar el esquema original del DCMI para incluir el cualificador en la descripción de cada término. Se trata de una modificación que tendría que contar con la aprobación del DCMI y que atenta contra la simplicidad y compatibilidad del esquema original.
- Redefinir los elementos del esquema original en un nuevo esquema, para incluir el cualificador en la definición de cada término. En este caso, se proporcionaría una solución *local* que modificaría la definición del esquema original, ampliamente aceptada y utilizada. Atentaría contra el principio de reutilización, al volver a definirse los elementos, y de nuevo, contra la simplicidad y compatibilidad del esquema original.

Crear un perfil de aplicación no resulta una solución viable, puesto que sólo permite la inclusión y, como máximo, el refinamiento de la semántica del elemento, lo cual es insuficiente para definir las relaciones semánticas de cada elemento.

- Crear un nuevo conjunto de elementos en nuevo esquema, donde la definición de cada término incluya el cualificador semántico. De este modo, no es necesaria la modificación del conjunto original de elementos, de hecho, definiendo los elementos semánticos como subpropiedades de los elementos originales, se facilita la compatibilidad y se reutilizan los términos del esquema original, manteniéndose su simplicidad.

De las opciones planteadas ha sido seleccionada la última, en base a las mejoras que aporta sobre los criterios de compatibilidad, simplicidad y reutilización. De este modo, la cualificación de un esquema de metadatos pasaría por:

- La creación de un nuevo esquema para albergar los elementos semánticamente cualificados. El nombre de este esquema provendrá del nombre del esquema original, precedido por el acrónimo 'sq' *Semantic Qualified (Cualificado Semánticamente)*.
- La generación de un elemento cualificado semánticamente para cada término incluido en el esquema original. Los elementos serán subtérminos de los términos originales e incluirán en su descripción el atributo *hasSemantics* que, posteriormente, permitirá albergar una referencia al significado del elemento.

A continuación se muestra un ejemplo de elemento semánticamente cualificado que, según las recomendaciones anteriores, pertenecería al esquema *Dublin Core Cualificado Semánticamente (Semantic Qualified Dublin Core Metadata Elements)*.

Nombre del Término: <i>rights</i>	
URI:	http://purl.org/semse/sqdcelements/rights
Etiqueta:	Rights
Definición:	Elemento equivalente al término <i>dc:rights</i> , pero desambiguado semánticamente mediante una ontología de referencia.
Subpropiedad de	http://purl.org/dc/elements/1.1/rights
Tipo de Término:	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
Semántica	<URI al concepto que alberga la semántica del elemento >
Fecha de Publicación:	22-02-2010

Tabla III-4: Ejemplo de cualificación semántica del elemento 'rights' del esquema Dublin Core

Como se puede observar a partir del ejemplo:

- En el nombrado del término se han seguido las indicaciones de nombrado del DCMI (Weibel, S.L.; Baker, T., 2004). Se ha mantenido el nombre del elemento que cualifica con el fin de facilitar la compatibilidad con el esquema original así como su uso en sustitución del término original no cualificado.
- El término debe publicarse en Web para hacerlo accesible. De nuevo, se siguen las recomendaciones sobre espacios de nombres del DCMI (Dublin Core Metadata Initiative, 2001) y se hace uso de una URL persistente (Online Computer Library Center, 2010b).
- Se define el significado y uso del elemento.
- Se indica el término original del cual proviene. Esto es, el término que refina dentro del esquema de metadatos original, en este caso *rights* de Dublin Core, a través de la relación de generalización.
- Tal y como establece el elemento original, es necesario establecer el tipo del elemento como propiedad, según se define en el esquema RDF (*rdf:Property*). De este modo se mantiene el tipo del elemento original, facilitándose la compatibilidad con el esquema original y la sustitución de éste por la nueva versión cualificada semánticamente.
- Se reserva un espacio en la definición para la futura definición del significado

del concepto. En el ejemplo se denota como una posible URI al concepto de una ontología compartida vía Web.

- Finalmente se incluye información sobre la fecha de publicación del elemento.

Hay que destacar que, para maximizar la compatibilidad de las aplicaciones con la nueva versión del esquema, se ha decidido mantener los tipos originales de los elementos que se cualifican. De este modo, igual que en el ejemplo anterior se mantenía el tipo del elemento semántico como *rdf:Property* (tipo del elemento original), se han encontrado otros casos en los que ha sido necesario definir los nuevos elementos como *rdfs:Class*, *owl:Class*, etc., como se mostrará en posteriores ejemplos.

Con lo expuesto hasta el momento, se han definido los elementos necesarios para ampliar esquemas de metadatos, de forma que sea posible incluir la semántica de los términos originales. Además, se han establecido las bases del procedimiento para la cualificación de esquemas.

Llegados a este punto, es necesario establecer un lenguaje de representación capaz de representar la solución propuesta. El lenguaje debe permitir la interpretación del esquema por computadoras y la publicación del esquema en la Web. Además de lo anterior, deben tenerse en cuenta criterios como popularidad en su uso, grado de estandarización y soporte por parte de aplicaciones, por ejemplo, navegadores.

Tal y como se mostraba a través del ejemplo anterior, con el objetivo de maximizar la compatibilidad de los nuevos esquemas con sus originales no cualificados, se ha decidido mantener el lenguaje utilizado en la definición del esquema original, en los cualificados hasta el momento, RDF y OWL. Flexibilizando de este modo la solución, al no imponer un lenguaje de definición para los nuevos esquemas, se facilita el uso del esquema cualificado en sustitución del esquema original, al tiempo que se mantiene la compatibilidad con este último para aquellas aplicaciones que precisen hacer uso del esquema de origen.

Para la definición del esquema de cualificadores semánticos y siguiendo las recomendaciones del W3C para la Web Semántica, los mejores candidatos para la representación son RDF y OWL. Ambos poseen la expresividad necesaria para representar la solución propuesta, son interpretables por computadoras y son estándares aprobados y en uso. La decisión entre uno y otro se ha tomado en base a criterios de popularidad y soporte por aplicaciones. Si bien OWL posee mayor capacidad semántica, actualmente no goza de la misma popularidad de uso que RDF (véase el apartado 2.2.1 *Esquemas de metadatos*). Además, el soporte de OWL en aplicaciones no está aún tan extendido como el de RDF. Por estos motivos, se ha seleccionado como lenguaje de representación RDF, aunque cabe destacar que la futura conversión a OWL de las implementaciones propuestas, no supondría apenas esfuerzo. De hecho, se han realizado pruebas con el programa de gestión de ontologías *Protégé* (Stanford University, 2010b), comprobándose cómo es perfectamente posible cargar y guardar el esquema propuesto en RDF, en las distintas versiones de OWL.

Una vez seleccionado el lenguaje de representación, se realizará una primera codificación de los esquemas y términos propuestos, para mostrar cómo sería la versión

procesable por computadoras.

Como primer ejemplo se presentará la codificación del cualificador semántico *hasSemantics*:

```
<rdf:Property rdf:about="&sq;hasSemantics">
  <rdfs:label xml:lang="en-EN">has semantics</rdfs:label>
  <rdfs:label xml:lang="es-ES">tiene semántica</rdfs:label>
  <rdfs:comment xml:lang="en-EN">Semantic relation with a Reference
  Ontology.</rdfs:comment>
  <rdfs:comment xml:lang="es-ES">Relación Semántica con una Ontología
  de Referencia.</rdfs:comment>
  <dc:description xml:lang="en-EN">Refinement Element from DC.Relation
  to link schema elements with their semantics through relations with
  concepts of the Reference Ontology.</dc:description>
  <dc:description xml:lang="es-ES">Elemento de Refinamiento de
  DC.Relation para enlazar elementos de un esquema con su semántica
  mediante relaciones con Conceptos de una Ontología de
  Referencia.</dc:description>
  <rdfs:isDefinedBy rdf:resource="&sq;"/>
  <dcterms:issued xml:lang="es-ES">25-01-2010</dcterms:issued>
  <dc:type rdf:resource="&dcdocns;element-refinement"/>
  <rdfs:subPropertyOf rdf:resource="&dc;relation" />
</rdf:Property>
```

Ejemplo III-1: Codificación del cualificador semántico 'hasSemantics' en RDF

Tal y como se puede observar, se trata de una codificación de la información definida anteriormente para el cualificador semántico. Aunque la definición de los espacios de nombres se incluirá en el siguiente ejemplo, se puede comprobar cómo se definen y utilizan los términos del esquema semántico, de forma idéntica al de los términos del resto de esquemas.

Respecto a la codificación, cabe destacar el modo en el que se ha expresado el hecho de que el elemento proviene del elemento original *dc:relation*. Concretamente, se ha hecho uso de la relación *rdfs:subPropertyOf*, mediante la cual se expresa que el término es un subtérmino del original. De este modo, se hace uso de la relación de generalización con las ventajas que aporta, comentadas anteriormente.

Una vez visto el ejemplo de codificación del cualificador semántico, el siguiente ejemplo muestra la codificación del esquema de cualificadores semánticos en el que se incluye el anterior. Para dar una visión completa del esquema, se ha incluido la definición del cualificador anterior.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY sq "http://purl.org/semse/sq/">
  <!ENTITY dc "http://purl.org/dc/elements/1.1/">
  <!ENTITY dcterms "http://purl.org/dc/terms/">
  <!ENTITY dcdocns
"http://dublincore.org/usage/documents/principles/#">
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
]>
```

```

<rdf:RDF
xmlns:sq="&sq;"
xmlns:dc="&dc;"
xmlns:dcterms="&dcterms;"
xmlns:dcdocns="&dcdocns;"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="&rdfs;"
>

<rdf:Description rdf:about="&sq;">
  <dc:title xml:lang="en-EN">Semantic Qualifiers for semantic scheme
  disambiguation through a Reference Ontology.</dc:title>
  <dc:title xml:lang="es-ES">Cualificadores Semánticos para la
  desambiguación semántica de esquemas mediante una Ontología de
  Referencia.</dc:title>
  <dc:publisher xml:lang="es-ES">Vicente Palacios
  Madrid.</dc:publisher>
  <dc:description xml:lang="es-ES">This namespace provides URIs for
  the semantic qualifiers schema. The vocabulary's terms are defined
  using RDFS language to support RDF applications. Semantic Qualifiers
  conforms a richer vocabulary that simplifies the terms' semantic
  disambiguation and the resource discovery.</dc:description>
  <dc:description xml:lang="es-ES">Este espacio de nombres proporciona
  URIs para el esquema de Cualificadores Semánticos. Los términos del
  vocabulario son definidos utilizando el lenguaje RDF Schema para
  soportar aplicaciones RDF. Los Cualificadores Semánticos conforman
  un vocabulario más rico, que simplifica la desambiguación semántica
  de términos y el descubrimiento de recursos.</dc:description>
  <dc:language xml:lang="en-EN">English</dc:language>
  <dcterms:issued xml:lang="es-ES">25-01-2010</dcterms:issued>
  <dc:source rdf:resource="&sq;"/>
  <dcterms:requires rdf:resource="&dc;"/>
  <dcterms:requires rdf:resource="&dcterms;"/>
</rdf:Description>

<rdf:Property rdf:about="&sq;hasSemantics">
  <rdfs:label xml:lang="en-EN">has semantics</rdfs:label>
  <rdfs:label xml:lang="es-ES">tiene semántica</rdfs:label>
  <rdfs:comment xml:lang="en-EN">Semantic relation with a Reference
  Ontology.</rdfs:comment>
  <rdfs:comment xml:lang="es-ES">Relación Semántica con una Ontología
  de Referencia.</rdfs:comment>
  <dc:description xml:lang="en-EN">Refinement Element from DC.Relation
  to link schema elements with their semantics through relations with
  concepts of the Reference Ontology.</dc:description>
  <dc:description xml:lang="es-ES">Elemento de Refinamiento de
  DC.Relation para enlazar elementos de un esquema con su semántica
  mediante relaciones con Conceptos de una Ontología de
  Referencia.</dc:description>
  <rdfs:isDefinedBy rdf:resource="&sq;"/>
  <dcterms:issued xml:lang="es-ES">25-01-2010</dcterms:issued>
  <dc:type rdf:resource="&dcdocns;element-refinement"/>
  <rdfs:subPropertyOf rdf:resource="&dc;relation" />
</rdf:Property>
</rdf:RDF>

```

Ejemplo III-2: Codificación completa del esquema de cualificadores semánticos en

RDF

Como se puede observar a partir del ejemplo, la definición del esquema comienza por la declaración de los espacios de nombres de los términos que serán utilizados en su definición. Entre ellos se encuentra la definición del espacio de nombres del propio esquema: <http://purl.org/semse/sq>. Una vez definidos los espacios de nombres, se incluyen los metadatos descriptivos del propio esquema: título del esquema, publicador, descripción, lenguaje, fecha de publicación, URI del esquema y URIs de los esquemas requeridos. A continuación se incluye la definición de los elementos del esquema, en este caso, se ha incluido la definición del único elemento que, de momento, contendría: el cualificador semántico *hasSemantics*.

Una vez definido el esquema, puede ser utilizado en el proceso de cualificación semántica de esquemas de metadatos, tal y como se describía anteriormente. Aunque aún es necesario resolver aspectos relacionados con la representación de los conceptos que albergarán la semántica de los elementos, a continuación se muestra la codificación del ejemplo de cualificación semántica anterior, correspondiente al término *rights* de Dublin Core:

```
<rdf:Property rdf:about="&sqdcelements;rights">
<rdfs:label xml:lang="en-EN">rights</rdfs:label>
<rdfs:comment xml:lang="en-EN">Equivalent element to rights term
semantically disambiguated through a reference
ontology.</rdfs:comment>
<rdfs:comment xml:lang="es-ES">Elemento equivalente al término rights,
pero desambiguado semánticamente mediante una ontología de
referencia.</rdfs:comment>
<rdfs:isDefinedBy rdf:resource="&sqdcelements;"/>
<dcterms:issued xml:lang="en-EN">22-02-2010</dcterms:issued>
<rdf:type rdf:resource="&rdf;Property"/>
<rdfs:subPropertyOf rdf:resource="&dc;rights"/>
<sq:hasSemantics>
<URI a la semántica del término>
</sq:hasSemantics>
</rdf:Property>
```

Ejemplo III-3: Codificación en RDF del término 'rights' de Dublin Core

Sobre el ejemplo cabe destacar: por simplicidad no se han incluido los espacios de nombres ya que serán definidos concretamente apartados posteriores, no obstante, comentar que se creará un nuevo espacio de nombres para este esquema, representado en el ejemplo como *&sqdcelements* (forma abreviada de *Semantic Qualified Dublin Core Elements*); el término generado incluye su propia definición, puesto que se trata de un nuevo elemento; el tipo del elemento mantiene el tipo del elemento original y es especificado como *&rdf;Property*; la relación de especialización con el elemento original es expresada mediante la propiedad *rdfs:subPropertyOf*, es decir, mediante la relación de generalización padre-hijo y, finalmente, la definición incluye el cualificador semántico *hasSemantics*, del espacio de nombres de los cualificadores semánticos *sq* (*Semantic Qualifiers*), que tomará como valor la URI al concepto que expresa el significado del término (en este caso se ha dejado indicado como *<URI a la semántica del término>*).

Una vez expuestas las líneas generales para la incorporación de semántica a esquemas de metadatos, es preciso analizar la solución conforme a los criterios de calidad definidos anteriormente:

- *Extensibilidad*: La solución propuesta se ha diseñado siguiendo este criterio, en cuanto a que es posible añadir nuevos cualificadores semánticos distintos al definido inicialmente. Para ello se ha definido un esquema independiente que los contiene, de forma que las modificaciones en él no afecten a los esquemas que lo utilicen. De este modo, añadir un nuevo cualificador no afectaría en modo alguno a los esquemas cualificados, siempre que se mantenga la compatibilidad hacia atrás de los cualificadores ya utilizados.
- *Compatibilidad*: La solución propuesta se adecúa a este criterio, en cuanto a que se ha diseñado para garantizar la compatibilidad con la versión original del esquema. De este modo, las aplicaciones no compatibles con esquemas cualificados semánticamente podrán hacer uso del principio de simplificación (*Dumb-Down*) y utilizar el esquema de metadatos original. El modo de lograr esta compatibilidad ha sido utilizar la relación de generalización en la definición de los cualificadores semánticos y en el proceso de cualificación semántica de esquemas. También se ha tenido en cuenta este criterio a la hora de establecer el modo de cualificación de esquemas, definido de forma compatible para cualquier esquema de metadatos. La selección del lenguaje de codificación también ha tenido en cuenta esta característica proponiéndose RDF, en el caso del esquema de cualificadores semánticos, por estar mejor soportado actualmente por aplicaciones de la Web. Para los esquemas cualificados se ha decidido mantener el lenguaje de definición del esquema original. Por último, la definición del cualificador semántico, en lo relativo a la asignación semántica del término, se ha realizado de forma que pueda ser compatible con cualquier sistema de identificación formal. Aunque será tratado posteriormente, de momento se ha definido con un rango de valores de un sistema de identificación formal, ejemplificado con URIs, de forma que los conceptos de la ontología de referencia podrán ser identificados en función del sistema de este sistema de identificación. Finalmente destacar que la versión cualificada del esquema proporciona una representación homogénea de todos los esquemas, lo cual redundará en la simplificación de su gestión, consulta y uso.
- *Reutilización*: Se ha atendido a este criterio en la definición del esquema de cualificación y del proceso de cualificación de esquemas. En ellos se importan distintos esquemas existentes para hacer uso de sus elementos en la definición de los términos. Además, la definición independiente del esquema de cualificación permite la reutilización del mismo en la cualificación futura de nuevos esquemas. La asignación de semántica a los elementos, dentro del proceso de cualificación de esquemas, también se realiza atendiendo a este criterio. Por ello, se propone el uso de un recurso compartido en Web que pueda ser reutilizado por múltiples esquemas.

Finalmente, debe destacarse la legibilidad de la solución. Aunque será tratado en apartados posteriores, debe tenerse en cuenta que la solución debe ser interpretable por personas y por computadoras. La forma de cubrir ambos colectivos ya se ha delineado,

al mostrarse las definiciones de los esquemas y elementos en dos formatos: mediante tablas y en código RDF. Como se verá posteriormente, los esquemas propuestos tendrán dos versiones: la primera, en HTML, estará dirigida a personas y facilitará su interpretación mediante formatos amigables y la definición de elementos, recursos relacionados con el esquema, etc.; la segunda, en RDF, estará dirigida a computadoras y codificada de forma similar a los ejemplos anteriormente expuestos. En ambos casos, estarán publicadas en Web y accesibles mediante URIs persistentes, facilitándose así su localización y uso¹⁴.

3.1.1.2 Ontologías Específicas

Una vez desarrollada la representación cualificada de esquema que parte del esquema original, proporcionando compatibilidad, homogeneidad en su definición y permite la relación de cada elemento con su semántica, el siguiente paso es abordar la definición de la semántica de los elementos.

El porqué de definir una nueva representación para albergar la semántica de los elementos obedece a varios motivos. Primeramente, separar la semántica de los elementos haciendo uso de un formalismo específico, en este caso una ontología, permite especializar la representación de los significados, favoreciéndose la modificabilidad y mantenibilidad de éstos (es posible, por ejemplo, que la persona que genera el esquema cualificado y la encargada de asignar y mantener los significados de los elementos no sea la misma, especializándose así en tareas distintas). Otra de las ventajas que aporta una representación especializada de la semántica es la posibilidad de utilizar lenguajes que aporten mayor capacidad de representación semántica, el objetivo fundamental de esta versión del esquema. Al tratarse de una representación distinta, no es preciso mantener el lenguaje de definición del esquema cualificado que, en la mayoría de los casos, no es el más indicado para representar o albergar semántica. Además de la ventaja en la especialización de los significados que aporta el uso de una ontología, este formalismo permite abordar otros aspectos como la sinonimia de términos o la posibilidad de definirlos en múltiples idiomas, todo ello evitando la necesidad de complicar el esquema original o su versión cualificada. Estas ventajas se han primado en la decisión de generar una nueva representación del esquema frente a la desventaja que supone la complejidad de gestionar la nueva formalización.

Evaluadas las ventajas de formalizar la semántica de los elementos mediante una representación ontológica derivada del esquema original, en adelante *ontología específica*, el primer paso consistió en definir la estructura más adecuada a este fin.

Al igual que el esquema de cualificadores semánticos define el conjunto de cualificadores que permiten generar los distintos esquemas cualificados, es preciso definir la ontología a partir de la cual generar las distintas ontologías específicas. Esta ontología padre, denominada *SEMSE*, es una ontología de representación que tiene como objetivo definir la estructura, conceptos y propiedades, destinados a albergar la semántica de los elementos, tal y como se define en los esquemas originales. Asimismo,

¹⁴ Actualmente se encuentra en desarrollo una aplicación que permite la generación de las versiones para personas de los esquemas. Utilizando las librerías de JavaReports, permite generar versiones amigables de los esquemas en formatos como PDF, HTML y DOC, entre otros. Se pretende hacer uso de la funcionalidad JavaReports Server para que estas versiones se generen bajo demanda y están accesibles vía web.

además de la semántica de los elementos, la ontología SEMSE también proporciona soporte para albergar sinónimos y múltiples idiomas en la definición de los conceptos.

La definición de la estructura de la ontología SEMSE se inició con un proceso de evaluación de distintas ontologías primando, además de los requisitos mencionados anteriormente, criterios de extensibilidad y grado de estandarización. En este proceso destacó la estructura de una de las ontologías evaluada como posible ontología de referencia, en concreto UMBEL (UMBEL.org, 2009). Aunque sólo permite la representación de clases, su estructura es clara y está preparada para dar soporte a sinonimia y plurilingüismo.

Tomando como base la estructura de la ontología UMBEL se procedió a definir la estructura de la ontología SEMSE, adaptando y añadiendo los elementos necesarios para dar cobertura a los requisitos planteados. La ontología define los siguientes metadatos.

Ontología SEMSE

Nombre:	SEMSE
Título:	Ontología Específica del proyecto SEMSE: SEMantic Metadata Search
Descripción:	Ontología para la representación ontológica de esquemas de metadatos. Basada en el esquema de la ontología UMBEL (Upper-level Mapping and Binding Exchange Layer), permite la representación de conceptos y propiedades, así como sus significados, soportando sinonimia y plurilingüismo.
Localización:	http://purl.org/semse/
Publicado por:	Vicente Palacios Madrid
Idioma:	Español
Fecha de publicación:	13-10-2009

Tabla III-5: Metadatos para la definición de la ontología SEMSE

Al igual que los artefactos desarrollados anteriormente, la ontología se ha publicado en Web mediante una URL persistente proporcionada por el servicio PURL (Online Computer Library Center, 2010b).

Como lenguaje de definición de la ontología se ha seleccionado OWL Full (Bechhofer, S. et al., 2004), dadas las posibilidades que ofrece para la representación de semántica y por estar diseñado para la publicación en el entorno Web, aspectos alineados con los objetivos de la ontología SEMSE. Complementando a OWL en aspectos concretos, con el fin de aportar mayor semántica y compatibilidad en la definición de conceptos y relaciones, se ha hecho uso de otros lenguajes como SKOS (World Wide Web

Consortium, 2009d) o RDFS (World Wide Web Consortium, 2004a), así como de una ontología específica para la anotación de vocabularios (Brickley, D., 2009).

Una vez definida la ontología, es necesario incluir en ella los conceptos y relaciones destinados a servir de base para la generación de ontologías específicas. Teniendo en cuenta que el objetivo es albergar la semántica de los elementos que pueden encontrarse en esquemas de metadatos, deben incluirse conceptos que permitan albergar definiciones de clases y propiedades. Además, debe modelarse el hecho de que un concepto puede tener nombres alternativos en el contexto definido por el esquema¹⁵. Finalmente, todos los elementos incluidos deben poder definirse utilizando distintos idiomas, con el fin de proporcionar soporte al plurilingüismo.

Comenzando por la representación de las clases de los esquemas, se ha incluido una primera clase denominada *SubjectConcept* o concepto temático. Este concepto expresa la semántica de un elemento de tipo clase, contextualizada por el esquema donde se define. A continuación se muestra la definición incluida en la ontología SEMSE:

```
<owl:Class rdf:about="http://purl.org/semse/SubjectConcept">
  <isDefinedBy rdf:resource="http://purl.org/semse/">
  <subClassOf rdf:resource="http://www.w3.org/2004/02/skos/core#Concept"/>
  <subClassOf rdf:parseType="Resource">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Restriction"/>
    <owl:minCardinality>1</owl:minCardinality>
    <owl:onProperty rdf:resource="http://purl.org/semse/hasSemset"/>
  </subClassOf>
  <owl:disjointWith rdf:resource="http://purl.org/semse/Semset"/>
  <owl:disjointWith rdf:resource="http://purl.org/semse/SubjectProperty"/>
  <vs:term_status>testing</vs:term_status>
  <skos:definition xml:lang="en-EN">Subject concept es un concepto concreto
  que representa la semántica de un concepto o clase definido en un esquema
  de metadatos.</skos:definition>
  <skos:prefLabel xml:lang="en-EN">subject concept</skos:prefLabel>
</owl:Class>
```

Ejemplo III-4: Codificación de la clase *SubjectConcept* incluida en la ontología SEMSE

Como se muestra en el ejemplo, la codificación comienza por la definición del concepto como clase, asignándole el nombre *SubjectConcept*, seguida por el espacio de nombres de la ontología donde se define, SEMSE. A continuación se contextualiza el concepto definiéndolo como subclase de concepto, según el vocabulario SKOS (*skos:Concept*). Seguidamente se incluye una restricción que expresa la necesidad de que, al menos, tenga asociado un *Semset* (como se mostrará más adelante, un *Semset* alberga una etiqueta preferente, descriptor del concepto, y cero o más etiquetas alternativas, sinónimos del descriptor), de este modo se establece que todo concepto debe tener una etiqueta preferente. El siguiente paso es definir las instancias del concepto disjuntas de las del resto de conceptos, para facilitar la inferencia sobre las ontologías generadas. El estado del término se ha establecido utilizando una ontología de anotación de vocabularios (Brickley, D., 2009), concretamente como *testing*. Para finalizar se incluye la definición del término y su etiqueta preferente.

¹⁵ Se tratan en este caso los sinónimos dentro del contexto definido por el esquema original. Como se verá posteriormente, la equivalencia entre conceptos de distintos esquemas será resuelta haciendo uso de la ontología de referencia.

Definido el concepto para albergar las clases de los esquemas, es preciso incluir el equivalente para las propiedades. El objetivo es representar la semántica de las propiedades, definida en el esquema del cual provienen. De este modo, se ha definido una clase denominada *SubjectProperty* o propiedad temática, que representa la semántica de una propiedad, contextualizada por el esquema en el que se incluye. La codificación del concepto se muestra a continuación:

```
<owl:Class rdf:about="http://purl.org/semse/SubjectProperty">
  <isDefinedBy rdf:resource="http://purl.org/semse/" />
  <subClassOf rdf:resource="http://www.w3.org/2004/02/skos/core#Concept" />
  <subClassOf rdf:parseType="Resource">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Restriction" />
    <owl:minCardinality>1</owl:minCardinality>
    <owl:onProperty
      rdf:resource="http://purl.org/semse/hasPropertySemset" />
  </subClassOf>
  <owl:disjointWith rdf:resource="http://purl.org/semse/Semset" />
  <owl:disjointWith rdf:resource="http://purl.org/semse/SubjectConcept" />
  <vs:term_status>testing</vs:term_status>
  <skos:definition xml:lang="en-EN"> Subject property es un concepto concreto
  que representa la semántica de una propiedad definida en un esquema de
  metadatos.</skos:definition>
  <skos:prefLabel xml:lang="en-EN">subject property</skos:prefLabel>
</owl:Class>
```

Ejemplo III-5: Codificación de la clase SubjectProperty incluida en la ontología SEMSE

Como se puede comprobar, la definición es muy similar a la de un *SubjectConcept*, lógicamente, puesto que el objetivo es común, albergar la semántica de un elemento definido en un esquema.

Un vez incluidos los conceptos para albergar la semántica de los elemento, el siguiente paso es definir la estructura necesaria para albergar las distintas etiquetas o nombres que puede tener el elemento. Se aborda así, parcialmente, la sinonimia de los elementos, dado que sólo se van a tener en cuenta los nombres alternativos dentro del contexto específico definido por el esquema que incluye el elemento¹⁶.

El primer paso es incluir un concepto que represente el conjunto de etiquetas que puede tener un concepto. Este concepto, denominado *Semset*, va a albergar al menos una etiqueta preferente y cero o varias etiquetas alternativas. La primera indicará el nombre con el que se identifica el elemento en el esquema y las otras, en su caso, otras formas alternativas de referirse al elemento. La codificación del concepto *Semset* se muestra a continuación:

```
<owl:Class rdf:about="http://purl.org/semse/Semset">
  <isDefinedBy rdf:resource="http://purl.org/semse/" />
  <subClassOf rdf:resource="http://www.w3.org/2004/02/skos/core#Concept" />
  <subClassOf rdf:parseType="Resource">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Restriction" />
    <owl:minCardinality>1</owl:minCardinality>
    <owl:onProperty
      rdf:resource="http://www.w3.org/2004/02/skos/core#prefLabel" />
  </subClassOf>
  <owl:disjointWith rdf:resource="http://purl.org/semse/SubjectConcept" />
```

¹⁶ Como se verá posteriormente, el tratamiento de la sinonimia se completa con la correspondencia entre los elementos y los conceptos de la ontología de referencia.


```

<owl:disjointWith rdf:resource="http://purl.org/semse/SubjectProperty"/>
<vs:term_status>testing</vs:term_status>
<skos:definition xml:lang="en-EN">Semset es un término semánticamente
próximo o sinónimo a un Concepto o Propiedad. Todo concepto y propiedad
tendrán un semset por cada idioma representado.</skos:definition>
<skos:prefLabel xml:lang="en-EN">semset</skos:prefLabel>
</owl:Class>

```

Ejemplo III-6: Codificación de la clase Semset incluida en la ontología SEMSE

Como se muestra en el ejemplo, la codificación comienza por la definición del concepto como clase, asignándole el nombre Semset, seguida por el espacio de nombres de la ontología donde se define, SEMSE. A continuación se contextualiza el concepto definiéndolo como subclase de concepto, según el vocabulario SKOS (*skos:Concept*). Seguidamente se incluye una restricción que expresa la necesidad de que, al menos, tenga asociada una etiqueta preferente. El siguiente paso es definir las instancias del concepto disjuntas de las del resto de conceptos, para facilitar la inferencia sobre las ontologías generadas. El estado del término se ha establecido utilizando la ontología de anotación, concretamente como *testing*. Para finalizar se incluye la definición del término y su etiqueta por defecto.

Una vez incluido el concepto que permite albergar las distintas denominaciones de un concepto, sólo queda definir las propiedades que permiten relacionar los conceptos y propiedades temáticas con sus respectivos conjuntos de etiquetas. Con este fin se han definido dos propiedades *hasSemset* y *hasPropertySemset*. La primera de ellas permite relacionar un Semset con un SubjectConcept, la segunda hace lo mismo entre un Semset y un SubjectProperty. La codificación de ambas propiedades se incluye a continuación:

```

<owl:ObjectProperty rdf:about="http://purl.org/semse/hasSemset">
<isDefinedBy rdf:resource="http://purl.org/semse/">
<domain rdf:resource="http://purl.org/semse/SubjectConcept"/>
<range rdf:resource="http://purl.org/semse/Semset"/>
<subPropertyOf rdf:resource="http://www.w3.org/2004/02/skos/core#related"/>
<vs:term_status>testing</vs:term_status>
<skos:definition xml:lang="en-EN">Propiedad que permite enlazar un concepto
con sus semsets.</skos:definition>
<skos:prefLabel xml:lang="en-EN">has semset</skos:prefLabel>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="http://purl.org/semse/hasPropertySemset">
<isDefinedBy rdf:resource="http://purl.org/semse/">
<domain rdf:resource="http://purl.org/semse/SubjectProperty"/>
<range rdf:resource="http://purl.org/semse/Semset"/>
<subPropertyOf rdf:resource="http://www.w3.org/2004/02/skos/core#related"/>
<vs:term_status>testing</vs:term_status>
<skos:definition xml:lang="en-EN">Propiedad que permite enlazar una
propiedad con sus semsets.</skos:definition>
<skos:prefLabel xml:lang="en-EN">has property semset</skos:prefLabel>
</owl:ObjectProperty>

```

Ejemplo III-7: Codificación de las propiedades hasSemset y hasPropertySemset incluidas en la ontología SEMSE

Como se muestra en el ejemplo, ambas propiedades se definen como propiedades de objeto (*owl:ObjectProperty*) dado que van a establecer relaciones ente objetos (a diferencia de las propiedades de tipo de datos, que relacionan un objeto con un valor, típicamente un literal). A continuación se define el nombre de cada propiedad y el

espacio de nombres de la ontología donde se define, SEMSE. Seguidamente se define el dominio y el rango de cada propiedad, en ambos casos el rango se define sobre Semset, puesto que el rango de valores se define sobre el conjunto de instancias del concepto Semset. Respecto al dominio, en el primer caso se define como SubjectConcept y en el segundo como SubjectProperty. La codificación continúa contextualizando las propiedades, definiéndolas como subclases de la relación simétrica que establece la relación entre dos conceptos, según el vocabulario SKOS (*skos:related*). El siguiente paso establece la versión de las relaciones, mediante el uso de la ontología de anotación, concretamente como *testing*. Finalmente, se incluye una definición para cada propiedad y una etiqueta por defecto.

Una vez definidos los elementos de la ontología SEMSE, sólo queda pendiente abordar el aspecto del plurilingüismo. Como se verá a continuación, al generar las ontologías específicas como descripciones basadas en la ontología SEMSE, se incluirán, tanto para las definiciones de los conceptos como para los Semsets relacionados con SubjectConcepts y SubjectProperties, una versión para cada idioma que se desee soportar. De este modo es posible contemplar tantos idiomas como se desee en la definición de la ontología específica, esto es, en la representación de la semántica de los elementos.

Una vez definida y codificada la ontología SEMSE, padre de las ontologías específicas, el siguiente paso es generar las ontologías específicas a partir de los esquemas originales. Aunque el proceso completo será descrito posteriormente, a continuación se incluye, a título de ejemplo, la representación de un elemento en la ontología específica.

Continuando con el esquema de elementos de Dublin Core como ejemplo, el primer paso en la generación de la ontología específica es incluir el conjunto de metadatos que la define.

Ontología Específica DC Elements

Nombre: DCELEMENTSSO

Título: DCMI dcelements specific ontology

Descripción: This ontology provides the semantics of the elements included in the original schema. It also defines synonyms and allows plurilinguistic searches.

Localización: <http://purl.org/semse/dcelementss/>

Publicado por: Proyecto SEMSE: SEMantic Metadata SEarch

Idioma:	English
Fecha de publicación:	08-02-2010
Versión	1.0

Tabla III-6: Metadatos para la definición de la ontología específica DCELEMENTSSO

Como en los anteriores casos, la ontología es publicada haciendo uso de una URL persistente, concretamente en el espacio de nombres <http://purl.org/semse/dcelementso/>. El nombre de la ontología proviene del nombre del esquema original seguido del acrónimo ‘so’ (*specific ontology*).

Al igual que en la ontología SEMSE, el lenguaje seleccionado para la definición de las ontologías específicas es OWL, complementado con el uso del vocabulario SKOS en los casos en los que ha pretendido proporcionar mayor semántica a las descripciones de los conceptos.

Definidos los metadatos de la ontología específica, sólo resta incluir la representación de la semántica de los elementos incluidos en el esquema original. Continuando como ejemplo con el elemento *dc:rights*, a continuación se muestra la codificación del término, incluyendo sus semsets asociados:

```
<semse:SubjectProperty rdf:about="&dcso;sp/rights">
  <semse:hasPropertySemset rdf:resource="&dcso;semset/es/Derechos"/>
  <semse:hasPropertySemset rdf:resource="&dcso;semset/en/Rights"/>
  <definition xml:lang="es-ES">Información sobre los derechos en y
  sobre el recurso. Típicamente, la información sobre derechos incluye
  una declaración acerca de varios derechos de propiedad asociados con
  el recurso, incluyendo derechos de propiedad
  intelectual.</definition>
  <definition xml:lang="en-EN">Information about rights held in and
  over the resource. Typically, rights information includes a
  statement about various property rights associated with the
  resource, including intellectual property rights.</definition>
</semse:SubjectProperty>

<semse:Semset rdf:about="&dcso;semset/es/Derechos">
  <prefLabel xml:lang="es-ES">derechos</prefLabel>
</semse:Semset>

<semse:Semset rdf:about="&dcso;semset/en/Rights">
  <prefLabel xml:lang="en-EN">rights</prefLabel>
</semse:Semset>
```

Ejemplo III-8: Codificación del término dc:rights y sus semsets asociados

A partir del ejemplo se puede comprobar cómo la codificación comienza por establecer el concepto como SubjectProperty, debido a que proviene de una propiedad en el esquema original. Esta distinción permitirá la posterior restricción y especialización en

los ámbitos de búsqueda. Junto con el tipo del concepto se incluye el espacio de nombres que será descrito en el apartado de generación de las ontologías específicas, no obstante, destacar que el espacio de nombres ubica el concepto dentro del subdominio ‘sp’ (*subject property*), se trata de nuevo de una clasificación destinada a la especialización en el ámbito de las búsquedas. Junto con espacio de nombres se especifica el nombre del concepto que coincide con el nombre del término original, en este caso, *rights*. Seguidamente se definen las relaciones del concepto con sus semsets, más concretamente, un semset por cada lenguaje soportado por la ontología. De nuevo destacar que también los semsets son clasificados, en su espacio de nombres, incluyendo el idioma en la definición del subdominio. Para finalizar la codificación del concepto, se incluye la definición del mismo, concretamente, una por cada idioma soportado, en este caso español e inglés. Una vez definido el concepto, el ejemplo muestra la codificación de los dos semsets asociados al mismo, en español e inglés. En ambos casos se incluye únicamente una etiqueta preferente, aunque podrían haberse incluido una o más etiquetas alternativas, como sinónimos para referirse al concepto, lógicamente, en cualquiera de los idiomas soportados.

Una vez planteado el modo de representar la semántica de los esquemas, mediante la definición de la ontología SEMSE y sus descripciones como ontologías específicas, es preciso analizar la solución conforme a los criterios de calidad:

- *Extensibilidad*: La solución propuesta se ha diseñado siguiendo este criterio, en cuanto a que es posible generar una representación como ontología específica de cualquier esquema, al tiempo que dicho esquema es independiente de la representación o alteración de su representación semántica. También el plurilingüismo y la sinonimia han sido definidos acorde con este criterio. En el primer caso es posible añadir cuantos idiomas se deseen en la definición de la ontología. En el segundo, un concepto puede tener cualquier número de sinónimos o etiquetas alternativas, dentro del contexto definido por el esquema original.
- *Compatibilidad*: La solución propuesta se adecúa a este criterio, en cuanto a que se ha diseñado para garantizar la independencia de la versión original del esquema. De este modo, definición de la semántica, los lenguajes soportados o la sinonimia de términos se gestionan de forma transparente al esquema original que, como elemento independiente, no precisa alteración o modificación alguna. También se ha tenido en cuenta este aspecto a la hora de seleccionar los lenguajes de definición, de este modo, se han utilizado lenguajes de uso ampliamente extendido, compatibles con la publicación en el entorno Web.
- *Reutilización*: Se ha atendido a este criterio en la definición de la ontología padre, SEMSE, así como en el proceso de generación de las descripciones a partir de ésta. La definición la ontología SEMSE incluye los conceptos y relaciones que serán reutilizados en la generación de las ontologías específicas. Además, en la definición de la ontología SEMSE se han reutilizado ontologías y vocabularios existentes, incluyendo el uso de UMBEL como ontología base, e igualmente se han reutilizado en la generación de las ontologías específicas.

Para finalizar la definición de la representación semántica de ontologías, a continuación

se incluye un ejemplo con los artefactos propuestos a tal fin. Con este objetivo, en la *Figura III-3* se muestra la representación semántica del elemento *rights* del esquema de metadatos Dublin Core.

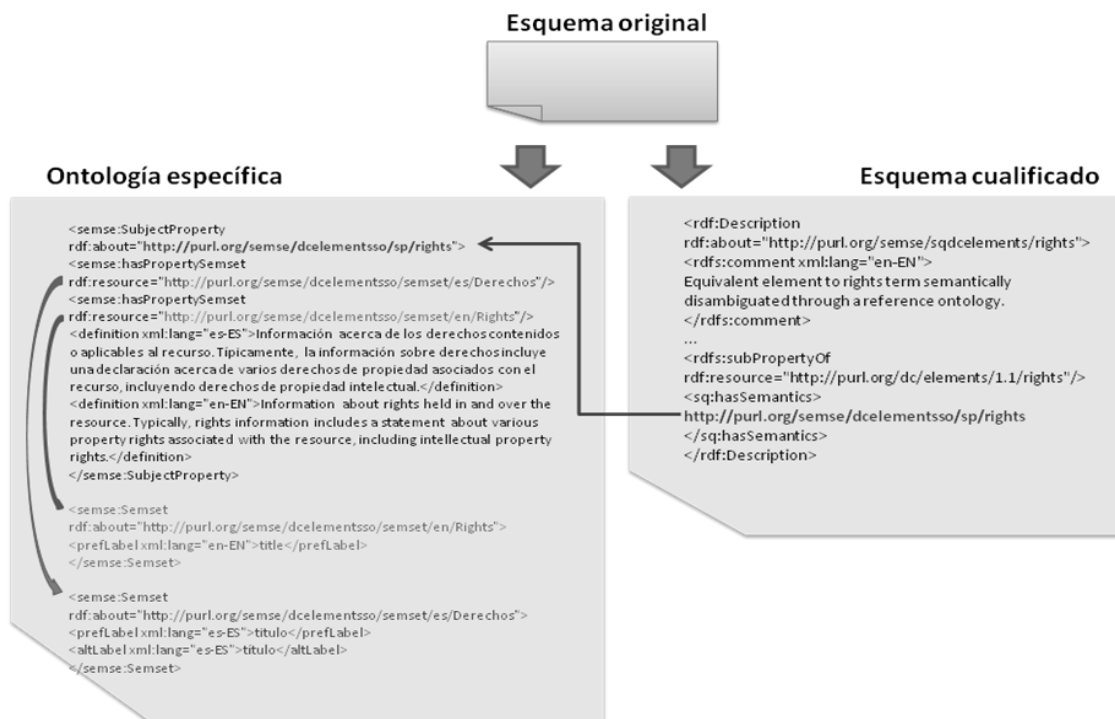


Figura III-3: Ejemplo de representación semántica del elemento 'rights' definido en el esquema de metadatos de Dublin Core

Tal y como se muestra en la figura, el esquema original, esquema en el que se definen originalmente los elementos y que se desea representar semánticamente, es la fuente a partir de la cual se obtienen las dos representaciones propuestas. A la derecha se sitúa la representación cualificada del esquema, donde se incluye un elemento por cada término definido en el esquema original, y en cuya definición se incorpora el cualificador semántico *sq:hasSemantics*. A través de este cualificador se incluye la semántica del elemento articulada mediante la URI al concepto que la alberga, definido en la ontología específica. La ontología específica, situada a la izquierda, representa un concepto por cada término definido en el esquema original, en este caso de tipo *SubjectProperty*. Dicho concepto es el destino de la URI incluida en el esquema cualificado y, además de la definición del término, una por cada idioma soportado, incluye relaciones con los *Semsets* que albergan etiquetas con las distintas formas de referirse al concepto en cuestión. Los *Semsets*, uno por cada idioma soportado, incluyen una etiqueta preferente y cero o más etiquetas alternativas.

Del mismo modo que con el ejemplo, el proceso se aplica para cada elemento incluido en el esquema original hasta completar la representación de éste.

3.1.2 Búsqueda y selección de esquemas

El siguiente paso dentro del proceso de representación ontológica de esquemas consiste

en la selección de un primer conjunto de esquemas de metadatos sobre el cual poner en práctica la representación semántica propuesta.

Con este objetivo, se inició un proceso de evaluación de esquemas de metadatos, basado en los criterios: ámbito, estabilidad, estandarización y popularidad.

En primer lugar, el conjunto de esquemas a representar no podía restringirse a un único elemento, era necesario seleccionar varios esquemas con el objetivo de demostrar cómo la propuesta es válida para la representación de cualquier esquema.

El conjunto de esquemas, en segundo lugar, debía tener un ámbito heterogéneo, esto es, los esquemas debían provenir de distintos dominios de conocimiento. El porqué de esta decisión se debe a la posibilidad de demostrar, de este modo, cómo se logra la interoperatividad de esquemas definidos para distintos dominios. Añadir que, si bien se han seleccionado dominios distintos, estos no están aislados sino que se han elegido de forma que estén próximos en cuanto a su ámbito. Con estas premisas, los dominios elegidos han sido: descripción de recursos bibliográficos, descripción de personas y relaciones personales, descripción información personal de contacto, descripción de tarjetas de visita, descripción de currículos y descripción de proyectos. Como se puede deducir de los dominios seleccionados, aunque son distintos, poseen solapamientos parciales que van a permitir ilustrar la interoperatividad que proporciona propuesta. Como ejemplo se puede intuir que la definición de conceptos como persona, organización o proyecto, entre otros, va a tener distintas representaciones en varios de los dominios seleccionados.

En tercer lugar, la selección de esquemas debía tener en cuenta la estandarización de los mismos. Era necesario que éstos tuvieran distinto grado de estandarización para mostrar, por un lado, como la propuesta no depende de este aspecto y, por otro, como permite completar y formalizar la definición de los esquemas, tanto en su estructura como en su semántica.

El siguiente criterio a tener en cuenta en el proceso de selección es el de estabilidad. Como en el caso de la estandarización, era necesario incluir esquemas con distinto nivel de estabilidad para mostrar cómo la propuesta no depende de este aspecto, así como para evaluar cómo responde a problemas derivados de este criterio, principalmente, modificaciones en la definición del esquema original.

Finalmente, la selección debía tener en cuenta la popularidad de los esquemas a incluir. De este modo, cuanto más popular es un esquema, mayor es su uso y mayor número de descripciones, documentos, habrán sido generados a partir de él. Dentro de cada ámbito concreto, se ha seleccionado el esquema más popular, teniendo en cuenta los criterios anteriores.

A partir de los criterios expuestos, el conjunto de esquemas seleccionados ha sido: *DCMI Dublin Core Elements Set* (dcelements) (Dublin Core Metadata Initiative, 2008b), *W3C vCard Ontology* (vcard-rdf) (World Wide Web Consortium, 2010b), *Friend of a Friend* (foaf) (Brickley, D.; Miller, L., 2010), *Description of a Career* (doac) (Parada, R., 2008), *Description of a Project* (doap) (Dumbill, E., 2008) y *Personal Information Markup* (pim) (Berners-Lee, T., 2007).

A continuación, *Tabla III-7*, se incluye un resumen de los valores de cada uno de los esquemas para los criterios de selección expuestos anteriormente.

Nombre	Ámbito	Estandarización	Estabilidad	Popularidad
Dcelements	Descripción de recursos bibliográficos	Alta	Alta	Alta
Vcard-rdf	Tarjetas de visita	Alta	Alta	Media
FOAF	Descripción de personas y relaciones personales	Alta	Baja	Alta
DOAC	Descripción de currículos	Baja	Baja	Baja
DOAP	Descripción de proyectos	Media	Media	Baja
PIM	Descripción de información personal de contacto	Baja	Media	Media

Tabla III-7: Conjunto inicial de esquemas y valoración según los criterios de selección propuestos

Los valores obtenidos para cada criterio provienen del análisis de los esquemas en cuanto a:

- **Estandarización**, donde se ha valorado la formalización del esquema en cuanto a notación, definición de sus elementos, etc.
- **Estabilidad**, medida en la novedad de los últimos cambios experimentados por el esquema y en el proceso de actualización del esquema.
- **Popularidad**, medida en documentos semánticos asociados al esquema, según buscadores semánticos como Swoogle (Swoogle, 2007b).

En los siguientes apartados se trabajará sobre este conjunto de esquemas, poniendo en práctica las soluciones propuestas y resolviendo los problemas que se encuentren durante su análisis. Debe destacarse, no obstante, que la solución se plantea en términos generales, de modo que es posible seleccionar cualquier otro conjunto de esquemas para su cualificación.

3.1.3 Representación semántica de los esquemas seleccionados

El objetivo de este apartado es aplicar la propuesta de representación semántica al conjunto de esquemas de metadatos seleccionado. De este modo, se pretende analizar y resolver los posibles problemas de representación, a la vez que se muestra el ejemplo de aplicación y se establecen los pasos para una metodología de representación semántica para cualquier esquema de metadatos.

Partiendo de la propuesta de cualificación semántica planteada y del conjunto de

esquemas de metadatos seleccionado, en este apartado se generarán las versiones cualificadas semánticamente, interpretables por personas y computadoras, así como las representaciones ontológicas que albergarán su correspondiente semántica.

3.1.3.1 Cualificación semántica de los esquemas seleccionados

Comenzando el proceso de cualificación por la generación de la versión interpretable por personas, tal y como se planteaba anteriormente en la propuesta, el objetivo es generar un nuevo esquema cualificado semánticamente, en formato HTML y publicado en Web, de forma que esté accesible para su localización y uso.

Después de analizar distintas versiones de esquemas publicados, se ha elaborado un conjunto de recomendaciones para la documentación de esquemas, validadas en varias publicaciones como: la *Fifth International Semantic Web Conference* (Palacios, V. et al., 2006a), la *Conferência Ibérica de Sistemas e Tecnologias de Informação* (Palacios, V. et al., 2006b) o la *International Conference on Dublin Core and Metadata Applications* (Palacios, V. et al., 2006c). Este conjunto de recomendaciones se presentan a continuación como una serie de pasos y un índice resumido como plantilla e incluida en el *Anexo A: Plantilla de descripción de Esquemas de Metadatos*.

Antes de iniciar el proceso de cualificación, es importante comprender la estructura y significado de los elementos incluidos en el esquema. En este punto, contar con una versión en RDF del esquema puede facilitar esta tarea. En el caso del esquema seleccionado, se han analizado tanto los documentos de definición como los que describen su uso y gramática. Como se puede consultar en el *Anexo A: Plantilla de descripción de Esquemas de Metadatos*, se ha incluido una plantilla de ejemplo que puede ser completada para describir futuros esquemas de metadatos.

1. Incluir un registro de estado del documento. El objeto de este registro es proporcionar información sobre el documento y su estado. Entre esta información se encuentra el título, autor, identificador del documento, etc. A continuación se muestra el registro de estado tomando como ejemplo el esquema de elementos de Dublin Core:

Título:	Dublin Core Metadata Element Set, Version 1.1 Semantically Qualified.
Autor:	Proyecto SEMSE: SEMantic Metadata SEarch
Identificador:	sqdcelements
Fecha de creación:	02-22-2010
Última versión:	http://purl.org/semse/sqdcelements/
Sustituye a:	No aplica
Traducciones:	No aplica

Estado del documento:	Testing
Descripción:	This schema provides a semantically disambiguated element set obtained from the original schema. Each element maintains the name of the original schema element and is defined as a subtype of it.
Fecha de aprobación:	No aplica
Lenguaje	English

Tabla III-8: Registro de estado del esquema Dublin Core Cualificado Semánticamente

2. Incluir información relevante para la interpretación del esquema

- Incluir el objetivo del esquema: Incluir un breve resumen del objetivo del esquema. En el caso del esquema seleccionado como ejemplo, el objetivo es la definición de los elementos que desambiguan semánticamente los elementos del esquema de metadatos de Dublin Core.
- Incluir el ámbito del esquema: Incluir una breve descripción del ámbito de publicación y uso del esquema. En el esquema seleccionado, en el ejemplo, el ámbito es público para su uso por cualquier persona u organización.
- Incluir definiciones, acrónimos y abreviaturas que faciliten la interpretación del esquema. En el esquema seleccionado, podrían incluirse distintas definiciones como el término *espacio de nombres* o acrónimos como DCMI.
- Incluir referencias útiles para la interpretación del esquema y a otros esquemas utilizados en su definición. En el esquema seleccionado, podrían incluirse referencias a documentos del DCMI o a esquemas como el de Dublin Core.
- Describir los atributos necesarios para la definición de cada elemento del esquema. Se recomienda la definición de un conjunto mínimo y un adicional recomendado. A continuación se muestra el conjunto recomendado mínimo, extraído del *Anexo F*.

Nombre:	<i>Identificador unívoco asignado al término dentro del esquema.</i>
URI:	<i>URI utilizado para identificar unívocamente un término.</i>
Etiqueta:	<i>Etiqueta asignada al término.</i>
Comentario:	<i>Información adicional sobre el término o su aplicación.</i>
Refina:	<i>Referencia al término original del esquema a cualificar.</i>

Tipo de término:	<i>Tipo de término, tal y como se define en el documento que contiene los principios gramaticales.</i>
Semántica:	<i>URI al concepto de la ontología específica.</i>
Esquema:	<i>URI al esquema en el que se define.</i>
Fecha de Publicación:	<i>Fecha en la que se ha publicado el término</i>

Tabla III-9: Conjunto mínimo de atributos para la definición de un término

3. Incluir un elemento cualificado semánticamente para cada elemento del esquema original. Siguiendo el conjunto de atributos definido en el punto anterior, definir cada término proporcionando la información correspondiente según cada atributo. A continuación se incluye como ejemplo la definición del elemento ‘rights’ del esquema de elementos de Dublin Core, cualificado semánticamente:

Nombre:	<i>rights</i>
URI:	<i>http://purl.org/semse/sqdelements/rights</i>
Etiqueta:	<i>rights</i>
Comentario:	<i>Equivalent element to rights term semantically disambiguated through a reference ontology.</i>
Refina:	<i>http://purl.org/dc/elements/1.1/rights</i>
Tipo de término:	<i>http://www.w3.org/1999/02/22-rdf-syntax-ns#Property</i>
Semántica:	<i>http://purl.org/semse/dcelementsso/sp/rights</i>
Esquema:	<i>http://purl.org/semse/sqdelements/</i>
Fecha de Publicación:	<i>02-22-2010</i>

Tabla III-10: Definición del elemento ‘rightsSem’ del esquema Dublin Core Cualificado Semánticamente

4. Incluir el extracto del documento de gramática para aquellos tipos de términos definidos para el esquema y no incluidos en el esquema original. En el caso de no definirse nuevos tipos, el apartado no será de aplicación y deberá etiquetarse como *No Aplicable*. Las referencias a los documentos de gramática donde se definen los tipos utilizados en el esquema original, deberán estar incluidas en el apartado de referencias.

5. Incluir la versión RDF del esquema cualificado semánticamente (opcional). En caso de contarse con la versión RDF del esquema, o una vez generada (como se verá en el siguiente apartado), se recomienda la inclusión de la misma en este documento.

Una vez definida la versión del esquema interpretable por personas, se procederá a definir la versión interpretable por computadoras. En este caso, tal y como se mostraba en los ejemplos de codificación incluidos anteriormente, el proceso consiste en la codificación de los elementos, definidos en la versión ya desarrollada, haciendo uso del lenguaje RDF.

A continuación se muestra un extracto del esquema de Dublin Core Cualificado Semánticamente, que se corresponde con los ejemplos mostrados para la versión interpretable por personas.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
  <!ENTITY dc "http://purl.org/dc/elements/1.1/">
  <!ENTITY dcterms "http://purl.org/dc/terms/">
  <!ENTITY sq "http://purl.org/semse/sq/">
  <!ENTITY sqdcelelements "http://purl.org/semse/sqdcelelements/">
  <!ENTITY dcelementssp "http://purl.org/semse/dcelementssso/sp/">
]>
<rdf:RDF xmlns:dc="&dc;"
  xmlns:dcterms="&dcterms;"
  xmlns:rdf="&rdf;"
  xmlns:rdfs="&rdfs;"
  xmlns:sqdcelelements="&sqdcelelements;"
  xmlns:sq="&sq;"
  xmlns:vs="http://www.w3.org/2003/06/sw-vocab-status/ns#">

  <rdf:Description rdf:about="&sqdcelelements;">
  <dcterms:title xml:lang="en-EN">Dublin Core Metadata Element Set,
  Version 1.1 Semantically Qualified.</dcterms:title>
  <dcterms:title xml:lang="es-ES">Dublin Core Metadata Element Set,
  Version 1.1 Cualificado Semánticamente.</dcterms:title>
  <rdfs:comment>This schema provides a semantically disambiguated
  element set obtained from the original schema. Each element maintains
  the name of the original schema element and is defined as a subtype of
  it.</rdfs:comment>
  <rdfs:comment>Este esquema proporciona un conjunto de elementos
  desambiguados semánticamente y generados a partir del esquema
  original. Cada elemento mantiene el nombre del elemento del esquema
  original aunque se define como subtipo del mismo.</rdfs:comment>
  <dcterms:publisher xml:lang="en-EN">Proyecto SEMSE: SEMantic Metadata
  SEarch</dcterms:publisher>
  <dcterms:language xml:lang="en-EN">English</dcterms:language>
  <dcterms:issued>02-22-2010</dcterms:issued>
  <vs:term_status>testing</vs:term_status>
  </rdf:Description>

  <rdf:Property rdf:about="&sqdcelelements;title">
  ...
  <rdf:Property rdf:about="&sqdcelelements;rights">
  <rdfs:label xml:lang="en-EN">rights</rdfs:label>
```

```

<rdfs:comment xml:lang="en-EN">Equivalent element to rights term
semantically disambiguated through a reference
ontology.</rdfs:comment>
<rdfs:comment xml:lang="es-ES">Elemento equivalente al término rights,
pero desambiguado semánticamente mediante una ontología de
referencia.</rdfs:comment>
<rdfs:isDefinedBy rdf:resource="&sqdcelements;"/>
<dcterms:issued xml:lang="en-EN">02-22-2010</dcterms:issued>
<rdf:type rdf:resource="&rdf;Property"/>
<rdfs:subPropertyOf rdf:resource="&dc;rights"/>
<sq:tieneSemantica>&dcelementssp;rights</sq:tieneSemantica>
    
```

Ejemplo III-9: Extracto de codificación en RDF del esquema Dublin Core Cualificado Semánticamente

Como se puede observar a partir del ejemplo, el esquema comienza incluyendo los espacios de nombres, entre los que se encuentran: el esquema original (*dc*), el esquema de cualificadores semánticos (*sq*), el esquema cualificado (*sqdcelements*) y la ontología específica (*dcelementssso*), entre otros. Una vez definidos los espacios de nombres, el esquema incluye una descripción donde se incluye: título, publicador, descripción, fecha de creación, entre otros. Finalizada la descripción del esquema se irían incluyendo, por cada elemento del esquema original, el término cualificado correspondiente. En este caso, se han dejado indicados los elementos a excepción de *rights*, cuya descripción se ha incluido completa.

Una vez definida la metodología para la cualificación semántica de esquemas, y de modo equivalente al ilustrado mediante los ejemplos anteriores, se ha procedido a la generación de los esquemas cualificados semánticamente. Concretamente, se ha generado un esquema cualificado para cada uno de los esquemas seleccionados en el apartado anterior 3.1.2 *Búsqueda y selección de esquemas*. En la *Tabla III-11* se resumen los esquemas generados, incluyendo sus nombres y espacios de nombres.

Esquema original	Esquema cualificado	Espacio de nombres
Dcelements	sqdcelements	http://purl.org/semse/sqdcelements
Vcard-rdf	sqvcard	http://purl.org/semse/sqvcard
FOAF	sqfoaf	http://purl.org/semse/sqfoaf
DOAC	sqdoac	http://purl.org/semse/sqdoac
DOAP	sqdoap	http://purl.org/semse/sqdoap
PIM	sqpim	http://purl.org/semse/sqpim

Tabla III-11: Resumen de los esquemas cualificados generados

Aplicando el método de cualificación propuesto, se han generado las representaciones cualificadas de los esquemas originales. Concretamente, las URIs mostradas en la tabla anterior referencian las versiones en RDF (OWL en el caso del esquema sqvcard, al ser este el lenguaje de definición del esquema original). De cara a una interpretación

adecuada de los esquemas por parte de las personas, en la plataforma de soporte se encuentra en desarrollo un generador de reportes compartido vía web, y accesible mediante la tecnología *JasperReports* (JasperForge, 2009a) y *JasperServer* (JasperForge, 2009b). A través de esta aplicación se pueden obtener versiones de los esquemas amigables para el usuario, en formatos como HTML o PDF. Este servicio será descrito en el apartado 3.5 *Proceso de desarrollo del sistema informático de soporte*.

3.1.3.2 Representación ontológica de la semántica de los esquemas seleccionados

Una vez creadas las versiones cualificadas de los esquemas originales, el siguiente paso es generar las ontologías específicas correspondientes a dichos esquemas.

Cabe destacar que, a diferencia de los esquemas cualificados semánticamente, las ontologías específicas no están dirigidas al usuario final y no precisan, en principio, de una versión amigable para el usuario final. No obstante, a través del generador de reportes incluido en la plataforma de soporte, mencionado en el apartado anterior, se puede obtener una versión amigable de las ontologías específicas en distintos formatos.

Antes de iniciar el proceso de generación de las ontologías específicas, es importante comprender el significado de cada uno de los elementos incluidos en el esquema. En este punto, contar con la especificación del formal puede facilitar esta tarea. En este punto, el grado de estandarización de los esquemas juega un papel fundamental, como se verá a continuación.

Siguiendo el planteamiento propuesto en el apartado 3.1.1.2 *Ontologías Específicas* se han generado las ontologías correspondientes a los esquemas seleccionados en el apartado 3.1.2 *Búsqueda y selección de esquemas*. Concretamente, los pasos se pueden resumir en:

1. Incluir el conjunto de metadatos que define la ontología específica, incluyendo: nombre, título, descripción, localización, publicador, idioma, fecha de publicación y versión.
2. Incluir, para cada elemento definido en el esquema original, la representación semántica según corresponda: *SubjectConcept* si el elemento original es una clase o *SubjectProperty* si el elemento original es una propiedad. Destacar que, en este punto, es fundamental incluir la definición precisa del término para cada uno de los lenguajes que se deseen cubrir. El grado de estandarización del esquema original es un aspecto clave dado que, como se ha podido comprobar en durante el proceso, aquellos esquemas con mayor grado de estandarización incluyen definiciones precisas para todos sus elementos. Esto no sucede para aquellos esquemas poco estandarizados, como doac, doap o pim. En estos casos ha sido necesario completar, precisar, e incluso definir, la semántica de varios de sus elementos, siempre dentro del contexto semántico establecido por el esquema. Por contexto semántico del esquema debe entenderse el significado que, dentro del esquema, se le da al término. De este modo, un elemento puede tener diferente semántica, en función del esquema que lo define. Las definiciones modificadas durante este proceso, y que difieren de las incluidas en el esquema original, han sido etiquetadas como [SEMSE] para su

posterior identificación.

- Incluir, para cada elemento representado, el *Semset* asociado, uno por cada idioma que se desee cubrir, donde se incluye la etiqueta preferente del elemento y sus etiquetas alternativas o sinónimos. En este punto destacar que todos los conceptos y propiedades poseen una etiqueta preferente y, sólo algunos, etiquetas alternativas. Es importante tener en cuenta que las etiquetas alternativas se han obtenido haciendo uso de diccionarios de definiciones y sinónimos, siempre dentro del contexto semántico definido por el esquema. Así, el conjunto de sinónimos de un mismo elemento puede diferir en función del esquema en el que se define.

Con el fin de ilustrar el método de generación de las ontologías específicas, a continuación se muestra un extracto de la ontología específica correspondiente al esquema *Description of a Project (DOAP)*:

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY doapsosc "http://purl.org/semse/doapso/sc/">
  <!ENTITY doapsosp "http://purl.org/semse/doapso/sp/">
  <!ENTITY doapsosemseten "http://purl.org/semse/doapso/semset/en/">
  <!ENTITY doapsosemsetes "http://purl.org/semse/doapso/semset/es/">
  <!ENTITY doapso "http://purl.org/semse/doapso/">
]>
<rdf:RDF xmlns="http://www.w3.org/2004/02/skos/core#"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:semse="http://purl.org/semse/"
  xmlns:skos="http://www.w3.org/2004/02/skos/core#"

  <owl:Ontology rdf:about="&doapso;">
    <dcterms:title>Description Of A Project specific ontology</dcterms:title>
    <dcterms:description>This ontology provides the semantics of the elements
    included in the original schema. It also defines synonyms and allows
    plurilinguistic searches.</dcterms:description>
    <dcterms:publisher xml:lang="en-EN">Proyecto SEMSE: SEmantic Metadata
    SEarch</dcterms:publisher>
    <dcterms:language xml:lang="en-EN">English</dcterms:language>
    <dcterms:issued>08-02-2010</dcterms:issued>
    <owl:versionInfo>"1.0"</owl:versionInfo>
  </owl:Ontology>

  <semse:SubjectConcept rdf:about="&doapsosc;CVSRepository">
    <semse:hasSemset rdf:resource="&doapsosemsetes;RepositorioCVS"/>
    <semse:hasSemset rdf:resource="&doapsosemseten;CVSRepository"/>
    <definition xml:lang="es-ES">Repositorio de código fuente CVS.</definition>
    <definition xml:lang="en-EN">CVS source code repository.</definition>
  </semse:SubjectConcept>
  <semse:Semset rdf:about="&doapsosemsetes;RepositorioCVS">
    <altLabel xml:lang="es-ES">repositorio CVS</altLabel>
    <prefLabel xml:lang="es-ES">repositorio sistema versiones
    concurrentes</prefLabel>
  </semse:Semset>
  <semse:Semset rdf:about="&doapsosemseten;CVSRepository">
    <altLabel xml:lang="en-EN">CVS repository</altLabel>
    <prefLabel xml:lang="en-EN">concurrent versions system
    repository</prefLabel>
  </semse:Semset>
  ...
  <semse:SubjectProperty rdf:about="&doapsosp;language">
    <semse:hasSemset rdf:resource="&doapsosemsetes;lenguaje"/>
```

```

    <semse:hasSemset rdf:resource="&doapsosemseten;language"/>
    <definition xml:lang="es-ES">Codigo ISO del lenguaje al que ha sido
traducido el proyecto.</definition>
    <definition xml:lang="en-EN">ISO language code a project has been
translated into.</definition>
</semse:SubjectProperty>
<semse:Semset rdf:about="&doapsosemsetes;lenguaje">
  <altLabel xml:lang="es-ES">idioma</altLabel>
  <prefLabel xml:lang="es-ES">lenguaje</prefLabel>
</semse:Semset>
<semse:Semset rdf:about="&doapsosemseten;language">
  <prefLabel xml:lang="en-EN">language</prefLabel>
</semse:Semset>

```

Ejemplo III-10: Extracto de codificación de la ontología específica correspondiente al esquema DOAP

En el ejemplo anterior se han incluido, además de los espacios de nombres y el conjunto de metadatos del esquema, una clase *CVSRepository* y una propiedad *language*. Como se comentaba anteriormente, las definiciones son específicas del esquema en el que se define el elemento original, de este modo, la propiedad *language* puede encontrarse en otros esquemas con una definición distinta, por ejemplo, en el esquema *dcelements* se incluye el elemento *language* con la siguiente definición: “*Idioma del recurso. El método más efectivo recomendado es utilizar un vocabulario controlado como la RFC 4646*”. Esta disparidad semántica es una de las principales dificultades a abordar si se desea lograr interoperatividad entre esquemas. Como se mostrará en capítulos posteriores, es la ontología de referencia la que proporciona la semántica común y permite que ambos elementos sean interoperables.

Respecto al problema del grado de estandarización de los esquemas, a continuación se muestra un extracto del esquema *Personal Information Markup* en el que se muestra la codificación del concepto *Address*.

```

<semse:SubjectConcept rdf:about="&pimsosc;Address">
  <semse:hasSemset rdf:resource="&pimsosemsetes;Direccion"/>
  <semse:hasSemset rdf:resource="&pimsosemseten;Address"/>
  <definition xml:lang="es-ES">Dirección Postal.[SEMSE]</definition>
  <definition xml:lang="en-EN">Postal Address.[SEMSE]</definition>
</semse:SubjectConcept>

```

Ejemplo III-11: Extracto de codificación de la ontología específica correspondiente al esquema PIM

Tal y como se puede comprobar en el ejemplo, ha sido preciso incorporar las definiciones del concepto, inexistentes en el esquema original. Dichas definiciones han sido establecidas según el contexto del esquema original y, para su posterior identificación, etiquetadas con *[SEMSE]*.

Aplicando el método propuesto para la generación de ontologías específicas, se ha procedido a generar una ontología específica para cada esquema seleccionado en el apartado 3.1.2 *Búsqueda y selección de esquemas*. En la *Tabla III-12* se resumen las ontologías específicas generadas, incluyendo sus nombres y espacios de nombres.

Esquema original	Esquema cualificado	Espacio de nombres
------------------	---------------------	--------------------

Dcelements	dcelementsso	http://purl.org/semse/dcelementsso
Vcard-rdf	vcardso	http://purl.org/semse/vcardso
FOAF	foafso	http://purl.org/semse/foafso
DOAC	doacso	http://purl.org/semse/doacso
DOAP	doapso	http://purl.org/semse/doapso
PIM	pimso	http://purl.org/semse/pimso

Tabla III-12: Resumen de las ontologías específicas generadas

Como se muestra en tabla anterior, las URIs referencian a las ontologías específicas, una vez publicadas en Web. En el siguiente apartado se resumirá el proceso de publicación y las decisiones tomadas en su realización.

3.1.4 Publicación de esquemas semánticos

Una vez generadas las representaciones semánticas de los esquemas, se ha procedido a su publicación en Web. El objetivo de la publicación es poner a disposición de los usuarios las representaciones generadas, permitiendo así su acceso, uso y reutilización, directamente desde un navegador. Además, en combinación con la plataforma informática de soporte, el usuario podrá consultar y utilizar las representaciones de forma amigable, a través de una aplicación Web.

Antes de la publicación de los esquemas y ontologías, ha sido necesario tomar una serie de decisiones de diseño que se resumen a continuación:

- Definición de la política de nombrado: Los esquemas cualificados recibirán el nombre del esquema original, precedido por el acrónimo ‘sq’ (*semantically qualified*), por ejemplo: *sqdcelements* será el esquema cualificado semánticamente correspondiente al esquema *dcelements*. De forma similar, las ontologías específicas serán nombradas a partir del esquema original, seguidas del acrónimo ‘so’ (*specific ontology*), por ejemplo: *dcelementssso* será la ontología específica correspondiente al esquema *dcelements*.
- Definición del método de publicación: Siguiendo las indicaciones del DCMI sobre Políticas de Espacios de Nombres (Dublin Core Metadata Initiative, 2001) se ha hecho uso de una URL persistente, concretamente, la proporcionada por el servicio *Persistent Uniform Resource Locator (PURL)* (Online Computer Library Center, 2010b). A través de este servicio se garantiza la accesibilidad del esquema a través de una URI que permanece constante, independientemente de cualquier posible cambio en la dirección real del esquema. De este modo, la URL base definida para la publicación de las representaciones generadas ha sido: <http://purl.org/semse>.
- Definición de los espacios de nombres: A partir de las recomendaciones del *World Wide Web Consortium* para la publicación de vocabularios RDF

(Berrueta, D.; Phipps, J., 2008), se decidió utilizar un esquema basado en *slash* para el acceso a las representaciones y su contenido. Respecto a la estructura, partiendo de la URL base, se han definido subdominios para cada uno de los esquemas cualificados así como para las ontologías específicas. De este modo, cada representación tiene un subdominio asignado, definido por el nombre asignado anteriormente, que permite el acceso unívoco a su contenido, por ejemplo: <http://purl.org/semse/sqdcelements/> es el subdominio asignado al esquema cualificado semánticamente correspondiente al esquema *dcelements*. Respecto a las ontologías específicas, con el fin de facilitar la organización y recuperación de elementos, dentro del subdominio correspondiente se han generado subdominios específicos, en función del contenido de los mismos. Concretamente, se ha incluido un subdominio en función del concepto representado: ‘*sc*’ (*subject concept*) para albergar conceptos temáticos, ‘*sp*’ (*subject property*) para albergar propiedades temáticas, y *semset* para albergar los semsets asociados a conceptos y propiedades temáticas. Para el caso de los semset, destacar que, además, se ha incluido un subdominio por cada idioma cubierto por la ontología. Así, son ejemplos de URIs: <http://purl.org/semse/foafso/sc/Person> concepto *Person* de la ontología *foafso*, <http://purl.org/semse/dcelementsso/sp/title> propiedad *title* de la ontología *dcelementsso*, <http://purl.org/semse/dcelementsso/semset/es/titulo> semset en idioma español correspondiente al concepto *title* de la ontología *dcelementsso*.

Una vez diseñado el modo de publicar esquemas cualificados y ontologías específicas, el siguiente paso ha consistido en desplegar la infraestructura necesaria a tal fin. La infraestructura se puede resumir en un servidor web desde el cual compartir las representaciones generadas. Concretamente, se ha desplegado en un PC Intel Core 2 Quad 2.4GHz, 4GB RAM y 500GB HDD, en el que se ha configurado un servidor web *Apache2 v2.2.11*, instalado sobre un sistema operativo *Ubuntu 9.04 Jaunty Jackalope*.

Una vez instalado el servidor web se ha procedido a añadir los recursos a publicar. Primeramente se ha generado una estructura de directorios para albergar los esquemas cualificados (directorio *sqscheme*) y las ontologías específicas (directorio *semanticontologies*). Una vez ubicados los recursos en el servidor, el siguiente paso ha sido crear las reglas de acceso a dichos recursos, en función de la estructura de directorios definida y de la definición de los espacios de nombres. El método para incluir estas reglas se ha basado en el uso de un fichero *.htaccess* cuya estructura se resume a continuación:

```
# Aseguramos que los ficheros rdf son devueltos con el tipo de
contenido apropiado
AddType application/rdf+xml .rdf
# Habilitamos el motor de sobreescritura
RewriteEngine On
RewriteBase /semse
# DCMI dcelements
RewriteRule ^dcelements/$ scheme/dcelements.rdf
RewriteRule ^sqdcelements/$ sqscheme/sqdcelements.rdf
RewriteRule ^dcelementsso/$ semanticontologies/dcelementsso.rdf
...
```

Ejemplo III-12: Extracto del fichero de configuración .htaccess

Como se muestra en el ejemplo, el fichero de configuración comienza estableciendo el tipo de contenido a devolver, en este caso RDF serializado como XML. A continuación se habilita el módulo de sobreescritura de URLs y se establece la URL base para las reglas de redefinición de direcciones. Finalmente, se muestran las reglas correspondientes al esquema cualificado semánticamente *sqdcelements* y a la ontología específica *dcelementsso*. A través de estas reglas se permite el acceso al contenido del esquema/ontología a partir de la URL del subdominio que corresponda, por ejemplo: <http://purl.org/semse/sqdcelements/> proporcionará acceso al contenido del esquema cualificado *sqdcelements* incluido en el fichero *sqdcelements.rdf*.

Para finalizar la publicación de esquemas semánticos, destacar que, junto con las representaciones generadas, se han incluido también los recursos necesarios para su definición, esto es: el esquema de cualificadores semánticos y la ontología SEMSE.

3.2 Selección de la ontología de referencia

Una vez publicada la representación semántica de los esquemas, el siguiente paso consiste en seleccionar una ontología de referencia que permita la desambiguación semántica de los conceptos incluidos en dichos esquemas.

Son varios los objetivos que se abordan en este apartado y que se resumen a continuación.

Primeramente, para poder realizar la selección de la ontología de referencia, es necesario establecer el conjunto de requisitos que debe satisfacer.

Partiendo de los requisitos establecidos, se llevará a cabo la selección de la ontología entre las ontologías de acceso libre existentes.

Una vez seleccionada la ontología será preciso definir un método de representación compatible con la propuesta, esto es, que permita la integración con las representaciones semánticas de los esquemas.

Finalmente, se procederá a la publicación de la ontología de referencia de forma que quede accesible e integrada con el resto de la propuesta.

Las principales tareas de esta fase se resumen en la siguiente figura:

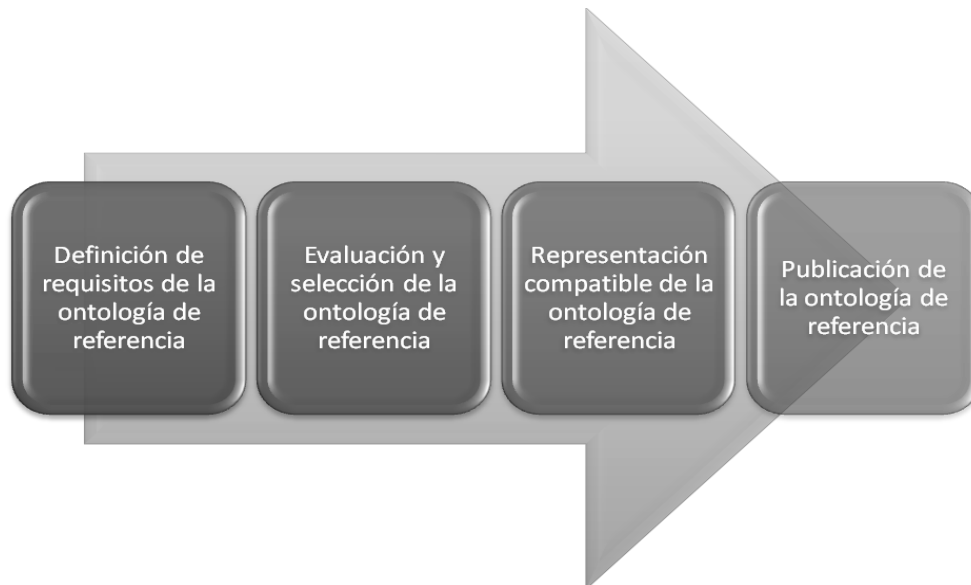


Figura III-4: Tareas de la fase de selección de la ontología de referencia

3.2.1 Definición de requisitos de la ontología de referencia

Para realizar una valoración de las ontologías es necesario establecer un conjunto de requisitos a satisfacer, sobre el cual realizar la selección. Estos requisitos se derivan de los criterios de calidad anteriormente expuestos, extensibilidad, compatibilidad y reutilización, y se resumen en:

- **Extensible/modificable:** Es importante el hecho de la ontología permita su modificación y extensión. El porqué es la necesidad futura de añadir nuevos conceptos, relaciones, axiomas, etc., para la representación de esquemas. Este requisito se basa en la suposición de que no va a ser posible encontrar una ontología de referencia que cubra la semántica de los elementos para cualquier esquema.
- **Consultable/navegable:** Es necesario que la ontología permita su acceso para realizar consultas sobre conceptos, así como la navegación a partir de las relaciones entre ellos.
- **Lenguaje:** Se ha valorado positivamente el hecho de que la ontología se encuentra en alguno de los lenguajes recomendados para la Web Semántica, principalmente OWL. No obstante, la mayoría de los ejemplos encontrados no han cumplido esta característica (aunque varios de ellos se encuentran en proceso de conversión a OWL).
- **Referenciable vía identificación formal:** Relacionado con el siguiente punto, es preciso que la ontología incluya algún modo de identificación unívoca de los conceptos. Teniendo en cuenta que el contexto de publicación de la propuesta será la Web, se han valorado positivamente aquellas ontologías que utilizan URIs como sistema de identificación formal.

- **Descargable:** En algunos casos, las ontologías encontradas permiten su descarga para su uso. De este modo se facilita su modificabilidad, aunque poseen el inconveniente de precisar publicación y mantenimiento locales.
- **Popularidad:** Se ha valorado la popularidad de las ontologías en base a los proyectos e iniciativas en los que se han utilizado.
- **Mantenimiento/Actualidad:** Se han valorado positivamente aquellas ontologías que poseen alguna entidad que actualiza y mantiene sus conceptos y relaciones, garantizando, además, la disponibilidad del recurso.
- **Eficiencia:** Se ha valorado positivamente la eficiencia del recurso, referida a tiempo de consulta e inferencia. En este caso, como es lógico, el número de conceptos y relaciones incluidos en la ontología influyen en el tiempo de respuesta del recurso.
- **Completitud:** La completitud del recurso se ha valorado positivamente aunque con menor prioridad que la extensibilidad y la eficiencia. El porqué es la necesidad de extensibilidad del recurso, mencionada anteriormente, así como la de realizar consultas en tiempos aceptables para el usuario.
- **Estructura:** Se han valorado positivamente aquellas ontologías que presentan una estructura clara y sencilla en cuanto a conceptos y relaciones básicas. Una estructura de este tipo facilita la adición de nuevos conceptos, así como los procesos de consulta e inferencia.

3.2.2 Evaluación y selección de la ontología de referencia

Una vez establecidos los criterios de evaluación, el siguiente paso es realizar una selección entre las ontologías de acceso libre disponibles actualmente.

A partir de las ontologías evaluadas, principalmente las incluidas en el apartado 2.3.4 *Ontologías de acceso libre*, ha de contemplarse la posibilidad de que existan varias ontologías que podrían servir como ontología de referencia. De este modo, se ha planteado una nueva característica a la solución, deberá permitir la utilización de varias ontologías simultáneas para la desambiguación de términos. De este modo, se pretende que la solución sea lo más extensible y abierta posible, al tiempo que se evita la dependencia con una única ontología.

Siguiendo la clasificación basada en la conceptualización del conocimiento (Heijst, G.v.; Schreiber, A.T.; Wielinga, B.J., 1997), se optó por iniciar la evaluación de ontologías comenzando por las de dominio con cobertura amplia (*broad coverage*). El por qué se debió a la posibilidad de obtener una mayor cobertura semántica con un recurso de amplio espectro, aumentando así la posibilidad de que la ontología de referencia proporcionara la semántica de un mayor número de elementos, disminuyendo así la necesidad de modificaciones y ampliaciones.

Entre las ontologías de cobertura amplia se seleccionó inicialmente WordNet por los siguientes motivos:

- Se trata de una ontología descargable, lo que proporciona gran flexibilidad en cuanto a su uso y gestión: es consultable, navegable y modificable.
- Es mantenida por reputados investigadores de la Universidad de Princeton, lo que garantiza la actualización y mantenimiento de los conceptos y relaciones que incluye.
- Se trata de una ontología popular, en el sentido de que ha sido utilizada en numerosos proyectos de investigación, así como en el desarrollo de otras ontologías.
- Posee una versión expresada en RDF/OWL, tal y como se describirá a continuación.

Una vez seleccionada una posible ontología de referencia, el siguiente paso consistió en obtener la versión más actual de la misma, expresada en un lenguaje semántico. Desgraciadamente, esto supuso un primer problema importante, la versión que desarrolla Princeton no está expresada en un lenguaje ontológico, de modo que fue necesario encontrar una versión alternativa.

Después de analizar distintas alternativas de WordNet expresado en RDF (Graves, A., 2005), (Ogbuji, U., 2001) y (SemanticWeb.org, 2002), aparecieron problemas de representación (WebKB, 2003b) que las versiones en RDF no resolvían. Se trata de problemas de la estructura y representación de WordNet, que se hacen patentes al desarrollar su representación ontológica, de forma resumida: 1) las categorías no tienen nombres intuitivos; 2) los niveles superiores están poco estructurados y no coinciden con los niveles superiores que se suelen representar en ontologías fundacionales; 3) en ocasiones se confunden las relaciones *subtipoDe* con *instanciaDe*, problema derivado de no distinguir tipos de individuos concretos; 4) finalmente, la semántica de *parte*, *miembro*, *substancia* y *objeto*, no está siempre clara, lo que lleva a la aparición de inconsistencias. Para más información (Bonilla, S., 2007).

A la luz de estos problemas, se realizó un estudio de proyectos relacionados con la representación ontológica de WordNet. Entre ellos destacaron: *knOWLer* (Université de Neuchatel, 2003), representación de WordNet 1.6 en OWL que no resolvía los problemas de representación expuestos anteriormente; *OntoWordNet* (Laboratory for Applied Ontology, 2003) representación ontológica de WordNet en OWL, resuelve varios de los problemas de representación, aunque el proyecto no parece tener continuidad actualmente; *WebKB-2* (WebKB, 2003a) ontología que, fundacionalmente, partió de WordNet y resuelve los problemas de representación, desafortunadamente el acceso no permite su descarga.

Finalmente, se encontró una propuesta del W3C para la representación de WordNet en RDF/OWL (World Wide Web Consortium, 2006c). Dicha propuesta se basaba en los proyectos anteriores y resolvía los problemas planteados en ellos, además, aunque aún no es una recomendación aprobada, cuenta con una versión descargable y el apoyo del W3C.

Llegados a este punto, se había encontrado una ontología candidata para ser utilizada como ontología de referencia. Teniendo en cuenta que la solución tiene como requisito el poder utilizar múltiples ontologías de referencia, era suficiente mostrar la solución con la versión ontológica de WordNet y establecer el método para incluir futuras ontologías. De este modo, y antes de seleccionarla definitivamente, se realizó una valoración de la cobertura semántica de la ontología WordNet como ontología de referencia. El objetivo era valorar si los significados de los elementos de los esquemas están incluidos en la ontología, antes de realizar el proceso de correspondencia.

Tomando como muestra el esquema de elementos de metadatos de Dublin Core, en el proceso de evaluación se obtuvieron los significados de los elementos, según su definición por el DCMI, y buscándose dichos significados en WordNet versión 2.0. Para cada uno de los quince elementos del esquema, se representó: el nombre del elemento según el esquema original; la definición y el comentario del elemento definidos por el DCMI; los significados en WordNet; y un análisis de la cobertura semántica del término, para establecer si el significado está correctamente representado o no. Cabe destacar que, para evitar pérdida de semántica en el proceso de traducción, se proporcionan las definiciones en el lenguaje original, es decir, en inglés:

1. **Elemento:** title
 - a) **Significado del DCMI**
 - i) Definition: A name given to the resource.
 - ii) Comment: Typically, a Title will be a name by which the resource is formally known.
 - b) **Significado(s) en WordNet**
 - i) (n) title#2 -- (the name of a work of art or literary composition etc.; "he looked for books with the word `jazz' in the title"; "he refused to give titles to his paintings"; "I can never remember movie titles")
 - c) **Cobertura:** El significado es impreciso.

2. **Elemento:** creator
 - a) **Significado del DCMI**
 - i) Definition: An entity primarily responsible for making the content of the resource.
 - ii) Comment: Examples of a Creator include a person, an organisation, or a service. Typically, the name of a Creator should be used to indicate the entity.
 - b) **Significado(s) en WordNet**
 - i) (n) creator#2 (a person who grows or makes or invents things).
 - ii) (n) writer#1, author#1 (writes (books or stories or articles or the like) professionally (for pay))
 - c) **Cobertura:** El significado es impreciso y parcial en el primer caso. En el segundo caso es preciso aunque parcial.

3. **Elemento:** subject
 - a) **Significado del DCMI**
 - i) Definition: The topic of the content of the resource.
 - ii) Comment: Typically, a Subject will be expressed as keywords, key phrases or classification codes that describe a topic of the resource. Recommended best practice is to select a value from a controlled vocabulary or formal classification scheme.
 - b) **Significado(s) en WordNet**

- i) (n) subject#1, topic#1, theme#1 (the subject matter of a conversation or discussion) "he didn't want to discuss that subject"; "it was a very sensitive topic"; "his letters were always on the theme of love".
 - ii) (n) topic#2, subject#2, issue#4, matter#3 (some situation or event that is thought about) "he kept drifting off the topic"; "he had been thinking about the subject for several years"; "it is a matter for the police".
 - iii) (n) discipline#1, subject#3, subject area#1, subject field#1, field#4, field of study#1, study#6, bailiwick#2, branch of knowledge#1 (a branch of knowledge) "in what discipline is his doctorate?"; "teachers should be well trained in their subject"; "anthropology is the study of human beings".
 - iv) (n) subject#4, content#7, depicted object#1 (something (a person or object or scene) selected by an artist or photographer for graphic representation) "a moving picture of a train is more dramatic than a still picture of the same subject".
 - v) (n) message#2, content#2, subject matter#1, substance#6 (what a communication that is about something is about).
- c) **Cobertura:** Ninguna de las definiciones se adapta exactamente al significado del elemento.

4. **Elemento:** description

a) **Significado del DCMI**

- i) Definition: An account of the content of the resource.
- ii) Comment: Description may include but is not limited to: an abstract, table of contents, reference to a graphical representation of content or a free-text account of the content.

b) **Significado(s) en WordNet**

- i) (n) description#1, verbal description#1 (a statement that represents something in words).

c) **Cobertura:** El significado es impreciso.

5. **Elemento:** publisher

a) **Significado del DCMI**

- i) Definition: An entity responsible for making the resource available.
- ii) Comment: Examples of a Publisher include a person, an organisation, or a service. Typically, the name of a Publisher should be used to indicate the entity.

b) **Significado(s) en WordNet**

- i) (n) publisher#2 (a person engaged in publishing periodicals or books or music).

c) **Cobertura:** El significado está incompleto.

6. **Elemento:** contributor

a) **Significado del DCMI**

- i) Definition: An entity responsible for making contributions to the content of the resource.
- ii) Comment: Examples of a Contributor include a person, an organisation, or a service. Typically, the name of a Contributor should be used to indicate the entity.

b) **Significado(s) en WordNet**

- i) (n) contributor#2 (a writer whose work is published in a newspaper or magazine or as part of a book)

c) **Cobertura:** El significado está incompleto.

7. **Elemento:** date

- a) **Significado del DDCMI**
 - i) Definition: A date associated with an event in the life cycle of the resource.
 - ii) Comment: Typically, Date will be associated with the creation or availability of the resource. Recommended best practice for encoding the date value is defined in a profile of ISO 8601 [182] and follows the YYYY-MM-DD format.
 - b) **Significado(s) en WordNet**
 - i) (n) date, day of the month (the specified day of the month) "what is the date today?".
 - ii) (n) date (a particular day specified as the time something happens) "the date of the election is set by law".
 - c) **Cobertura:** Ambas definiciones son imprecisas.
8. **Elemento:** type
- a) **Significado del DDCMI**
 - i) Definition: The nature or genre of the content of the resource.
 - ii) Comment: Type includes terms describing general categories, functions, genres, or aggregation levels for content. Recommended best practice is to select a value from a controlled vocabulary (for example, the DDCMI Type Vocabulary [35]). To describe the physical or digital manifestation of the resource, use the Format element.
 - b) **Significado(s) en WordNet**
 - i) (n) type (a subdivision of a particular kind of thing) "what type of sculpture do you prefer?".
 - c) **Cobertura:** La definición es imprecisa.
9. **Elemento:** format
- a) **Significado del DDCMI**
 - i) Definition: The physical or digital manifestation of the resource.
 - ii) Comment: Typically, Format may include the media-type or dimensions of the resource. Format may be used to determine the software, hardware or other equipment needed to display or operate the resource. Examples of dimensions include size and duration. Recommended best practice is to select a value from a controlled vocabulary (for example, the list of Internet Media Types [72] defining computer media formats).
 - b) **Significado(s) en WordNet**
 - i) (n) format, formatting, data format, data formatting (the organization of information according to preset specifications (usually for computer processing)).
 - c) **Cobertura:** Bastante próximo al significado.
10. **Elemento:** identifier
- a) **Significado del DDCMI**
 - i) Definition: An unambiguous reference to the resource within a given context.
 - ii) Comment: Recommended best practice is to identify the resource by means of a string or number conforming to a formal identification system. Example formal identification systems include the Uniform Resource Identifier (URI) (including the Uniform Resource Locator (URL)), the Digital Object Identifier (DOI) and the International Standard Book Number (ISBN).
 - b) **Significado(s) en WordNet**
 - i) (n) identifier (a symbol that establishes the identity of the one bearing it).
 - c) **Cobertura:** Significado más restringido, no tiene en cuenta el contexto.

11. **Elemento:** source

a) **Significado del DDCMI**

- i) Definition: A reference to a resource from which the present resource is derived.
- ii) Comment: The present resource may be derived from the Source resource in whole or in part. Recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system.

b) **Significado(s) en WordNet**

- i) (n) reference, source (a publication (or a passage from a publication) that is referred to) "he carried an armful of references back to his desk"; "he spent hours looking for the source of that quotation".
- ii) (n) source (a document (or organization) from which information is obtained) "the reporter had two sources for the story".
- iii) (n) source, seed, germ (anything that provides inspiration for later work).

c) **Cobertura:** Bastante próximo al significado.

12. **Elemento:** language

a) **Significado del DDCMI**

- i) Definition: A language of the intellectual content of the resource.
- ii) Comment: Recommended best practice is to use RFC 3066 [4], which, in conjunction with ISO 639-2 [96], defines two- and three-letter primary language tags with optional subtags. Examples include "en" or "eng" for English, "akk" for Akkadian, and "en-GB" for English used in the United Kingdom.

b) **Significado(s) en WordNet**

- i) (n) language, linguistic communication (a systematic means of communicating by the use of sounds or conventional symbols) "he taught foreign languages"; "the language introduced is standard throughout the text"; "the speed with which a program can be executed depends on the language in which it is written".

c) **Cobertura:** El significado es impreciso.

13. **Elemento:** relation

a) **Significado del DDCMI**

- i) Definition: A reference to a related resource.
- ii) Comment: Recommended best practice is to reference the resource by means of a string or number conforming to a formal identification system.

b) **Significado(s) en WordNet**

- i) (n) relation (an abstraction belonging to or characteristic of two entities or parts together)

c) **Cobertura:** Significado más preciso.

14. **Elemento:** coverage

a) **Significado del DDCMI**

- i) Definition: The extent or scope of the content of the resource.
- ii) Comment: Coverage will typically include spatial location (a place name or geographic coordinates), temporal period (a period label, date, or date range) or jurisdiction (such as a named administrative entity). Recommended best practice is to select a value from a controlled vocabulary (for example, the Thesaurus of Geographic Names [54]) and that, where appropriate, named places or time periods be used in preference to numeric identifiers such as sets of coordinates or date ranges.

b) **Significado(s) en WordNet**

- i) (n) coverage (the extent to which something is covered) "the dictionary's coverage of standard English is excellent".

c) **Cobertura:** Bastante próximo al significado.

15. **Elemento:** rights

a) **Significado del DCMI**

- i) Definition: Information about rights held in and over the resource.
- ii) Comment: Typically, a Rights element will contain a rights management statement for the resource, or reference a service providing such information. Rights information often encompasses Intellectual Property Rights (IPR), Copyright, and various Property Rights. If the Rights element is absent, no assumptions can be made about the status of these and other rights with respect to the resource.

b) **Significado(s) en WordNet**

- i) (n) right ((frequently plural) the interest possessed by law or custom in some intangible thing) "mineral rights"; "film rights".

c) **Cobertura:** Bastante próximo al significado.

A partir de esta evaluación se puede comprobar cómo, en la mayoría de los casos, las definiciones proporcionadas por WordNet son imprecisas o están incompletas. La falta de cobertura del esquema seleccionado por WordNet plantea un problema, ante el que se plantean dos soluciones:

- La primera solución consistiría en precisar los significados para mejorar la cobertura semántica del esquema. El modo de realizarlo sería mediante la inclusión de nuevos significados, *synsets*, más precisos y próximos a los elementos de Dublin Core. Esta solución es perfectamente viable puesto que se cuenta con la versión descargada de la ontología, la principal pega que presenta es la gestión de dichos cambios. Por un lado, realizar los cambios en una versión *local* de la ontología implicaría volver a aplicar los cambios en la versión de WordNet disponible en el momento de publicar y compartir la ontología. En otras palabras, puesto que el mantenimiento de WordNet, realizado por Princeton, no tendría en cuenta las modificaciones realizadas, los nuevos significados deberían volverse a añadir en futuras versiones. Por otro lado, e igualmente importante, la versión de WordNet de Princeton no es una ontología como tal ni está expresada en un lenguaje semántico, como OWL. De hecho, la versión evaluada es la propuesta realizada por el W3C (World Wide Web Consortium, 2006c), por tanto, sería preciso que las actualizaciones de WordNet fueran adaptadas del mismo modo, antes de poder ser utilizadas como ontología de referencia.
- La segunda solución pasaría por utilizar, como ontología de referencia, un recurso de cobertura amplia ya compartido en Web, que garantice que los cambios realizados van a mantenerse sin interferir en el proceso de mantenimiento de la ontología. En otras palabras, sería preciso encontrar un recurso semántico en Web que garantice una versión actualizada de la ontología, que permita la inclusión de nuevos significados y conceptos, y que comparta dichos cambios, preservándolos en el tiempo. Una vez desarrollado el sistema informático de gestión de esquemas, éste podrá hacer uso del recurso semántico compartido para la desambiguación de términos.

Por los motivos expuestos, se decidió aplicar la segunda opción, descartando el uso de WordNet como ontología de referencia. De este modo se inició la evaluación de ontologías web que permitieran compartir las actualizaciones sobre la ontología original, aunque es necesario precisar que no se descarta la creación y gestión de ontologías locales, como nivel intermedio de abstracción en la representación de esquemas y dominios más concretos.

Siguiendo este nuevo enfoque, la evaluación de posibles ontologías de referencia llevó a seleccionar como candidata a WebKB-2 (WebKB, 2003a). Los motivos de esta elección son: se trata de una ontología compartida en Web que permite la consulta y actualización de conceptos; en sus primeras versiones incorporó WordNet, además de otras ontologías fundacionales como *SUMO* (Suggested Upper Merged Ontology, 2010) o *DOLCE* (Laboratory for Applied Ontology, 2006); resuelve los problemas de la representación ontológica de WordNet; y permite la adición de nuevos conceptos y relaciones, compartidos en Web, preservándolos en futuras actualizaciones de la ontología.

Al realizar la evaluación de la cobertura semántica del esquema por esta ontología, se comprobó cómo una de las fuentes incluidas en ella había sido el esquema de Dublin Core. Si bien las definiciones han sido modificadas, los términos son cubiertos de forma precisa, lo que responde perfectamente a las necesidades actuales de la propuesta para dicho esquema. A continuación se muestra un ejemplo de la cobertura de WebKB-2:

1. **Elemento:** title
 - a) **Significado del DDCMI**
 - i) Definition: A name given to the resource.
 - ii) Comment: Typically, a Title will be a name by which the resource is formally known.
 - b) **Significado(s) en WebKB-2**
 - i) tap#title (? -> entity) to specify a name of a resource. Supertype: name. Equal: Title.
 - ii) dc#Title (? -> entity) to specify a name of a resource. Supertype: name. Equal: title.
 - iii) pm#title (? -> string) for connecting an object to its "title" in a natural language. Supertype: Description.
 - iv) #title the name of a work of art or literary composition etc.; "he looked for books with the word `jazz' in the title"; "he refused to give titles to his paintings"; "I can never remember movie titles". Supertype: name. Subtype: rubric.heading
 - c) **Cobertura:** El término es cubierto por la primera y segunda definición.

Como se puede comprobar en el ejemplo, WebKB-2 incluye un acrónimo, como prefijo en los identificadores de los elementos, que representa la fuente de la cual proviene el término. De este modo, podemos comprobar cómo aparece *dc#Title* entre los significados. Aunque no es exactamente el propuesto por el DDCMI, se trata de un significado equivalente contra el que puede desambiguarse el término. Destacar que, en último lugar y sin prefijo, se incluye el significado de WordNet, claramente incompleto.

Desafortunadamente durante la evaluación de WebKB-2 también se observaron múltiples problemas. Primeramente, la cobertura para el esquema de muestra es completa, debido a que se tomó como fuente para la generación del recurso, no ocurre

así para el resto de esquemas evaluados. Las modificaciones realizadas sobre la ontología, nuevos conceptos y relaciones, son compartidas vía Web pero no con otros usuarios, esto es, son privadas para el usuario que las realiza; además, deben ser validados por los responsables del recurso. El lenguaje en el que está expresada la ontología y el modo de publicación en Web hacen complejo el uso de posibles ontologías locales e impiden su descarga.

Los problemas expuestos llevaron a descartar WebKB-2 como ontología de referencia y, siguiendo en la línea de ontologías de cobertura amplia, a evaluar OpenCyc (Cycorp, 2009a) como ontología candidata.

La evaluación de OpenCyc arrojó los resultados que se resumen a continuación. Aún siendo una de las ontologías más completas entre las existentes, la cobertura semántica de los elementos evaluados es, en ocasiones, parcial o inexistente. A esto debemos añadir la complejidad de la estructura a alto nivel que presenta la ontología, haciendo, en ocasiones, complejo el encontrar y/o añadir nuevos conceptos. El planteamiento de la ontología, en cuanto a uso y mantenimiento, se basa en la instalación de un servidor en el que compartir y gestionar la ontología. Esto no implicaría un problema, de no ser por el lenguaje en el cual está expresada la ontología: *CycL*. *CycL* es el lenguaje en el que se define OpenCyc, se basa en lógica de predicados y proporcionan un buen rendimiento en inferencia y consultas. Desafortunadamente, no es un lenguaje adecuado para la publicación web de una ontología y dificulta la interoperatividad con ontologías en dicho entorno. Para salvar esta carencia, fundamental, existe una versión de OpenCyc expresada en OWL que sí permitiría interoperar con otras ontologías web. No obstante, la versión OWL presenta dos problemas importantes (Soto, J.; García, E.; Sanchez, S., 2006): por un lado no proporciona la ontología completa, carece de muchos términos y no incluye aserciones; por otro lado, el rendimiento de esta versión es muy inferior al obtenido con la versión nativa de OpenCyc. En resumen, implica la instalación de un servidor con altas prestaciones para obtener una implementación parcial de la ontología. Los problemas expuestos llevaron a descartar OpenCyc como posible ontología de referencia.

Antes de evaluar otra ontología de cobertura amplia, se evaluaron los problemas comunes a este tipo de ontologías. Se trata de recursos de amplio espectro, en cuanto a cobertura semántica, pero que van a necesitar ser ampliados en cualquier caso. Esto hace de la extensibilidad una característica fundamental en el proceso de selección. Por otro lado, el entorno de publicación de la solución y la necesidad de integración con el resto de la propuesta hacen necesario que el recurso sea interoperable con otras ontologías en Web. Los lenguajes en los que se expresan este tipo de ontologías no suelen estar pensados para el entorno Web ni para facilitar la interoperatividad con otras ontologías y, en el caso de encontrarse una adaptación para dicho entorno, suelen ser versiones incompletas o desactualizadas que, además, penalizan el rendimiento del recurso.

Los problemas identificados en la evaluación de ontologías de cobertura amplia provocaron un cambio en el enfoque del proceso de selección. Concretamente, se pasó a evaluar ontologías genéricas por los siguientes motivos: las ontologías de aplicación son más específicas pero no cubren un espectro lo suficientemente amplio como para dar cobertura semántica a la propuesta; la extensibilidad es necesaria y este tipo de ontologías están diseñadas para ello, puesto que sólo proporcionan cobertura para los

niveles superiores del conocimiento; cabe la posibilidad de que la extensión sea un proceso más laborioso pero, por el contrario, tiene a favor el aspecto del rendimiento, dado que el recurso va a extenderse y especializarse, en sucesivas iteraciones, pero no va a incluir miles de conceptos independientes semánticamente; por último, en cuanto a la interoperatividad, es ya una característica incluida en el diseño de estos recursos fundacionales a la que, en el contexto de la propuesta, sólo es preciso añadir que el lenguaje permita la publicación Web.

Así, se inició de nuevo la evaluación, comenzando por ontologías ligeras para entornos Web, entre las que destacaron SIOC (Digital Digital Enterprise Research Institute, 2010) y UMBEL (UMBEL.org, 2009). La primera de ellas fue descartada al tener como ámbito específico la integración de información de comunidades online (foros, blogs, etc.). La segunda es, aparentemente, una solución compatible con la propuesta. Se trata de una ontología ligera diseñada para relacionar ontologías en Web. Está diseñada para facilitar la interoperatividad y definida en OWL-Full, al tiempo que proporciona un buen rendimiento en consultas e inferencia, de hecho, ha servido como base para el planteamiento de la ontología SEMSE.

Desafortunadamente, UMBEL también presentó problemas que provocaron su descarte. El primero de ellos es la extensibilidad del recurso, los conceptos incluidos en UMBEL provienen de un proceso de filtrado aplicado a la ontología OpenCyc, de este modo, la inclusión de conceptos no incluidos en OpenCyc y no obtenidos por el proceso de filtrado atentaría contra los fundamentos de UMBEL. Además, como consecuencia de tener OpenCyc como base, UMBEL hereda la complejidad en los niveles superiores de su jerarquía de conceptos, lo que dificulta la consulta y adición de nuevos conceptos. En resumen, se trata de una ontología diseñada con un propósito distinto al necesario en la presente propuesta.

La evaluación prosiguió en el ámbito de las ontologías fundacionales entre las que destacaron Dolce (Laboratory for Applied Ontology, 2006), SUMO (Suggested Upper Merged Ontology, 2010) y PROTON (Terziev, I.; Kiryakov, A.; Manov, D., 2005). Todas ellas cubren un espectro semántico amplio, son extensibles y estructuran de forma simple los niveles superiores del conocimiento. Las dos primeras tienen como inconveniente estar definidas en el lenguaje KIF¹⁷, pero PROTON utiliza OWL Lite como lenguaje de definición de conceptos y relaciones.

Tras una evaluación en profundidad de PROTON se obtuvieron las siguientes conclusiones.

- Posee una estructura jerárquica, modular y sencilla, con amplia cobertura semántica, tomada parcialmente de Dolce.
- Está diseñada para ser extendida de modo que, aunque no cubre dominios específicos, sí proporciona la estructura que facilita su inclusión.

¹⁷ En (Laboratory for Applied Ontology, 2006) y en (Suggested Upper Merged Ontology, 2010) pueden encontrarse versiones en OWL de Dolce y SUMO, respectivamente. Desafortunadamente, se trata de versiones provisionales e incompletas de estas ontologías.

- Se define utilizando OWL, lo que facilita su publicación y la integración con ontologías publicadas un entorno Web.
- Otras de sus características son: inclusión de definiciones ligeras de conceptos y relaciones, alineamiento con estándares y documentación completa.

Como último paso en la evaluación, se realizó una evaluación de la cobertura semántica del recurso, con el fin de comprobar la cobertura de los significados de los conceptos incluidos en los esquemas seleccionados. Se pudo comprobar cómo el esquema de muestra, los elementos de metadatos de Dublin Core, estaba prácticamente alineado con PROTON. Del mismo modo, una segunda valoración con FOAF mostró también una cobertura parcial. En ambos casos, la inclusión de la semántica necesaria era viable, gracias a las posibilidades de extensión que brinda la ontología.

Con esto concluyó la fase de selección y evaluación tomándose, finalmente, PROTON como ontología de referencia.

3.2.3 Representación compatible de la ontología de referencia

Una vez seleccionada la ontología de referencia, y habiendo comprobado que cubre las necesidades de la propuesta, el objetivo de este apartado es desarrollar la representación de dicha ontología de forma que sea compatible e integrable con el resto de artefactos desarrollados en la propuesta.

Más concretamente, tal y como se definían en el apartado 3.2.1 *Definición de requisitos de la ontología de referencia*, la representación debe cumplir un conjunto de requisitos, a saber:

- Debe permitir la extensión y la modificación de la ontología de referencia. Como se ha evidenciado a través del proceso de selección y evaluación, va a ser preciso incluir nuevos términos y precisar algunos de los existentes en la ontología. Con este fin, la solución más recomendable pasaría por mantener la ontología de referencia original como componente independiente en el proceso de extensión, de modo que las actualizaciones no precisen la modificación de la ontología de referencia original.
- Debe permitir la consulta de los conceptos incluidos en la ontología así como de los añadidos durante el proceso de especialización/extensión. Asimismo debe mantener las relaciones entre conceptos que permitan la navegación entre ellos, tanto para los pertenecientes a la ontología de referencia original como para los añadidos/especializados con posterioridad.
- Debe estar expresada en un lenguaje que permita la publicación: compartición, acceso y relación, en un entorno Web. Dicho lenguaje, además, deberá soportar la representación de semántica de conceptos y relaciones.
- Debe hacer uso de un sistema de identificación formal para la identificación unívoca de los conceptos incluidos. En relación con el punto anterior, el sistema de identificación formal deberá estar, preferiblemente, basado en URIs.

- Debe permitir su descarga. También en relación con la publicación, debe permitirse el acceso y descarga de la ontología de referencia así como de sus extensiones.
- Debe facilitar el mantenimiento y la actualización de la ontología de referencia original. Previendo futuras actualizaciones en la ontología de referencia, por parte de la entidad responsable de su desarrollo, la representación debe permitir dicho proceso y minimizar el impacto que dichos cambios puedan producir. La independencia de la ontología original de los componentes de extensión permitiría, ante una actualización del recurso original, realizar el análisis y posibles modificaciones únicamente en dichos componentes de extensión, siendo la ontología de referencia original independiente en dicho proceso.
- Debe garantizar la eficiencia del recurso. Logrado en parte por el tipo de recurso seleccionado, la representación debe permitir la inclusión incremental de aquellos términos que sean pertinentes en el proceso de extensión, evitando la necesidad de incluir términos espúreos o semánticamente no relacionados.
- Debe facilitar la completitud del recurso. Aunque esta característica ha sido tratada con menor prioridad en pro de la extensibilidad y la eficiencia, la representación debe facilitar la completitud a través de la facilidad de extensión y el ámbito de la ontología seleccionada.
- Debe proporcionar una estructura clara y compatible con la de la ontología de referencia original. La representación debe mantener la sencillez de la estructura de la ontología de referencia y ser compatible con ella, teniendo en cuenta que podrían utilizarse múltiples ontologías de referencia, además de la seleccionada.

A partir de estos requisitos, se propone como representación la definición de una ontología de aplicación que, a través de mecanismos de importación, relacione las extensiones con la ontología de referencia original. De este modo, la ontología de aplicación dependería de la ontología de referencia pero mantendría como elemento independiente a la ontología original.

Más concretamente, la representación implica la definición de una ontología de aplicación para cada ontología de referencia que se desee utilizar. Dentro de la ontología de aplicación se incluirán los conceptos y propiedades, no incluidos en la ontología de referencia original, así como las modificaciones sobre los elementos originales, es decir, las extensiones de la ontología original. El lenguaje de definición de la ontología de aplicación será OWL, habilitándose la inclusión de relaciones con los conceptos y propiedades de la ontología original, así como la publicación Web del recurso. La definición de conceptos y propiedades se realizará haciendo uso de URIs como sistema de identificación formal, lo que facilitará su consulta y acceso, una vez publicado el recurso. A continuación se muestra un extracto de la ontología de aplicación correspondiente a la ontología de referencia PROTON.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE owl [
  <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#' >
  <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#' >
  <!ENTITY owl 'http://www.w3.org/2002/07/owl#' >
```

```

<!ENTITY xsd 'http://www.w3.org/2001/XMLSchema#' >
<!ENTITY psys 'http://proton.semanticweb.org/2005/04/protons#'>
<!ENTITY ptop 'http://proton.semanticweb.org/2005/04/protont#'>
<!ENTITY pupp 'http://proton.semanticweb.org/2005/04/protonu#'>
<!ENTITY papp 'http://purl.org/semse/proton/protona#'>
]>
<rdf:RDF
  xmlns:owl="&owl;"
  xmlns:rdf="&rdf;"
  xmlns:rdfs="&rdfs;"
  xmlns:psys="&psys;"
  xmlns:ptop="&ptop;"
  xmlns:pupp="&pupp;"
  xmlns="&papp;"
  xml:base="&papp;"
>
<owl:Ontology rdf:about="">
  <rdfs:comment>PROTON (Proto Ontology), Application Ontology</rdfs:comment>
  <owl:imports rdf:resource="http://proton.semanticweb.org/2005/04/protons"/>
  <owl:imports rdf:resource="http://proton.semanticweb.org/2005/04/protont"/>
  <owl:imports rdf:resource="http://proton.semanticweb.org/2005/04/protonu"/>
  <owl:versionInfo>"0.1"</owl:versionInfo>
</owl:Ontology>

<!-- DCMI dcelements schema added concepts -->
<owl:ObjectProperty rdf:about="#hasRelation"
  rdfs:label="hasRelation">
  <rdfs:comment>A related resource. Recommended best practice is to identify
the related resource by means of a string conforming to a formal
identification system.</rdfs:comment>
  <rdfs:range rdf:resource="&ptop;InformationResource"/>
  <rdfs:domain rdf:resource="&ptop;InformationResource"/>
</owl:ObjectProperty>

<!-- FOAF schema added concepts -->
<!-- Classes -->
<owl:Class rdf:about="#Image"
  rdfs:label="Image">
  <rdfs:comment>Class corresponding to those documents which are images.
Digital images (such as JPEG, PNG, GIF bitmaps, SVG diagrams etc.) are
examples of Image.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="&ptop;InformationResource"/>
</owl:Class>
...

```

Ejemplo III-13: Extracto de codificación de la ontología de aplicación correspondiente a la ontología de referencia PROTON

En el ejemplo anterior se ha incluido el encabezado de la ontología así como un ejemplo de propiedad y otro de clase no incluidos en la ontología de referencia original. La codificación comienza definiendo los espacios de nombres necesarios para la definición de clases y propiedades. A continuación se define la ontología importando los módulos en los que se descompone la ontología PROTON, de este modo, al cargar la ontología de aplicación, se incluirán automáticamente los conceptos y relaciones definidos en la ontología de referencia. Seguidamente se incluye la propiedad “*hasRelation*” proveniente del esquema de metadatos de Dublin Core y no cubierta semánticamente por la ontología original. Como se puede observar, se define la URI de la propiedad, su etiqueta y su significado. Tanto el rango como el dominio de la propiedad se definen sobre conceptos incluidos en la ontología de referencia o, en su caso, sobre conceptos incluidos en la propia ontología de aplicación. De este modo la nueva propiedad queda

integrada en el contexto definido por la ontología de aplicación, como extensión de la ontología original. Finalizando el ejemplo, se incluye la definición de la clase “*Image*” proveniente del esquema FOAF y no cubierta semánticamente por la ontología de referencia. De nuevo se incluye su URI, su etiqueta y su definición. La integración dentro del contexto de la ontología de referencia se realiza incorporando el concepto a la estructura de la ontología de aplicación, en este caso, definiendo la nueva clase como subclase de “*InformationResource*”. Cabe destacar que el proceso de extensión será abordado durante el proceso de correspondencia, incluido en apartados posteriores.

Una vez planteada la representación de la ontología de referencia, se pueden extraer las siguientes conclusiones respecto a los criterios de calidad:

- *Extensibilidad*: La representación propuesta se ha diseñado siguiendo este criterio, en cuanto a que mantiene la independencia de la ontología de referencia original, permitiendo así la inclusión de nuevos conceptos y propiedades, sin necesidad de modificar la ontología original. Además, al utilizar un sistema de identificación formal, permite la gestión de conceptos y relaciones de forma individual al resto de elementos definidos en la ontología de referencia. Finalmente, permite el uso de múltiples recursos ontológicos al definir una ontología de aplicación por cada ontología de referencia, restringiendo así el ámbito de gestión de los conceptos y propiedades a la ontología correspondiente.
- *Compatibilidad*: La solución propuesta se adecúa a este criterio, en cuanto a que se ha diseñado para garantizar la independencia de la versión original de la ontología, pero manteniendo la compatibilidad con su estructura. De hecho, la ontología de aplicación incluye los conceptos y relaciones en la jerarquía de la ontología original, logrando que el proceso de gestión y consulta sea homogéneo en el ámbito de la ontología de referencia (la ontología original extendida). El lenguaje utilizado en la definición juega un papel importante en este aspecto, permitiendo la definición de relaciones semánticas con la estructura de la ontología original. Al mismo tiempo, el lenguaje y el sistema de identificación formal utilizados van a facilitar la integración de la ontología de referencia con el resto de artefactos definidos en la propuesta. Aunque será tratado posteriormente, la integración de los artefactos se va a restringir al problema de relacionar dos ontologías publicadas en Web y definidas en OWL: una ontología específica frente a una ontología de referencia.
- *Reutilización*: Se ha atendido a este criterio en cuanto a que la ontología de referencia es compartida por todas las representaciones semánticas de los esquemas. Del mismo modo, también comparten los conceptos y relaciones incluidos en la ontología de aplicación como resultado de la extensión de la ontología de referencia original. En esta línea también se encuentra el hecho de que la ontología va a ser publicada y, por tanto, va a estar accesible para su consulta, descarga y reutilización; aspecto apoyado también por el lenguaje de definición y el sistema de identificación formal utilizados, que facilitan la publicación e identificación de conceptos y relaciones de forma unívoca, para su posterior reutilización.

Definido el modo de representación de las ontologías de referencia, sólo resta establecer los mecanismos de publicación de los artefactos propuestos. Este aspecto se aborda en el siguiente apartado.

3.2.4 Publicación de la ontología de referencia

Una vez definido el modo de representación de las ontologías de referencia, se ha procedido a su publicación en Web. El objetivo de la publicación es poner a disposición de los usuarios las representaciones generadas, permitiendo así su acceso, uso y reutilización, directamente desde un navegador. Además, en combinación con la plataforma informática de soporte, el usuario podrá consultar y utilizar las representaciones de forma amigable, a través de una aplicación Web.

Antes de la publicación de las representaciones propuestas, ha sido necesario tomar una serie de decisiones de diseño que se resumen a continuación:

- Definición de la política de nombrado: Las ontologías de aplicación recibirán el nombre de la ontología de referencia que extienden, seguido del acrónimo ‘a’ (*application*), por ejemplo: *protona* será la ontología de aplicación correspondiente a la ontología de referencia PROTON.
- Definición del método de publicación: Tal y como se indicaba en el apartado 3.1.4 *Publicación de esquemas semánticos*, la ontología de aplicación se ha publicado utilizando como base la URI persistente <http://purl.org/semse>.
- Definición de los espacios de nombres: A partir de las recomendaciones del *World Wide Web Consortium* para la publicación de vocabularios RDF (Consortium, W.W., 2008) y por compatibilidad con las ontologías de referencia, que habitualmente utilizan este esquema, se decidió utilizar un esquema basado en *hash* para el acceso a las representaciones y su contenido. Respecto a la estructura, partiendo de la URL base, se han definido subdominios para cada una de las ontologías de referencia, dentro de los cuales, se ubican la ontología de aplicación correspondiente. De este modo, cada representación tiene un subdominio asignado, definido por el nombre asignado anteriormente, que permite el acceso unívoco a su contenido, por ejemplo: <http://purl.org/semse/proton/protona#> es el subdominio asignado a la ontología de aplicación correspondiente a la ontología PROTON.

Dentro del servidor desplegado para la publicación de recursos, se ha generado una estructura de directorios para albergar las ontologías de aplicación (directorio *applicationontologies*). Una vez ubicados los recursos en el servidor, el siguiente paso ha sido crear las reglas de acceso a dichos recursos, en función de la estructura de directorios definida y de la definición de los espacios de nombres. El método para incluir estas reglas se ha basado en el uso del fichero *.htaccess* definido en el apartado 3.1.4 *Publicación de esquemas semánticos* cuya estructura, relativa a la publicación de ontologías de aplicación, se resume a continuación:

```
...  
# Reglas de sobrescritura para devolver ontologías de aplicación  
RewriteRule ^proton/protona$ applicationontologies/protona.rdf  
...
```

Ejemplo III-14: Extracto del fichero de configuración .htaccess

Como se muestra en el ejemplo, el fichero de configuración añade una regla por cada ontología de aplicación a compartir, en este caso *protona*. A través de estas reglas se permite el acceso al contenido de la ontología de aplicación a partir de la URL del subdominio que corresponda, por ejemplo: <http://purl.org/semse/proton/protona#> proporcionará acceso al contenido de la ontología de aplicación *protona*, incluida en el fichero *protona.rdf*.

3.3 Correspondencia entre esquemas y la ontología de referencia

Una vez desarrolladas las representaciones semánticas de los esquemas y seleccionada la ontología de referencia, se llevará a cabo la interrelación de los conceptos incluidos en los esquemas semánticos con los incluidos en la ontología de referencia. Para ello, son varios los objetivos que se abordan en este apartado y que se resumen a continuación, a modo de introducción.

En primer lugar es necesario establecer el modo de interrelacionar las representaciones semánticas de los esquemas con la ontología de referencia. Para ello deberán analizarse los métodos existentes y seleccionar aquel que mejor se adecúe a la propuesta. A continuación se propondrá el modo de articularlo, cumpliendo unos criterios de calidad.

Una vez establecido el método, se llevará a cabo el proceso de correspondencia. Con este fin, será necesario definir una metodología para, a continuación, aplicarla a las representaciones semánticas de esquemas generadas y a la ontología de referencia seleccionada. Durante el proceso se analizarán y resolverán los problemas encontrados.

Finalmente, se procederá a la publicación de los artefactos generados durante el proceso de correspondencia para su uso y reutilización, tanto por usuarios como por la plataforma informática de soporte.

En la siguiente figura se resumen las tareas incluidas en esta fase:



Figura III-5: Tareas de la fase de correspondencia entre esquemas y la ontología de referencia

3.3.1 Evaluación y selección del método de correspondencia

El método de correspondencia integra los artefactos desarrollados en la propuesta, relacionando semánticamente los conceptos de las representaciones semánticas de esquemas con los de la ontología de referencia. De este modo, permite la interoperatividad entre conceptos de distintos esquemas, estableciendo una semántica común entre ellos. Se logra así una sinonimia entre elementos de distintos esquemas, manteniendo las definiciones concretas especificadas en el contexto del esquema.

Es preciso definir un método de correspondencia que, además de integrarse en la propuesta, cumpla los siguientes requisitos:

- **Extensible/modificable:** Es importante el hecho de que el método de correspondencia permita su modificación y extensión. El porqué es la necesidad futura de interrelacionar nuevos conceptos y propiedades en el proceso de incorporación de nuevos esquemas semánticos. Del mismo modo debe ser posible la definición de nuevas relaciones o especialización de las existentes, de cara a interrelacionar conceptos.
- **Consultable/navegable:** Es necesario que el método de correspondencia genere artefactos que puedan ser utilizados posteriormente en el proceso de consulta e inferencia, estableciendo relaciones navegables entre los recursos interrelacionados.
- **Lenguaje:** Es preciso que el lenguaje utilizado para la definición de los artefactos, generados por el método de correspondencia, permita la incorporación de relaciones semánticas y facilite su publicación en un entorno Web.
- **Referenciable vía identificación formal:** Es preciso que el método de correspondencia sea compatible con los sistemas de identificación formal

utilizados en las ontologías específicas y en la ontología de referencia, con el fin de poder establecer correspondencias entre conceptos de ambas fuentes. Teniendo en cuenta que el contexto de publicación de la propuesta será la Web, el sistema de identificación formal más adecuado es el basado en URIs.

- **Accesible/Descargable:** El método de correspondencia debe generar artefactos susceptibles de ser publicados en un entorno Web. El objetivo es que, de esta forma, queden disponibles para su consulta y/o descarga por parte de usuarios y aplicaciones.
- **Independencia:** Motivado en parte por la necesidad de extensibilidad del método de correspondencia, es necesario que éste reduzca en lo posible las dependencias entre los artefactos desarrollados en la propuesta. Reduciendo el acoplamiento entre los distintos componentes (ontologías específicas, ontología de referencia y artefactos de correspondencia) se facilita la modificabilidad de la solución. De este modo el método debería evitar que los componentes desarrollados, tanto las ontologías específicas como la ontología genérica, se vean implicados en el proceso de correspondencia.
- **Eficiencia:** El método de correspondencia debería permitir la gestión de relaciones con un grano fino. Esto es, la posibilidad de gestionar relaciones entre conceptos y relaciones a nivel unitario, evitando la necesidad de incorporar correspondencias espúreas o semánticamente independientes.
- **Completitud:** El método de correspondencia deberá permitir la interrelación de cualquier elemento, independientemente de su tipo concreto, entre las ontologías específicas y la ontología de referencia.
- **Estructura:** El método de correspondencia deberá establecer, en los artefactos que defina, una estructura sencilla y que facilite su mantenibilidad. De este modo se facilitarían, además, los procesos de consulta e inferencia.

Una vez evaluados distintos métodos y atendiendo a los requisitos expuestos anteriormente, se ha definido un método de correspondencia que, siguiendo la clasificación expuesta en el apartado 2.3.6 *Métodos de correspondencia entre ontologías*, posee las siguientes características:

- La información de entrada a procesar será la relativa a la definición del esquema. La información referente a las instancias será recuperada posteriormente, a través de un proceso de rastreo web.
- El alineamiento se realizará de forma manual. Se pretende, de este modo, obtener la máxima calidad y fiabilidad en las correspondencias generadas.
- No se devolverán correspondencias incorrectas. Relacionado con el punto anterior, se asume que todas las correspondencias de los alineamientos son correctas.
- El conjunto de correspondencias devuelto como resultado de un proceso de

consulta será completo, esto es, se devolverán todas las correspondencias.

- La correspondencia entre esquemas y la ontología de referencia se realizará en tiempo de diseño, debido al carácter del sistema y a la posibilidad de obtener el alineamiento previo a la ejecución del sistema.
- La función de alineamiento hará uso de un algoritmo híbrido que incluirá técnicas terminológicas, estructurales y semánticas. La técnica terminológica se basará en la comparación de las etiquetas de los elementos del esquema y la ontología de referencia, lo que proporcionará un conjunto de correspondencias tentativas. En paralelo se hará uso de la estructura proporcionada por la ontología de referencia con el fin de obtener nuevas correspondencias y refinar las anteriores. Finalmente, las correspondencias serán validadas o descartadas mediante la semántica de los conceptos, definida en el esquema y en la ontología de referencia.

Respecto al ciclo de vida de los alineamientos, su gestión se puede resumir en las siguientes fases:

- Creación, evaluación y mejora: Se realizará de forma manual, siguiendo la metodología incluida en el apartado 3.3.2 *Correspondencia de conceptos entre esquemas y la ontología de referencia*.
- Comunicación: Incluirá el almacenamiento y compartición del alineamiento, tal y como se describe en el apartado 3.3.3 *Publicación de la ontología de correspondencia*.
- Explotación: A través del sistema informático de soporte y mediante un proceso de mezcla, se generará una ontología global que permita la recuperación de conceptos relacionados, utilizando la ontología de referencia como vocabulario común. En el apartado 3.4 *Validación y verificación de la propuesta* se ilustrará el proceso de consulta sobre la ontología global generada.

Una vez seleccionado el método de correspondencia, y en base a los requisitos definidos, se propone una representación articulada mediante un recurso independiente formalizado mediante una ontología de aplicación y definido en lenguaje OWL. Este recurso, u ontología de alineamiento, permitirá definir las relaciones entre los conceptos de las ontologías específicas y los de la ontología de referencia. Se logra así que el recurso sea: extensible, permitiendo añadir futuras interrelaciones; consultable, ya que puede ser gestionado como una ontología más expresada en OWL; referenciable, puesto soporta URIs como sistema de identificación formal; accesible, en cuanto a que permite su publicación en Web; garantiza la independencia de los recursos que interrelaciona, de modo que un cambio en la interrelación de dos conceptos no implica la modificación de los recursos relacionados; eficiente, en cuanto que permite la gestión individual de cada interrelación; facilita la relación de cualquier elemento independientemente de su tipo concreto, dado que admite distintos tipos de relación (clase-clase, propiedad-propiedad, propiedad-clase, clase-propiedad); finalmente, proporciona una estructura simple de alineamiento que facilita su mantenimiento y consulta.

Inspirado en el recurso de correspondencia incluido en la ontología UMBEL, se propone una ontología de alineamiento que consistirá en un conjunto de reglas en las que se establecen las relaciones entre conceptos de las ontologías específicas y de referencia. Partiendo de la posibilidad de utilizar múltiples ontologías de referencia, se definirá una ontología de alineamiento por cada ontología de referencia. Más concretamente, para cada ontología de referencia se definirá una ontología para albergar el alineamiento “interno” (con conceptos de las ontologías específicas) y otra para albergar el alineamiento “externo” (con conceptos de ontologías externas al sistema). Aunque en la presente propuesta no se han definido relaciones con ontologías externas, se ha definido el recurso de cara a cubrir la necesidad de ampliar la ontología de referencia, incluyendo relaciones entre ésta y recursos externos, ampliándose de este modo su contexto. A continuación se muestra un extracto de la ontología de alineamiento interno correspondiente a la ontología de referencia PROTON.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF>
<rdf:RDF
  xmlns="http://purl.org/semse/"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  ...
<rdf:Description rdf:about="http://purl.org/semse/dcelementsso/sp/title">
  <owl:equivalentProperty
    rdf:resource="http://proton.semanticweb.org/2005/04/protont#title"/>
</rdf:Description>
...
<rdf:Description rdf:about="http://purl.org/semse/foafso/sc/Agent">
  <owl:equivalentClass
    rdf:resource="http://proton.semanticweb.org/2005/04/protont#Agent"/>
</rdf:Description>
...
<rdf:Description rdf:about="http://purl.org/semse/doapso/sp/tester">
  <equivalentConcept
    rdf:resource="http://purl.org/semse/proton/protona#Tester"/>
</rdf:Description>
...

```

Ejemplo III-15: Extracto de la ontología de alineamiento interno correspondiente a la ontología de referencia PROTON

Como su muestra en el ejemplo, las ontologías de alineamiento comienzan definiendo los espacios de nombres de los vocabularios que utilizan para la definición de las relaciones, básicamente, la ontología SEMSE, el lenguaje OWL y el lenguaje RDF. El resto del recurso lo componen reglas que establecen relaciones entre conceptos y propiedades. Se incluye en primer lugar la equivalencia entre la propiedad *title* de la ontología específica *dcelementsso* y la propiedad *title* de la ontología de referencia PROTON. A continuación se incluye la descripción de la equivalencia entre el concepto *Agent* del esquema *foafso* y el concepto *Agent* de la ontología de referencia PROTON. Finalmente, para ilustrar otros tipos de relación, se incluye la correspondencia entre la propiedad *tester* del esquema *doapso* y la clase *Tester* de la ontología PROTON.

Respecto a las relaciones definidas es preciso destacar varios aspectos importantes. En primer lugar, las reglas siempre van a tener dos partes, en primer lugar incluirán una referencia a un concepto de una ontología específica, en segundo lugar incluirán una referencia a un concepto de la ontología de referencia contra la cual relacionan. En

segundo lugar, las reglas establecidas son bidireccionales, de modo que establecen relaciones de equivalencia entre ambos elementos. En tercer lugar la semántica de las relaciones, expresada como equivalencia (*equivalent*), debe interpretarse como proximidad semántica y no como igual-a (*same-as*). De este modo es posible interrelacionar elementos con distinta semántica específica, definida en contexto del esquema que los incluye, con un mismo concepto “próximo”, definido en la ontología de referencia. Finalmente, destacar que se ha hecho uso de las relaciones de equivalencia definidas, de forma nativa, por el lenguaje OWL (*equivalentClass* y *equivalentProperty*). No obstante, debería ser posible definir relaciones propias, por ejemplo con semánticas específicas. Como ejemplo, la tercera regla hace uso de la relación *equivalentConcept* que permite establecer una relación de equivalencia clase-propiedad y propiedad-clase. Con el fin de extender el número y tipo de relaciones de alineamiento, se ha hecho uso de la ontología SEMSE como recurso donde definir nuevas correspondencias. A continuación se muestra la codificación de la relación *equivalentConcept*, extracto de la ontología SEMSE.

```

...
<owl:ObjectProperty rdf:about="http://purl.org/semse/equivalentConcept">
  <domain rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <range rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <isDefinedBy rdf:resource="http://purl.org/semse/">
  <vs:term_status>testing</vs:term_status>
  <skos:definition xml:lang="en-EN">Establece una relación de equivalencia
entre dos conceptos de distinto caracter Clase-Propiedad o Propiedad-
Clase</skos:definition>
  <skos:prefLabel xml:lang="en">equivalent concept</skos:prefLabel>
</owl:ObjectProperty>
...

```

Ejemplo III-16: Extracto de la ontología SEMSE correspondiente a la codificación de la relación equivalentConcept

Como se puede comprobar a partir del ejemplo, la definición de una relación comienza por establecer su URI, en este caso *equivalentConcept* dentro del espacio de nombres donde se define la ontología SEMSE. A continuación, puesto que se trata de una propiedad, es preciso definir dominio y rango de los recursos que relaciona. En este caso se definen como *owl:Thing*, dado que puede tratarse tanto de una clase como de una propiedad. Seguidamente, se indica que la propiedad está definida en la ontología SEMSE y que su estado está en pruebas (*vs:testing*). Finalmente, se incluye la definición de la relación y su etiqueta preferente. Destacar que, tal y como se comentaba en el apartado 2.3.6 *Métodos de correspondencia entre ontologías*, las relaciones podrían haberse definido de forma más específica incluyendo, por ejemplo otros tipos, el grado de correspondencia, etc. La relación incluida tiene como objetivo probar la propuesta minimizando el silencio en las consultas, de ahí su carácter general. El mecanismo de extensión para las relaciones es la ontología SEMSE, donde es posible definir reglas de correspondencia más complejas y/o específicas.

Una vez planteada la representación de la ontología de alineamiento, se pueden extraer las siguientes conclusiones respecto a los criterios de calidad:

- *Extensibilidad*: La representación propuesta se ha diseñado siguiendo este criterio en varios aspectos. Primeramente permite el uso de múltiples ontologías de referencia, al definirse la ontología de alineamiento en función de la

ontología de referencia contra la que relacione. Se proporciona, además, la posibilidad de incluir relaciones entre la ontología de referencia y otras ontologías externas, de modo similar a las establecidas con las ontologías específicas. Permite la adición de relaciones individualmente al ser compatible con el sistema de identificación formal basado en URIs. Permite establecer relaciones de forma independiente a los recursos que relaciona, lo cual facilita la modificación de éstos. Proporciona una estructura sencilla que facilita la actualización del recurso y, finalmente, permite la definición y uso de nuevas relaciones establecidas a través de la ontología SEMSE.

- *Compatibilidad*: La solución propuesta se adecúa a este criterio, en cuanto a que se ha diseñado para garantizar la compatibilidad con los artefactos desarrollados en la propuesta a través del sistema de identificación formal de conceptos y relaciones. El lenguaje seleccionado también facilita este aspecto en cuanto a que proporciona homogeneidad respecto al utilizado en el resto de artefactos.
- *Reutilización*: Se ha atendido a este criterio en cuanto a que la ontología alineamiento permite relacionar una misma ontología de referencia con todas las representaciones semánticas de los esquemas. Del mismo modo, también se relacionan con los conceptos y relaciones incluidos en la ontología de aplicación resultado de la extensión de la ontología de referencia original. En esta línea también se encuentra el hecho de que la ontología de alineamiento va a ser publicada y, por tanto, va a estar accesible para su consulta, descarga y reutilización; aspecto apoyado también por el lenguaje de definición y el sistema de identificación formal utilizados, que facilitan la publicación e identificación de las relaciones entre conceptos y propiedades, para su posterior reutilización.

Definido el modo de correspondencia, a través de las ontologías de alineamiento, el siguiente paso consistirá en realizar la correspondencia entre los conceptos de los esquemas seleccionados y la ontología de referencia, aspecto que se aborda en el siguiente apartado.

3.3.2 Correspondencia de conceptos entre esquemas y la ontología de referencia

El objetivo de este apartado es describir el proceso de correspondencia semántica entre los esquemas seleccionados y la ontología de referencia. Para ello, partiendo del modo de correspondencia definido en el apartado anterior, se establecerán, primeramente una metodología de alineamiento y, a continuación, las relaciones entre los conceptos incluidos en las representaciones semánticas de los esquemas y sus equivalentes en la ontología de referencia.

Durante el proceso de correspondencia y mediante un proceso inductivo, se ha elaborado una metodología de alineamiento a partir del análisis de los casos concretos encontrados. De modo iterativo, cada paso de metodología ha sido aplicado y validado durante el proceso de correspondencia, esto ha obligado a realinear elementos ya procesados pero ha permitido refinar cada paso hasta obtener la metodología que se presenta en este apartado.

Antes de presentar la metodología es preciso indicar los axiomas aplicados en la elaboración de ésta y derivados del contexto de la propuesta. Dicho contexto puede resumirse como: La necesidad de establecer una correspondencia entre elementos con distinto nivel de especificidad en su semántica, lo que implica un proceso de abstracción semántica (específico-genérico) para su interrelación.

- Derivado del contexto, la relación de equivalencia¹⁸ entre dos elementos debe interpretarse como proximidad semántica, abarcando las relaciones de generalización, composición y agregación¹⁹.
- La semántica concreta de los elementos se establece en las ontologías específicas.
- La semántica de los elementos de la ontología de referencia tiene como objetivo recoger la semántica común a múltiples elementos de las ontologías específicas, abstrayéndose de aspectos concretos como formato o codificación.

El porqué de los axiomas establecidos proviene del contexto de la propuesta: la necesidad de relacionar conceptos provenientes de dos ontologías, en el caso más simple, con distinto nivel de abstracción. Con este fin, los axiomas establecen los postulados que permiten relacionar elementos con distinto nivel semántico de forma coherente con el resto de artefactos implicados. Así, los elementos de la ontología de referencia representan a uno o varios elementos específicos, definidos en los esquemas, mediante una semántica común a todos ellos. La semántica específica de los elementos se mantiene en su contexto original, concretamente, en las ontologías específicas. De este modo, aunque el proceso de interrelación implica una pérdida de semántica en el sentido esquema-ontología de referencia, ésta queda preservada en el contexto del esquema y, por tanto, es recuperable. Este aspecto será tratado posteriormente en el apartado *3.4 Validación y verificación de la propuesta*, en el que se mostrará el funcionamiento de la propuesta, una vez integrados todos sus componentes.

Una vez definidos los axiomas provenientes del contexto de la propuesta, a continuación se incluye la descripción de la metodología, resumida en la *Figura III-6*. Destar que, aunque se ha obtenido en el proceso de alineamiento de los esquemas seleccionados con la ontología de referencia PROTON, es aplicable a cualquier esquema y a cualquier ontología de referencia.

¹⁸ Es posible definir relaciones específicas para cada tipo de relación, dado que la propuesta soporta y prevé su extensión en este aspecto a través de la ontología SEMSE.

¹⁹ Las relaciones entre elementos, tal y como se definen en Ingeniería del Software, son: dependencia, asociación, agregación, composición y generalización, siendo las tres últimas las de mayor aportación semántica.

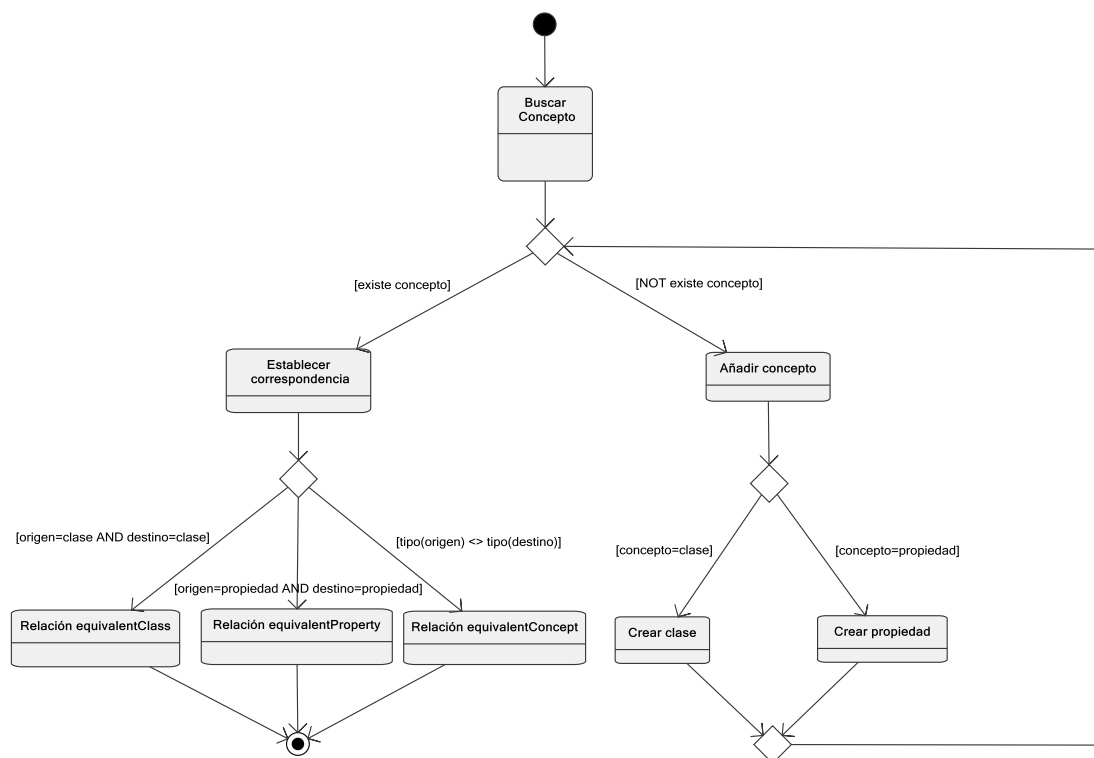


Figura III-6: Diagrama de actividad del proceso de correspondencia de un concepto.

- El proceso de correspondencia de un esquema debe realizarse elemento a elemento²⁰. Esto es, para cada elemento incluido en el esquema o, para ser más precisos, para cada concepto representado en la ontología específica correspondiente al esquema a alinear (*concepto específico*), se establecerá su correspondencia con el concepto equivalente en la ontología de referencia (*concepto destino*). Se trata, por tanto de un proceso iterativo donde cada elemento es analizado de forma independiente según los pasos descritos a continuación.
- Dado un concepto específico a alinear, clase o propiedad, se realizará una búsqueda en la ontología de referencia para encontrar su concepto equivalente. La búsqueda se realizará en base al significado pudiendo, de este modo, relacionarse conceptos específicos con distinta sintaxis a un mismo concepto de la ontología de referencia. En función del tipo de concepto pueden distinguirse dos casos:
 - Si el concepto es una clase: la búsqueda se realizará sobre la jerarquía de conceptos de la ontología de referencia y en base a la semántica de la clase a relacionar. La jerarquía de conceptos de PROTON está detallada en (Terziev, I.; Kiryakov, A.; Manov, D., 2005) y el proceso consiste en recorrerla, partiendo de los niveles superiores, hasta encontrar el concepto buscado.

²⁰ Se recomienda abordar el proceso realizando el alineamiento de clases antes que el alineamiento de propiedades. De este modo, una vez alineadas las clases se tienen establecidas las correspondencias de los dominios y rangos de las propiedades, lo cual facilita su alineamiento.

Como ejemplo, supongamos el alineamiento de la clase *foaf:Agent*. La definición recogida en la ontología específica es “*un agente (ej. Persona, grupo, aplicación software o artefacto físico)*”. La búsqueda partiría de concepto *Entity* como raíz de la ontología; a continuación continuaría por la rama *Object*, puesto que se trata de una entidad que puede existir (real o virtualmente); finalmente se seleccionaría el concepto *Agent* que en PROTON se define como “*algo que puede llevar a cabo una acción, de forma consciente o no*”, definición se completa con la propuesta en DOLCE “*objetos a los cuales adscribimos intenciones, creencias y deseos*” y que se amplía añadiendo “*servicios automáticos, servicios web y servidores*”. Cabe destacar cómo la definición proporcionada por la ontología de referencia es más amplia que la del concepto específico, tal y como se adelantaba anteriormente. El objetivo es poder relacionar varios conceptos específicos con un mismo concepto genérico, de este modo los términos específicos quedan relacionados entre sí, al tiempo que la semántica concreta queda salvaguardada en la ontología específica.

○ Si el concepto es una propiedad: la búsqueda se realizará sobre la jerarquía de propiedades de la ontología de referencia y en base a la semántica de la propiedad a relacionar. Las propiedades de PROTON y su jerarquía están detalladas en (Terziev, I.; Kiryakov, A.; Manov, D., 2005). El proceso de búsqueda es básicamente el mismo que para las clases sólo que las propiedades están organizadas siguiendo una estructura más plana. De este modo, a diferencia de las clases, la búsqueda puede requerir el análisis de toda la estructura. No obstante, en este caso se cuenta con información extra relativa al dominio y al rango sobre el que se define la propiedad, lo cual facilita en gran medida su localización. En última instancia, existe la posibilidad de realizar una búsqueda sintáctica por el nombre del concepto²¹.

Como ejemplo, supongamos el alineamiento de la propiedad *dcelements:title*. La definición recogida en la ontología específica es “*Nombre asignado a un recurso*”. La búsqueda requeriría el análisis de la estructura de las propiedades de PROTON, o bien buscar el concepto equivalente a *recurso* (que según la definición anterior, es el dominio de la propiedad), o bien realizar una búsqueda sintáctica. La propiedad equivalente en PROTON es *protont:title* (estructuralmente dependiente de la raíz de las propiedades de *datos*, puesto que su contenido es un literal) que, además, se define en el dominio de *protont:InformationResource* (representación del concepto *recurso*), y que, sintácticamente, coincide con la etiqueta del elemento *title*. De este modo, cualquiera de los tres modos de búsqueda habría dado como resultado la propiedad buscada, que se define como “*Nombre asignado a un recurso de información. Típicamente, título será un nombre por el que el recurso es formalmente conocido*”. Como ejemplo complementario, si la propiedad hubiera sido *dcelements:creator*, no habríamos obtenido resultado utilizando la búsqueda sintáctica, sin embargo, partiendo del significado de la propiedad “*Entidad principalmente responsable de la creación del recurso. Los ejemplos de Creador incluyen una persona, una organización o un servicio. Típicamente, debe utilizarse el nombre de un Creador para indicar la entidad*”, puede buscarse una

²¹ Este tipo de consultas son soportadas por la herramienta utilizada para la visualización y chequeo de la ontología: Protégé en sus versiones 3.4 y 4.0.

propiedad con rango en el concepto *Agent* (concepto que abstrae persona, organización y servicio en PROTON). En este caso encontramos la propiedad con semántica equivalente *proton:statedBy*, definida como “*Relaciona un recurso con el agente que lo ha creado...*”.

Como resultado de la búsqueda pueden darse dos casos: que el concepto exista en la ontología de referencia o que no se encuentre un concepto con semántica equivalente.

- Si el concepto existe el último paso es establecer la relación en la ontología de alineamiento. En este caso pueden distinguirse tres posibilidades en función de los tipos de los conceptos a interrelacionar.
 - Si el concepto específico es una clase y el concepto destino es una clase: se establece una relación en la ontología de alineamiento haciendo uso de la propiedad *owl:equivalentClass*. Siguiendo el ejemplo anterior, la relación para la clase *foaf:Agent* sería:

```
...
<rdf:Description rdf:about="http://purl.org/semse/foafso/sc/Agent">
  <owl:equivalentClass
    rdf:resource="http://proton.semanticweb.org/2005/04/protont#Agent"/>
</rdf:Description>
...
```

Ejemplo III-17: Extracto de la ontología de alineamiento correspondiente a la relación entre los conceptos foaf:Agent y proton:Agent

- Si el concepto específico es una propiedad y el concepto destino es una propiedad: se establece una relación en la ontología de alineamiento haciendo uso de la propiedad *owl:equivalentProperty*. Siguiendo el ejemplo anterior, las relaciones para las propiedades *dcelements:title* y *dcelements:creator* serían:

```
...
<rdf:Description rdf:about="http://purl.org/semse/dcelementssso/sp/title">
  <owl:equivalentProperty
    rdf:resource="http://proton.semanticweb.org/2005/04/protont#title"/>
</rdf:Description>
<rdf:Description rdf:about="http://purl.org/semse/dcelementssso/sp/creator">
  <owl:equivalentProperty
    rdf:resource="http://proton.semanticweb.org/2005/04/protont#statedBy"/>
</rdf:Description>
...
```

Ejemplo III-18: Extracto de la ontología de alineamiento, correspondiente a la relación entre las propiedades dcelements:title y dcelements:creator con proton:title y proton:statedBy

- Si el concepto específico y el concepto destino son de distinto tipo (propiedad-clase o clase-propiedad): se establece una relación en la ontología de alineamiento, haciendo uso de la propiedad *semse:equivalentConcept*. Como ejemplo, se incluye la relación entre la propiedad *doap:maintainer* y la clase *proton:Maintainer*.

```

...
<rdf:Description rdf:about="http://purl.org/semse/doapso/sp/developer">
  <equivalentConcept
    rdf:resource="http://purl.org/semse/proton/protona#Developer"/>
</rdf:Description>
...

```

Ejemplo III-19: Extracto de la ontología de alineamiento, correspondiente a la relación entre la propiedad doap:maintainer y la clase proton:Maintainer

- Si el concepto no existe, es preciso añadirlo para poder establecer la relación con el concepto específico y, en su caso, con futuros conceptos específicos semánticamente equivalentes. De nuevo, es necesario diferenciar el procedimiento en función del tipo de concepto.
 - Si el concepto es una clase: durante la fase de búsqueda se habrá identificado la posición en la jerarquía donde debería encontrarse la clase destino. El siguiente paso consistirá en crear una nueva clase, equivalente al concepto específico, en la ontología de aplicación correspondiente a la ontología de referencia. La nueva clase creada será integrada en la jerarquía, a través de la relación de generalización, teniendo como resultado la extensión de la ontología original con el nuevo concepto. Destacar que, como se mostrará en el segundo ejemplo, la semántica del concepto deberá preservar el nivel de abstracción del concepto bajo el que se incluya. Como ejemplo, se muestra la extensión de la ontología de referencia con el concepto *foaf:Image*.

```

...
<owl:Class rdf:about="#Image" rdfs:label="Image">
  <rdfs:comment>Class corresponding to those documents which are images.
  Digital images (such as JPEG, PNG, GIF bitmaps, SVG diagrams etc.) are
  examples of Image.</rdfs:comment>
  <rdfs:subClassOf rdf:resource="&ptop;InformationResource"/>
</owl:Class>
...

```

Ejemplo III-20: Extracto de la ontología de aplicación asociada a la ontología de referencia PROTON correspondiente a la inclusión de la clase Image

Como se muestra en el ejemplo, al igual que el resto de conceptos de la ontología de referencia, el nuevo concepto define su URI y su etiqueta. Incluye también, para facilitar el alineamiento de futuros conceptos específicos, una definición. Finalmente, se integra en la jerarquía de la ontología de referencia, en este caso, como subclase del concepto *ptop:InformationResource*. El concepto queda, de este modo, disponible para establecer la relación con el concepto específico *foaf:Image*, cuya correspondencia se realizará bajo el supuesto de que existe, tal y como se establecía anteriormente,.

Destacar que, en algunos casos, es posible que el proceso de extensión implique la inclusión de más de un concepto, con el fin de preservar el grado de abstracción del nivel de la jerarquía donde se vaya a incluir. En concreto, es posible que el concepto a crear sea demasiado específico y requiera crear clases intermedias en la jerarquía, hasta llegar a su nivel de especialización. Como

ejemplo se muestra la inclusión del concepto *EmailReader*.

```
...
<owl:Class rdf:about="#SoftwareProduct" rdfs:label="Software Product">
  <rdfs:comment>A software product abstracted from its organizational and
  ownership aspects</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#top;Product"/>
</owl:Class>
<owl:Class rdf:about="#EmailReader" rdfs:label="Email Reader">
  <rdfs:comment>A software product designed to read and write
  Email</rdfs:comment>
  <rdfs:subClassOf rdf:resource="#papp;SoftwareProduct"/>
</owl:Class>
...
```

Ejemplo III-21: Extracto de la ontología de aplicación protón, correspondiente a la inclusión del concepto EmailReader

En el proceso de inclusión del concepto *EmailReader*, la búsqueda lo ubica como subclase de *Product*. El problema es que no puede definirse como subclase directa de *Product*, puesto que el nivel de abstracción sería inconsistente con el resto de subclases (*MediaProduct*, *WeaponModelOrSystem*, ..., entre otros). Ha sido preciso, como se muestra en el ejemplo, definir una clase intermedia denominada *SoftwareProduct* como subclase directa de *Product* y, a continuación, definir el concepto *EmailReader* como subclase de *SoftwareProduct*. De este modo, *EmailReader* es una subclase indirecta de *Product*, es decir, queda correctamente ubicado en la jerarquía, al tiempo que se preserva el nivel de abstracción de la jerarquía de conceptos.

○ Si el concepto es una propiedad: durante la fase de búsqueda se habrá identificado la posición en la jerarquía donde debería encontrarse la propiedad destino. El siguiente paso consistirá en crear una nueva propiedad, equivalente al concepto específico, en la ontología de aplicación correspondiente a la ontología de referencia. La nueva propiedad creada será integrada en la jerarquía, a través de la relación de generalización, teniendo como resultado la extensión de la ontología original con el nuevo concepto. A diferencia de las clases, la creación de la propiedad debe tener en cuenta el dominio y el rango en los que debe definirse. En este punto, el haber realizado la correspondencia de las clases en primer lugar facilita esta labor, puesto que identificar los rangos y dominios se convierte en un proceso de traducción entre las clases de la ontología específica y las clases de la ontología de referencia (en el caso de que el rango sea un literal, la correspondencia se establece igualmente con un literal). Como ejemplo, a continuación se muestra la extensión de la ontología de referencia con la propiedad *dcelements:hasRelation*.

```

...
<owl:ObjectProperty rdf:about="#hasRelation"
  rdfs:label="hasRelation">
  <rdfs:comment>A related resource. Recommended best practice is to
  identify the related resource by means of a string conforming to a
  formal identification system.</rdfs:comment>
  <rdfs:range rdf:resource="&#x2191;InformationResource"/>
  <rdfs:domain rdf:resource="&#x2191;InformationResource"/>
</owl:ObjectProperty>
...

```

Ejemplo III-22: Extracto de la ontología de aplicación protona, correspondiente a la inclusión de la propiedad hasRelation

A partir del ejemplo destaca, primeramente, el tipo de la propiedad añadida. Concretamente, se ha codificado como *ObjectProperty* puesto que relaciona dos objetos o, visto de otro modo, su rango se define sobre una clase (el tipo *DatatypeProperty* se utilizará en la codificación de aquellas propiedades cuyo rango sea un literal). Al igual que el resto de conceptos de la ontología de referencia, la nueva propiedad define su URI y su etiqueta. Incluye también, para facilitar el alineamiento de futuras propiedades específicas, una definición. Al no especificarse una propiedad padre mediante la relación de generalización, se integra en la jerarquía de la ontología de referencia con subclase de la raíz (debido a que el proceso de búsqueda no ha indicado una propiedad a la cual pueda especializar *hasRelation*). También como parte de la integración en la ontología de referencia, se definen el rango y el dominio de la propiedad. En este caso, ambos coinciden al tratarse de una propiedad que, según su definición, relaciona dos recursos de información, modelados en PROTON mediante la clase *proton:InformationResource*. Una vez añadida la propiedad, sólo restaría incluir su relación dentro de la ontología de alineamiento, tal y como se describió anteriormente.

Respecto al proceso de inclusión de propiedades en la ontología de referencia, durante el alineamiento de los esquemas seleccionados se han encontrado casos especiales que han requerido un análisis y una solución específicos. Concretamente, se trata de propiedades con similitud semántica pero con diferentes ámbitos, rango y/o dominio, ver . A continuación se resumen los casos encontrados y las soluciones propuestas.

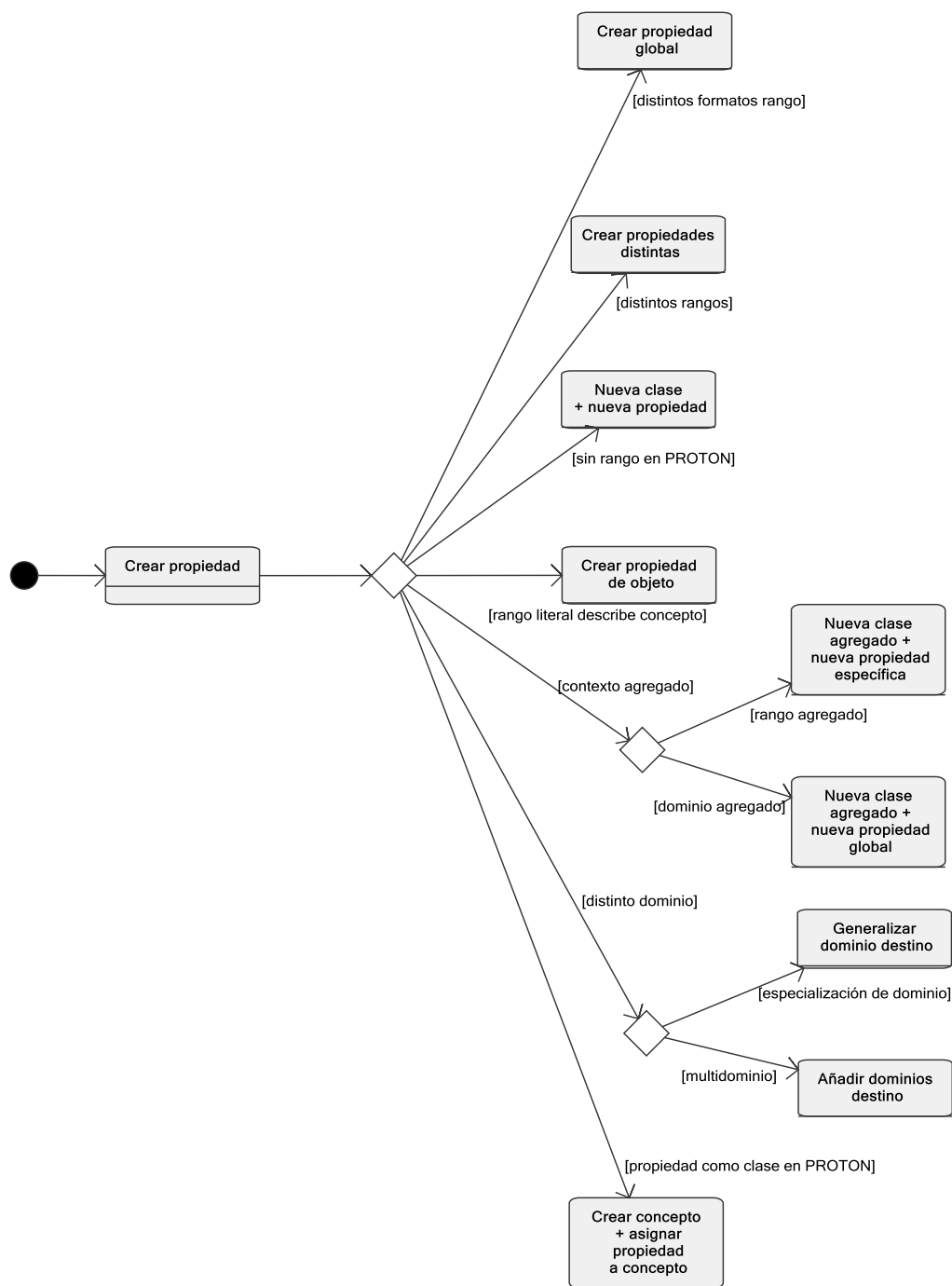


Figura III-7: Diagrama de actividad del tratamiento de casos específicos en la correspondencia de propiedades.

- Propiedades con diferencias en el formato de sus rangos: Son propiedades que difieren en aspectos de formato, rango de valores, etc., asociados al rango de la propiedad. En este caso se ha optado por incluir una propiedad que las englobe semánticamente y relacionar contra ella las propiedades específicas. Un ejemplo de este caso serían las propiedades *foaf:bday* y *vcard:birthday*. La primera se define como “Fecha de nacimiento de una persona” y la segunda

como “*La fecha de nacimiento de este agente, representada como una cadena mm-dd, p.e. '12-31'*”. Como se puede observar en el primer caso no se especifica formato, mientras que en el segundo es necesario incluir la fecha utilizando un formato concreto. La solución ha sido crear una propiedad llamada *protona:birthday* con semántica “*La fecha de nacimiento del Agente, representado por una fecha*”. De este modo, ambas propiedades específicas quedarían relacionadas con la nueva propiedad, permitiéndose así su interoperatividad y manteniéndose los significados concretos a través de las ontologías específicas.

- **Propiedades con distinto rango:** En este caso, aunque las propiedades tenían similitud semántica, sus rangos se definían sobre clases distintas. Como solución se ha optado por incluir propiedades distintas para cada propiedad específica, definiendo sus rangos sobre las clases equivalentes en la ontología de referencia. En función de la semántica de las propiedades, y para mantener la coherencia con la similitud semántica, se ha definido una propiedad como subpropiedad de la otra o, al menos, ambas como subpropiedades de una propiedad que las engloba. Un ejemplo de este caso serían las propiedades *foaf:depiction* y *foaf:logo*. La primera se define como “*Una representación gráfica de alguna cosa*”, la segunda como “*Logo que representa alguna cosa*”. A partir de las definiciones se comprueba que existe similitud semántica en cuanto que ambas propiedades son representaciones gráficas de algo, la diferencia reside en el rango, esto es, en el elemento utilizado para realizar la descripción gráfica: una imagen o un logo. La solución ha consistido en crear dos propiedades *protona:hasImage* y *protona:hasLogo*, correspondientes a *foaf:depiction* y *foaf:logo*, respectivamente. La primera se ha definido con el rango *protona:Image* y la segunda con *protona:Logo*. Considerando que un logo es un tipo específico de imagen, se ha establecido *protona:hasImage* como superpropiedad de *protona:hasLogo*. Finalmente, se han relacionado las propiedades específicas con sus equivalentes en la ontología de referencia. Con esta solución es posible devolver resultados concretos en el caso de consultarse, por ejemplo, por las subpropiedades²². Por otro lado, el establecimiento de la jerarquía entre propiedades permite su uso para contextualizar las consultas pudiéndose, por ejemplo, incluir en los resultados las subpropiedades o las superpropiedades, según la consulta.

- **Propiedades con rango no definido en PROTON:** Se ha observado este caso en aquellas propiedades cuyo rango hacía referencia a una clase no representada en la ontología de referencia²³. Como solución se propone la inclusión de la clase referencia siguiendo el procedimiento descrito anteriormente. Una vez definida la clase, se creará una nueva propiedad con rango asignado a la clase nueva clase. Un ejemplo de este caso se ha encontrado en la propiedad *vcard:mailer*, donde el rango de la propiedad hace referencia a un software de

²² Estos resultados no habrían podido obtenerse de haber optado por una solución más generalista, por ejemplo, relacionar ambas propiedades específicas con la más general, en este caso *protona:hasImage*.

²³ Destacar que, en el conjunto de esquemas seleccionado, no se ha encontrado el caso en el que el dominio no estuviera definido en la ontología de referencia (teniendo en cuenta que se ha realizado antes la correspondencia de las clases que la de las propiedades). En el supuesto de haberse ese caso, el tratamiento sería igual al propuesto para el caso de los rangos no incluidos en la ontología de referencia, asumir que se trata de un concepto no incluido en la ontología y tratarlo como tal.

lectura de correo. Como solución se ha creado el concepto *protona:EmailReader*, definido como “*Producto software diseñado para leer y escribir correo electrónico*”. A continuación, se ha creado la propiedad *protona:hasMailer* estableciendo su rango a *protona:EmailReader*. Finalmente, se ha incluido la relación entre la propiedad específica *vcard:mailer* y la propiedad destino *protona:hasMailer*. A través de esta solución se logra completar la ontología de referencia, con aquellos conceptos necesarios para la completa integración de las propiedades de los esquemas.

- Propiedades con rango literal que describen conceptos: Este caso se ha observado en aquellas propiedades que, definiendo su rango como literal, hacían referencia a un concepto representado como una clase en la ontología de referencia. Considerando que las propiedades de objetos (*ObjectProperties*) permiten representar mayor semántica e integran mejor con la jerarquía de conceptos de la ontología de referencia, en la solución se ha optado por utilizar esta representación en lugar de la propiedad de tipo de datos (*DatatypeProperty*) que correspondería a la propiedad específica. Como ejemplo de este caso, la propiedad *vcard:title* se define con rango literal y expresa el puesto que ocupa un empleado en la organización. En la ontología de referencia existe una clase para representar el rango de la propiedad *vcard:title*, concretamente, la clase *protont:JobPosition*, así como una propiedad que permite asociar el puesto que ostenta con una persona, en concreto *protont:hasPosition*. De este modo, no ha sido necesario crear una nueva propiedad, el problema se ha resuelto relacionando la propiedad específica *vcard:title* con la propiedad *protont:hasPosition*. De este modo, la propiedad específica no sólo queda relacionada con la ontología de referencia, sino que se relaciona con una propiedad que permite mayor representación semántica al establecer su rango, no como un simple literal, sino como una clase integrada en la jerarquía de conceptos.

- Propiedades en el contexto de una agregación: Se trata, en este caso, de propiedades que se definen, bien en su dominio o en su rango, en relación con un concepto agregado. Son ejemplos de este tipo de conceptos aquellos que describen un nombre de persona, agregado por nombre, apellidos, etc., o una dirección postal, agregada por calle, código postal, provincia, etc., entre otros. La solución es diferente en función de dónde se haga referencia al agregado, en dominio o en rango. Para aquellas propiedades que se definen en el dominio de un agregado se propone la creación de una clase que represente el concepto agregado, en caso de que no exista, y la definición de una propiedad que englobe las propiedades componentes, definida en el dominio del agregado. Para aquellas propiedades cuyo rango esté definido en el concepto agregado, se creará una clase que represente el concepto agregado, si no existe, y se creará una propiedad con rango definido sobre dicho concepto. Son ejemplo de propiedades definidas con un agregado como dominio: *pim:city*, *pim:country*, *pim:postalCode*, *pim:stateOrProvince*, *pim:street*, *pim:street2*, *pim:street3* y *pim:zip*, concretamente, sobre el concepto de dirección postal. La solución ha consistido en hacer uso del concepto existente en PROTON *protonu:PostalAddress*, añadiéndole la propiedad *protona:postalAddress*, que

generaliza todas las propiedades específicas²⁴, y contra la cual, finalmente, se han relacionado. Como ejemplo de propiedad cuyo rango se define sobre el concepto agregado se ha encontrado, entre otras, *pim:address*. En este caso se ha definido, al no existir, la propiedad *protona:hasPostalAddress* con rango establecido sobre el concepto agregado *protonu:PostalAddress*. Finalmente, se ha relacionado la propiedad específica *pim:address* con la propiedad destino *protona:hasPostalAddress*. Destacar que la solución propuesta prima la relación de agregación frente a la semántica específica de las propiedades, concretamente en el caso de dominios definidos sobre conceptos agregados, por dos motivos: en primer lugar se ha considerado que, semánticamente, los componentes no tienen sentido por sí mismos sino como parte del agregado, dicho de otro modo, la relación de agregación se ha considerado una composición; en segundo lugar, se ha considerado de mayor interés, de cara a los resultados de una consulta, el devolver todos los conceptos relacionados, teniendo en cuenta que la semántica específica siempre queda salvaguardada por las ontologías específicas²⁵. En resumen, el resultado de la consulta devolverá todos los conceptos relacionados con el agregado y, a través de su significado específico, el usuario tendrá la posibilidad de decidir cuál es el que mejor cubre sus necesidades. En el caso de las propiedades con rango definido sobre un concepto agregado, su representación queda completamente integrada mediante la representación del concepto agregado y la propiedad destino, cuyo rango se define en dicho concepto.

- **Propiedades con distinto dominio:** Este caso se refiere a aquellas propiedades con similitud semántica pero con dominios distintos, concretamente, engloba aquellos casos en los que un dominio es más específico que el otro y a las propiedades multidominio. El primer caso se da cuando una propiedad específica se define en un dominio y existe una propiedad destino cuyo dominio es más general o más específico que el de la propiedad a incluir. El segundo caso se da en aquellas propiedades que se definen para varios dominios o, lo que es lo mismo, su dominio se asocia a varias clases. Como solución para el primer caso, se propone la modificación del dominio de la propiedad destino, de forma que englobe ambos dominios, esto es, se tomará entre el dominio de la propiedad específica y la propiedad destino, el más general y se asignará a la propiedad destino. Finalmente se establecerá la relación entre la propiedad específica y la propiedad destino. Como solución para el segundo caso, se ha optado por crear una propiedad destino, si no existiera, y añadir los dominios concretos a dicha propiedad, esto es, modificar o crear una nueva propiedad cuyo dominio está formado por el conjunto de dominios definidos en la propiedad específica (incluyendo los que ya tuviera la propiedad destino, en caso de que ya existiera).

Como ejemplo del primer caso tenemos las propiedades *foaf:bdays* y *vcard:birthday*. La primera se define en el dominio de una persona, mientras que la segunda lo hace sobre el dominio de un agente. Dependiendo de cuál se haya

²⁴ En este caso ha sido suficiente definir una única propiedad, dado que todas las propiedades del agregado eran semánticamente homogéneas. En el caso de haberse tratado de propiedades heterogéneas, se habrían generado propiedades destino distintas para cada semántica. Un ejemplo de este caso podría ser los atributos que componen un certificado X509.

²⁵ En el caso de haber primado la semántica de las propiedades, la solución también es posible y consistiría en generar distintas propiedades destino con el agregado como dominio.

alineado en primer lugar, el dominio de la propiedad destino será persona o agente. En cualquier caso, al alinear la segunda propiedad estaremos ante una discrepancia en el dominio de la propiedad específica y la propiedad destino. Independientemente del orden de alineamiento, la propiedad resultante *protona:birthday* tendrá asignado como dominio el concepto más general, en este caso, *protont:Agent*. Finalmente, sólo resta establecer la relación entre la propiedad específica y la propiedad destino.

Un ejemplo del segundo caso es la propiedad *doac:startQualificationDate*, que representa la fecha de inicio de un proceso de cualificación, y que se define, tanto para procesos educativos *protona:Education*, como para experiencia laboral *protona:Experience*. La solución ha pasado por crear una propiedad destino cuyo dominio ha sido definido como la conjunción de los dominios establecidos en la propiedad específica. Como último paso, se establecerá la relación entre la propiedad específica y la propiedad destino. Destacar que la solución tiene como objetivo maximizar la interoperatividad entre las propiedades específicas²⁶ al tiempo que la semántica específica es preservada por las ontologías específicas. También es importante el hecho de que el alineamiento de cada propiedad es independiente del resto de propiedades específicas y del orden en el que se realice. Estos dos aspectos son fundamentales teniendo en cuenta, por un lado, la complejidad creciente que supondría que el alineamiento de cada propiedad dependiera de las relaciones del concepto destino con propiedades interrelacionadas anteriormente, y por otro, que el orden de alineamiento no puede influir en el resultado final puesto que, en tal caso, se obtendrían distintos resultados dependiendo de la secuencia en la que se alinearan los conceptos, del mismo o distinto esquema.

- Propiedades como clases en la ontología de referencia: Este caso trata de aquellas propiedades específicas que tienen representación en la ontología de referencia como clases. Aunque podrían tratarse como propiedades no incluidas en la ontología de referencia, esta solución se ha descartado, dado que generaría incongruencias al tener dos representaciones alternativas, clase y propiedad. La solución seleccionada ha consistido en relacionar las propiedades específicas con su representación en la ontología de referencia, en este caso, como clases. Sólo se ha encontrado un caso durante el proceso de correspondencia, concretamente, ligado a la representación de roles. El caso en cuestión se ha producido en el alineamiento de las propiedades: *doap:maintainer*, *doap:tester*, *doap:translator*, etc., es decir, propiedades que establecen roles asumidos por personas en un proyecto. En el caso de PROTON, los roles son representados mediante la clase *protont:Role*, que permite representar el rol que juega una entidad en algo que ocurre. La solución ha pasado por crear el concepto destino en función de la representación incluida en la ontología de referencia. De este modo, se han

²⁶ Es preciso destacar el hecho de la aparente asimetría en el tratamiento de los dominios y los rangos cuando estos son distintos, siempre bajo el supuesto de similitud semántica. El tratamiento de los rangos implica la generación de propiedades específicas que se relacionen con las clases concretas, es decir, tiene como objetivo devolver resultados precisos. El tratamiento de los dominios tiene como objetivo maximizar la interoperatividad. Para ello se apoya en la relación que establece la jerarquía de conceptos: la generalización. En virtud de esta relación, las subclases heredan las propiedades de la clase padre y, debido a esta propiedad, aquellas relaciones definidas en las superclases se encuentran también en las subclases. Este comportamiento es aprovechado en este caso, dado que generalizar el dominio implica que ambas propiedades van a mantener coherencia con su dominio original, directa o indirectamente, a través de la herencia.

creado subclases de la clase *protont:Role* que se corresponden con los roles a incluir, a saber *protona:Maintainer*, *protona:Tester*, *protona:Translator*, etc. Para finalizar la interrelación, una vez definidos los roles, se ha establecido la correspondencia entre las propiedades específicas y las clases destino. La solución, en este caso, prima la consistencia del conocimiento incluido en la ontología de referencia, sacrificando la representación del tipo de concepto específico. En cualquier caso, favorece la interoperatividad, salvaguarda la semántica a través de la ontología específica y el tipo de la propiedad a través del esquema cualificado.

Como se introducía anteriormente, la metodología presentada se ha obtenido y aplicado durante el proceso de correspondencia. Así, para el conjunto de esquemas seleccionados, se ha interrelacionado cada uno de sus elementos con los conceptos de la ontología de referencia. Como resultado del proceso y siguiendo el método expuesto en el apartado 3.2.3 *Representación compatible de la ontología de referencia*, la ontología de referencia ha sido extendida con los conceptos destino necesarios para el alineamiento, incluidos en la ontología de aplicación correspondiente. Del mismo modo, el alineamiento ha producido un conjunto de relaciones que, según el método propuesto en el apartado 3.3.1 *Evaluación y selección del método de correspondencia*, han sido incluidas y publicadas en la ontología de alineamiento correspondiente a la ontología de referencia.

Una vez realizado el proceso, sólo resta publicar los resultados obtenidos, aspecto que se aborda en el siguiente apartado.

3.3.3 Publicación de la ontología de correspondencia

Una vez definido y completado el proceso de correspondencia, se ha procedido a su publicación en Web. El objetivo de la publicación es poner a disposición de la plataforma de gestión las correspondencias generadas, permitiendo así su acceso, uso y reutilización. Aunque no es el objetivo principal, salvo para los administradores, también queda disponible para su consulta por parte de los usuarios.

Antes de la publicación de las representaciones propuestas, ha sido necesario tomar una serie de decisiones de diseño que se resumen a continuación:

- Definición de la política de nombrado: Las ontologías de alineamiento recibirán el nombre de la ontología de referencia que extienden, seguido de la etiqueta '*specific_ontologies_linkage*', si es alineamiento interno, o '*external_ontologies_linkage*', si es alineamiento externo. Como ejemplo, para la ontología de referencia PROTON, las ontologías de alineamiento interno y externo son, respectivamente, *proton_specific_ontologies_linkage* y *proton_external_ontologies_linkage*.
- Definición del método de publicación: Tal y como se indicaba en el apartado 3.1.4 *Publicación de esquemas semánticos*, las ontologías de alineamiento se ha publicado utilizando como base la URI persistente <http://purl.org/semse>.

- Definición de los espacios de nombres: A partir de las recomendaciones del *World Wide Web Consortium* para la publicación de vocabularios RDF (Consortium, W.W., 2008), se ha decidido utilizar un esquema basado en *slash* para el acceso a las representaciones y su contenido. Respecto a la estructura, partiendo de la URL base, se han definido subdominios para cada una de las ontologías de referencia, dentro de los cuales, se ubican la ontología de aplicación correspondiente. De este modo, cada representación tiene un subdominio asignado, establecido como *internalmapping* y *externalmapping*, que permite el acceso unívoco a su contenido, por ejemplo: <http://purl.org/semse/proton/internalmapping/> es el subdominio asignado a la ontología de alineamiento interno correspondiente a la ontología PROTON.

Dentro del servidor desplegado para la publicación de recursos, se ha generado una estructura de directorios para albergar las ontologías de alineamiento (directorio *mappingontologies*). Una vez ubicados los recursos en el servidor, el siguiente paso ha sido crear las reglas de acceso a dichos recursos, en función de la estructura de directorios definida y de la definición de los espacios de nombres. El método para incluir estas reglas se ha basado en el uso del fichero *.htaccess* definido en el apartado 3.1.4 *Publicación de esquemas semánticos* cuya estructura, relativa a la publicación de ontologías de alineamiento, se resume a continuación:

```
...
# Reglas de sobrescritura para devolver ontologías de alineamiento
RewriteRule ^proton/internalmapping/$
mappingontologies/proton_specific_ontologies_linkage.rdf
RewriteRule ^proton/externalmapping/$
mappingontologies/proton_external_ontologies_linkage.rdf
...
```

Ejemplo III-23: Extracto del fichero de configuración .htaccess

Como se muestra en el ejemplo, el fichero de configuración añade dos reglas por cada ontología de referencia, una regla por cada ontología de alineamiento a compartir, en este caso *proton_specific_ontologies_linkage* y *proton_external_ontologies_linkage*. A través de estas reglas se permite el acceso al contenido de las ontologías de alineamiento a partir de la URL del subdominio que corresponda, por ejemplo: <http://purl.org/semse/proton/internalmapping/> proporcionará acceso al contenido de la ontología de alineamiento interno, incluida en el fichero *proton_specific_ontologies_linkage.rdf*.

3.4 Validación y verificación de la propuesta

Una vez expuesto el planteamiento de la propuesta, es necesario mostrar y verificar su funcionamiento.

Según los conceptos de comprobabilidad de Ingeniería del Software (Braude, E.J., 2000), se procederá a validar y verificar la propuesta, al tiempo que se ejemplifica su funcionamiento. Se validará la propuesta comprobando que satisface los requisitos para los que se diseñó, esto es, que realiza su función. Se verificará comprobando que realiza su función correctamente, esto es, comprobando la corrección de los resultados que proporciona. Con este fin, se definirá y aplicará un método de prueba, cuyos resultados

serán comparados y discutidos.

Cabe destacar que, al carecer de una herramienta que automatice las pruebas, no es posible aplicar métodos cuantitativos de valoración, de modo que las consultas y ejemplos incluidos en este apartado tienen un carácter ilustrativo. La evaluación de la propuesta será realizada en el *Capítulo IV: Resultados, Evaluación y Discusión*, haciendo uso de la metodología de evaluación DESMET (Kitchenham, 1996) y aplicando métodos cualitativos de valoración.

Previo a la verificación y validación de la propuesta, es preciso definir cuáles van a ser los indicadores de calidad a utilizar. Estos indicadores van a permitir obtener una valoración de la calidad del resultado, esto es, la verificación de los resultados y, por ende, de la propuesta. En el proceso de obtención de los resultados de prueba se estará ejemplificando el funcionamiento, cubriéndose así el aspecto de validación de la propuesta.

En el ámbito de la recuperación de información, los indicadores de calidad se suelen definir en base al concepto de relevancia (Gómez, R., 2003). Este concepto expresa si un documento resuelve una necesidad informativa, esto es, si debería o no estar en el conjunto de resultados, dada una consulta. Los documentos pueden ser recuperados o rechazados al establecer la comparación entre la pregunta y la colección. El conjunto de documentos recuperados se divide, salvo en los sistemas perfectos, en dos grupos: documentos relevantes recuperados, es decir aquellos que se han recuperados correctamente y los no relevantes, recuperados erróneamente que provocan ruido en la salida. Los documentos no recuperados, que a su vez se dividen en los relevantes, rechazados por el sistema de manera errónea y los no relevantes, rechazados de manera correcta por el sistema. La siguiente figura resume la casuística posible:

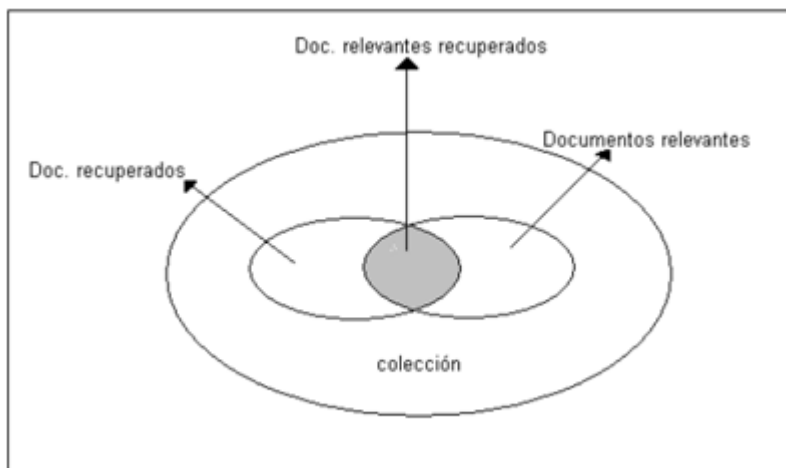


Figura III-8: Esquema de recuperación de documentos. Extraído de (Baeza, R.; Ribeiro, B., 1999).

En base al concepto de relevancia, se definen (Kent, A. et al., 1955) dos indicadores de calidad básicos: precisión (*precision*) y exhaustividad (*recall*).

La precisión es la proporción de material recuperado realmente relevante, del total de los documentos recuperados. El resultado de esta operación está entre 0 y 1. Así, la recuperación perfecta es en la que únicamente se recuperan los documentos relevantes y por lo tanto tiene un valor de 1.

$$\text{Precisión} = \frac{\text{Documentos Relevantes Recuperados}}{\text{Documentos Recuperados}}$$

Figura III-9: Definición de la precisión.

Esta medida está relacionada con el concepto de ruido. De este modo, cuanto más se acerque el valor de la precisión a 0, mayor será el número de documentos recuperados que no le sirvan al usuario y por lo tanto el ruido que encontrará será mayor.

La exhaustividad, aunque en menor medida que la precisión, es el otro concepto más utilizado en la evaluación de los sistemas de recuperación. Es la proporción de material relevante recuperado, del total de los documentos que son relevantes en la colección, independientemente de que éstos se recuperen o no. Esta medida es inversamente proporcional a la precisión.

$$\text{Exhaustividad} = \frac{\text{Documentos Relevantes Recuperados}}{\text{Documentos Relevantes}}$$

Figura III-10: Definición de la exhaustividad.

Si el resultado de este cálculo tiene como valor 1, tendremos la exhaustividad máxima, habiéndose recuperado todos los documentos relevantes que había en la colección, por tanto, la recuperación será perfecta. Esta medida está relacionada el concepto de silencio informativo. De este modo, cuanto más se acerque el valor de la exhaustividad a 0, menor será el número de documentos relevantes recuperados y por lo tanto el silencio del resultado será mayor.

Una vez definidos los indicadores de calidad, el siguiente paso es establecer la metodología de prueba. Concretamente:

- Las definición de las consultas que se van a evaluar.
- La selección del conjunto de términos a consultar.
- La aplicación de las medidas de calidad y su comparativa.

Para realizar la evaluación se han definido las siguientes consultas:

- Consulta sintáctica: Se realizará la consulta de un término sobre la representación semántica del conjunto de esquemas, discriminando si se trata de una clase o una propiedad. La consulta se realizará por proximidad sintáctica, esto es, se seleccionarán aquellos conceptos que contengan el término buscado o un superconjunto del mismo, por ejemplo: si el término buscado es *cuenta* se incluirán en los resultados *cuenta*, *cuentaOnline*, etc. El resultado devuelto al usuario será el conjunto de elementos que contengan el término.
- Consulta conceptual: El usuario introducirá un concepto y se le devolverán los

conceptos y sus significados para que pueda escoger cuál de ellos cubre mejor sus necesidades. La consulta se iniciará con una búsqueda sintáctica, definida en el punto anterior, pero añadiendo información de retroalimentación del usuario. De este modo, al usuario se le presentará un primer conjunto de resultados, obtenidos de la consulta sintáctica, para que éste pueda precisar el significado que más se aproxima al concepto que busca. Partiendo del concepto seleccionado, se consultará, mediante la ontología de alineamiento, qué concepto de la ontología de referencia está relacionado con el concepto buscado. A través del concepto de la ontología de referencia se consultará, mediante la ontología de alineamiento, qué conceptos de las ontologías específicas se relacionan con él, obteniéndose de este modo el conjunto completo de conceptos que se relacionan semánticamente con el concepto de la ontología de referencia. Este conjunto final de conceptos conformará el resultado de la consulta.

- Consulta conceptual contextualizada: Similar a la consulta conceptual definida anteriormente, una vez obtenidos los conceptos de la ontología de referencia mediante la ontología de alineamiento, directamente relacionados, se seleccionarán también los conceptos relacionados, padres e hijos, en la ontología de referencia. Estos nuevos conceptos, indirectamente relacionados, serán incluidos aunque con un grado de relevancia (*rank*) menor. El resto de la consulta se realizará del mismo modo para los conceptos con relación directa y para aquellos con relación indirecta. El resultado mostrará un tipo y otro de relación de forma diferenciada.
- Consulta con buscadores semánticos: Además de estas consultas, y con el fin de poder establecer una comparación con otros sistemas de recuperación conceptual, se realizarán consultas utilizando los buscadores semánticos más populares, concretamente: *Swoogle*, *Falcons*, *Sindice* y *Watson*. En este caso, y para que los resultados puedan ser comparables, la consulta se realizará del siguiente modo: primeramente, se buscará el concepto haciendo uso del buscador concreto y, a continuación, se filtrarán aquellos resultados que no provengan de los esquemas seleccionados. De este modo, los resultados son comparables en cuanto a que el ámbito del resultado se limita al conjunto de esquemas utilizado en el estudio.

Los resultados de estas consultas serán comparados con los obtenidos al realizar la consulta sintáctica sobre los esquemas originales, incluyendo también el tipo del elemento, clase o propiedad, como discriminante.

Una vez definidas las consultas, es preciso definir el conjunto de términos a consultar. Al no contarse con una herramienta que permita su automatización, se ha seleccionado un número reducido de términos a título ilustrativo. La selección ha tenido como objetivo cubrir la mayor casuística posible, de este modo, se han incluido clases y propiedades que implicaran la resolución de aspectos como sinonimia, homonimia y plurilingüismo. Concretamente, los términos a buscar serán: *agent*, *title* e *imagen*.

Una vez establecidos los términos a consultar, se obtendrán los conjuntos de resultados de cada consulta y se calcularán, para cada uno de ellos, los valores de los indicadores de calidad. Una vez calculados, se compararán y discutirán los resultados. Destacar en

este punto que la evaluación de los resultados depende del conjunto de documentos relevantes (*ground truth*), definido por la consulta. En las pruebas realizadas se ha establecido como consulta: obtener el conjunto de documentos que incluyan el concepto buscado, según la definición seleccionada por el usuario, y todos los conceptos equivalentes, pertenecientes al conjunto de esquemas seleccionado, es decir, sinónimos en el mismo u otros esquemas.

Como ejemplo básico del proceso de evaluación, a continuación se incluye la descripción, incluyendo resultados parciales, de la búsqueda del término *Agent*.

- Conjunto de documentos relevantes: El conjunto de documentos relevantes, obtenidos a partir de las definiciones de los elementos en los esquemas originales, está formado por los elementos: *foaf:Agent* y *pim:SocialEntity*.
- Consulta sintáctica: La consulta sintáctica, realizada sobre las ontologías específicas, devolvería como resultados los conceptos: *foaf:Agent* y *foaf:Group*.
- Consulta conceptual: La consulta conceptual partiría del conjunto de resultados devuelto por la consulta sintáctica, de los cuales, el usuario seleccionaría el que más se aproxime al significado del concepto que busca, en este caso: *foaf:Agent*. Partiendo del concepto seleccionado, la búsqueda en la ontología de alineamiento devolvería el concepto *ptop:Agent* de la ontología de referencia. La siguiente fase de la consulta buscaría los conceptos de las ontologías específicas relacionados con *ptop:Agent*, obteniéndose como resultado: *foaf:Agent* y *pim:SocialEntity*. Este conjunto de conceptos conformaría el resultado de la consulta.
- Consulta conceptual contextualizada: En este caso también se partiría de la consulta sintáctica y de la selección del concepto por parte del usuario. También se obtendría *ptop:Agent* como concepto relacionado en la ontología de referencia y como resultados *foaf:Agent* y *pim:SocialEntity*. A estos resultados, la consulta podría añadir otros, provenientes del contexto del concepto *ptop:Agent* en la ontología de referencia. Como ejemplo se han incluido los obtenidos del concepto padre y de los conceptos hijos con grado uno (padre e hijos de primer nivel). De este modo, el contexto de *ptop:Agent* se define como *ptop:Object*, concepto padre, y *ptop:Group*, *ptop:Person*, como conceptos hijos. Los conceptos de las ontologías específicas, referidos al contexto, serían *foaf:Person*, *pim:Person* y *foaf:Group*. Destacar que los conceptos obtenidos del contexto complementan el resultado pero deben ser tratados con un nivel de relevancia menor. Al poder ser identificados y discriminados, no se han incluido en cálculo de los indicadores de calidad, puesto que no interfieren sino que complementan el resultado.
- Consulta con buscadores semánticos: En este tipo de consulta se puso de manifiesto algunas limitaciones que presentan este tipo de recuperadores, y que, añadidas a la necesidad de obtener resultados comparables, han provocado que algunos de los propuestos no hayan podido ser utilizados. Concretamente, los problemas encontrados en cada caso, se resumen en:

- *Swoogle*: En el proceso de consulta, aún restringiendo la búsqueda por tipo de elemento, el número de resultados es muy elevado, por ejemplo: buscando ontologías, la propiedad “*title*” devuelve 51.000 resultados. El problema, en este caso, se debe a que no es posible especificar como parámetro el conjunto de ontologías sobre las que se desea buscar. El intento de filtrar los resultados tampoco es posible, puesto que no se puede recuperar el conjunto completo, debido a su tamaño. Por estos motivos, Swoogle no ha podido ser utilizado en esta comparativa.
- *Falcons*: Las consultas en este buscador tampoco permiten especificar el rango de ontologías sobre el que se desea recuperar. No obstante, sí permite filtrar los resultados utilizando el espacio de nombres. Desafortunadamente, este buscador no incorpora dos de los esquemas seleccionados, concretamente, *DOAC* y *PIM*. Debido a que *PIM* interviene en las consultas planteadas, esto hace que los resultados no sean comparables y descarta el uso de este buscador en esta evaluación.
- *Sindice*: En la línea de *Swoogle*, incluye todos los esquemas del conjunto seleccionado. Desafortunadamente, tampoco permite especificar el rango de ontologías sobre el que buscar, ni filtrar el resultado, limitándolo al conjunto de esquemas seleccionado. Tampoco ha sido posible utilizar este buscador en la comparativa.
- *Watson*: En este caso, tampoco es posible especificar el rango de ontologías destino, no obstante, el conjunto de resultados se organiza utilizando el espacio de nombres y se devuelve completo. Esto permite realizar un filtrado manual de los resultados, permitiendo que sean comparables con los obtenidos en las consultas propuestas. Respecto al conjunto de esquemas, *Watson* no incluye el esquema *DOAC* pero, afortunadamente, no interviene en las consultas planteadas, ni sintáctica ni semánticamente. De este modo, se han incluido los resultados de este buscador en la comparativa. Concretamente, para el caso en cuestión, la búsqueda del concepto *Agent*, el resultado devuelto es: *foaf:Agent*.
- Consulta sintáctica sobre esquemas originales: Con el fin de comparar las consultas anteriores con los resultados de un buscador común, se ha realizado una última consulta sintáctica sobre los esquemas originales²⁷. Al igual que en la consulta sintáctica, la búsqueda ha incluido el tipo de elemento como discriminante. Como resultado de la consulta sobre los esquemas originales, se han obtenido los siguientes conceptos: *foaf:Person*, *foaf:Organization*, *foaf:Group* y *foaf:Agent*.

²⁷ Para obtener resultados más realistas, se ha considerado que los documentos son procesados antes de realizar la consulta. De este modo, no se han tenido en cuenta correspondencias sintácticas encontradas en comentarios, encabezados, etc., dentro de los esquemas originales.

A continuación se resumen los resultados, incluyendo los valores de los indicadores de calidad propuestos.

	Sintáctica	Conceptual	Contextual	Sintáctica original	Watson
Documentos Relevantes Recuperados	1	2	2	1	1
Documentos Recuperados	2	2	2	4	1
Documentos Relevantes	2				
Precisión	0.5	1	1	0.25	1
Exhaustividad	0.5	1	1	0.5	0.5

Tabla III-13: Resumen de los resultados de las consultas sobre el término ‘Agent’ y valores de los indicadores de calidad

La siguiente consulta evaluada se ha definido sobre un término con múltiples acepciones, el objetivo es verificar el comportamiento de la propuesta en el caso de homonimia. Concretamente, el término a buscar es la propiedad *vcard:title*, con la acepción concreta de “cargo que ocupa una persona en una organización”. El término *title* aparece en otros esquemas con distintas acepciones, tales como “nombre asignado a un recurso de información” o “título o tratamiento de una persona”.

La siguiente tabla resume los resultados y los valores de los indicadores de calidad para la consulta propuesta.

	Sintáctica	Conceptual	Contextual	Sintáctica original	Watson
Documentos Relevantes Recuperados	2	2	2	2	1
Documentos Recuperados	6	2	2	6	4
Documentos Relevantes	2				

Precisión	0.3	1	1	0.3	0.25
Exhaustividad	1	1	1	1	0.5

Tabla III-14: Resumen de los resultados de las consultas sobre el término ‘title’ y valores de los indicadores de calidad

Destacar en este caso que la consulta contextualizada podría incorporar, por tratarse de una propiedad, información relativa a los conceptos que relaciona el término, esto es, dominio (*ptop:Person*) y rango (*ptop:JobPosition*).

Como último ejemplo de consulta, se propone la búsqueda de propiedades relacionadas con el término *imagen*, el objetivo en este caso es doble: por un lado, verificar el soporte que proporciona la propuesta al plurilingüismo; y por otro, realizar una consulta sin hacer uso de una propiedad conocida, esto es, sin utilizar conocimiento previo sobre los elementos de los esquemas. Como es lógico, el término no va a ser encontrado en la versión original de los esquemas, sin embargo, la propuesta debe proporcionar resultados independientemente del idioma.

Respecto al conjunto de documentos relevantes, dependerá del significado del término que el usuario desee. En el conjunto de esquemas seleccionado, existen propiedades relacionadas con el término *imagen* con dos acepciones distintas: “miniatura de una imagen” e “imagen personal”. Suponiendo la segunda acepción como la buscada por el usuario, el conjunto de documentos relevantes sería *vcard:photo* y *foaf:img*.

La siguiente tabla resume los resultados y los valores de los indicadores de calidad para la consulta propuesta.

	Sintáctica	Conceptual	Contextual	Sintáctica original	Watson
Documentos Relevantes Recuperados	1	2	2	0	0
Documentos Recuperados	2	2	2	0	0
Documentos Relevantes	2				
Precisión	0.5	1	1	0	0
Exhaustividad	0.5	1	1	0	0

Tabla III-15: Resumen de los resultados de las consultas sobre el término ‘imagen’ y valores de los indicadores de calidad

Destacar en este caso que la consulta contextualizada podría incorporar, por tratarse de una propiedad, información relativa a los conceptos que relaciona el término, esto es, dominio (*ptop:Person*) y rango (*ptop:Image*). Además, hay que añadir que el contexto de la ontología de referencia habría incluido, como superpropiedad, la propiedad *papp:hasImage*. A partir de este concepto, se habría podido ampliar el conjunto de resultados con la propiedad *foaf:depiction* y la información relativa a sus conceptos asociados por dominio y rango.

A partir de las consultas definidas y los términos seleccionados, se pueden destacar los siguientes aspectos sobre los resultados obtenidos:

- Mediante la ejemplificación que proporcionan la consultas y términos propuestos, se ha podido validar la propuesta, confirmando que cubre los requisitos a partir de los cuales se definió, y que se pueden resumir en: mejora de la interoperatividad entre esquemas heterogéneos, posibilidad de la reutilización de esquemas, recuperación de un conjunto de resultados derivado de relaciones entre conceptos y recuperación conceptual.
- Se ha verificado la propuesta a través del establecimiento de los criterios de calidad, cuyos valores obtenidos muestran como las búsquedas conceptuales poseen la máxima precisión y exhaustividad en los resultados, muy por encima de los valores generados para una búsqueda sintáctica clásica y mejorando los obtenidos mediante el recuperador semántico *Watson*.
- En relación al uso de buscadores semánticos, se han encontrado deficiencias en cuanto a los resultados devueltos al usuario. La imposibilidad de restringir el dominio de la búsqueda y de filtrar los resultados, tienen como efecto un conjunto de resultados ingente, y en algunos casos incompleto, que puede resultar de escasa utilidad para el usuario. Esto hace que mejorar la exhaustividad de los resultados y la presentación al usuario de los mismos, tal y como se realiza en la propuesta, sean aspectos fundamentales a alcanzar en este tipo de recuperadores.
- Respecto a los esquemas utilizados, destacar que, a diferencia de los utilizados en la evaluación de algoritmos de correspondencia, habitualmente con pocos elementos (Do, H.H.; Melnik, S.; Rahm, E., 2002), en este caso se trata de esquemas reales con un número de elementos significativo.
- Indicar también que las consultas definidas tienen un carácter ilustrativo y que podrían definirse consultas más complejas. Por ejemplo, podría habilitarse el uso de operadores booleanos, consultas generadas a partir de técnicas de procesamiento del lenguaje natural, etc. En cuanto al procesamiento de las consultas, éstas podrían sacar más partido de la base ontológica generada, por ejemplo, incluyendo rangos y/o dominios en las consultas de propiedades, incluso, haciendo uso de las relaciones definidas en la ontología para los conceptos establecidos en dichos rangos y dominios.
- Respecto a los resultados devueltos, se ha indicado cómo podrían completarse mediante el contexto proporcionado por la ontología de referencia. Asimismo

podrían ordenarse en base a metadatos definidos para los esquemas, por ejemplo, por popularidad o número de documentos relacionados.

- Finalmente, es importante el carácter 2.0 que se propone en las consultas. A partir de la participación del usuario, y de una forma sencilla, es posible lograr una gran precisión y exhaustividad en los resultados obtenidos. Esta filosofía también se aplica a la hora de proporcionar los resultados, permitiendo al usuario seleccionar el concepto que mejor se adecúa a sus necesidades.

3.5 Proceso de desarrollo del sistema informático de soporte

Una vez descrito el planteamiento de la propuesta y ejemplificado su funcionamiento, el objetivo que aborda este apartado es la definición de un sistema informático que soporte la gestión de la solución propuesta. Desde octubre de 2007, se está desarrollando un proyecto, dentro del Plan Nacional de I+D+i, titulado “*Desarrollo de un sistema de Recuperación Conceptual mediante Niveles Semánticos en la representación de Esquemas de Metadatos*”, cuyo objetivo es el desarrollo un sistema de soporte basado en el planteamiento propuesto en esta tesis.

Desde un punto de vista general, la aplicación intentará potenciar el uso de estructuras reusables de conocimiento, ontologías, para su aplicación en el trabajo diario de los interesados en la materia, resolviendo los problemas planteados en la transcripción, registro, recuperación y explotación de fuentes documentales mediante técnicas de la ingeniería del conocimiento y, más específicamente, bajo el punto de vista de la ingeniería ontológica. El objetivo último es que el sistema permita gestionar, reutilizar e interoperar ontologías, especificadas como esquemas semánticos, a partir de los conceptos registrados en recursos semánticos compartidos, teniendo en cuenta toda la propuesta teórica presentada anteriormente y definida para tal fin. Además de lo anterior, el sistema deberá proporcionar una síntesis e integración de la información existente en las fuentes, eliminando redundancias y realizando inferencias.

El producto estará destinado a un amplio abanico de usuarios, que no tienen por qué ser expertos en el manejo de este tipo de herramientas. De este modo, el sistema pretende ser de fácil manejo y, sobretodo, intuitivo. Para ello, se deberá prescindir de adornos que retrasen o dificulten la interacción. La aplicación deberá tener un diseño minimalista, que favorezca la concentración en su función principal, establecida por los requisitos de usuario.

Aunque existen soluciones parciales, este sistema aún no ha sido desarrollado, lo que supone de un incentivo más para el producto. Se trata de una herramienta eficaz que, si el usuario lo estima conveniente, puede sustituir o complementar con otras herramientas parciales que existen en el mercado actual.

El sistema a desarrollar, bautizado como *Sistema de Gestión del Conocimiento basado en Ontologías (SGCO)*, deberá cubrir las siguientes funciones:

- Gestión de ontologías: creación, edición, eliminación, exportación, importación, consultas, fusiones y correspondencias.

- Gestión de perfiles de aplicación: añadir, eliminar, importación y exportación.
- Gestión de documentos: creación, edición, eliminación, búsqueda e indización.
- Gestión de perfiles.
- Gestión de usuarios: alta, baja, edición, validación, permisos.
- Consultar ayuda.

Las principales tareas incluidas en esta fase se resumen en la siguiente figura:



Figura III-11: Tareas del proceso de desarrollo del sistema informático de soporte

3.5.1 Especificaciones informales

Como primer paso para el desarrollo del sistema, se ha elaborado un conjunto de *Especificaciones Informales*. El objetivo de este documento es obtener una idea general de los requisitos funcionales, funcionalidad, que debe proporcionar la aplicación, así como de otros aspectos de interés como: usuarios de la aplicación, sistemas con los que debe interoperar, etc.

Como resumen de las especificaciones informales del sistema, en los siguientes párrafos se incluyen los aspectos más destacables del sistema.

- El sistema debe facilitar la gestión y manejo, de ontologías y documentos relacionados con la ingeniería del conocimiento. Su función principal es la reutilización de conocimiento heterogéneo, en cuanto a su dominio.
- La aplicación debe permitir compartir y gestionar representaciones ontológicas. El sistema que se propone permite realizar una representación ontológica de conocimiento expresada mediante distintos formalismos, inicialmente esquemas

de metadatos, para posteriormente, relacionarla con ontologías de referencia, locales y remotas. Asimismo, el sistema ha de permitir que el usuario pueda crear sus propias conceptualizaciones, reutilizando el conocimiento ya incorporado en el sistema. Finalmente, el sistema debe disponer de un buscador/índice que permitirá incorporar nuevos documentos relacionados con el tema sobre el que quiera profundizar el usuario. Además, ha de soportar búsquedas por conceptos y recuperar documentos en los que se traten dichos conceptos.

- No es necesario que el usuario sea un experto en ontologías, o en software de estas características, para poder manejar con soltura la aplicación. Con esto se quiere indicar que la aplicación debe potenciar la usabilidad, a través de interfaces intuitivos y de fácil manejo.

Las principales funcionalidades que proporcionará la aplicación son las siguientes:

- **Gestionar Esquemas:** El sistema deberá proporcionar soporte a la gestión de esquemas, tanto originales como cualificados:
 1. Dar de alta un esquema.
 2. Editar datos de un esquema existente.
 3. Eliminar un esquema existente.
- **Gestionar Ontologías:** El sistema debe soportar la gestión de ontologías, tanto específicas como de referencia, incluyendo sus ontologías de alineamiento asociadas:
 1. Crear una nueva ontología.
 2. Editar datos de una ontología existente.
 3. Eliminar una ontología.
 4. Importación y exportación de ontologías a formato RDF, OWL y esquemas XML.
 5. Consultar ontologías existentes asociadas al concepto de búsqueda.
 6. Alinear ontologías específicas contra ontologías genéricas.
- **Gestionar Perfiles de Aplicación:** El sistema proporcionará soporte a la gestión de perfiles de aplicaciones, incluyendo:
 1. Crear un perfil de aplicación.
 2. Eliminar un perfil de aplicación.
 3. Editar los datos de un perfil de aplicación.
 4. Importar y exportar un perfil de aplicación.
- **Gestionar Documentos:** La gestión de documentos incluirá las siguiente funcionalidades:
 1. Consultar documentos en los que aparece un concepto determinado, o bien que estén clasificados en un dominio o en los que aparezcan un conjunto específico de metadatos.

2. Incluir documentos asociados a un esquema.
 3. Eliminar documentos asociados a un esquema.
- **Gestionar Usuarios:** Funcionalidad relacionada con la administración del sistema, concretamente, con la gestión de usuarios y perfiles de usuario. El sistema deberá permitir:
 1. Crear perfiles de usuario.
 2. Eliminar perfiles de usuario.
 3. Modificar perfiles de usuario.
 4. Asignar perfiles a usuarios.
 5. Dar de alta a nuevos usuarios.
 6. Dar de baja a antiguos usuarios.
 7. Modificar datos de usuarios registrados en el sistema.
 8. Validar a usuarios que se registran en la aplicación.
 - **Consultar ayuda:** En cualquier momento de interacción con el sistema, se debe tener la posibilidad de consultar la ayuda que ofrece la aplicación sobre los temas que puedan resultar más complicados, en cuanto al manejo de ésta.

Esta funcionalidad se resume gráficamente en el siguiente diagrama general de casos de uso.

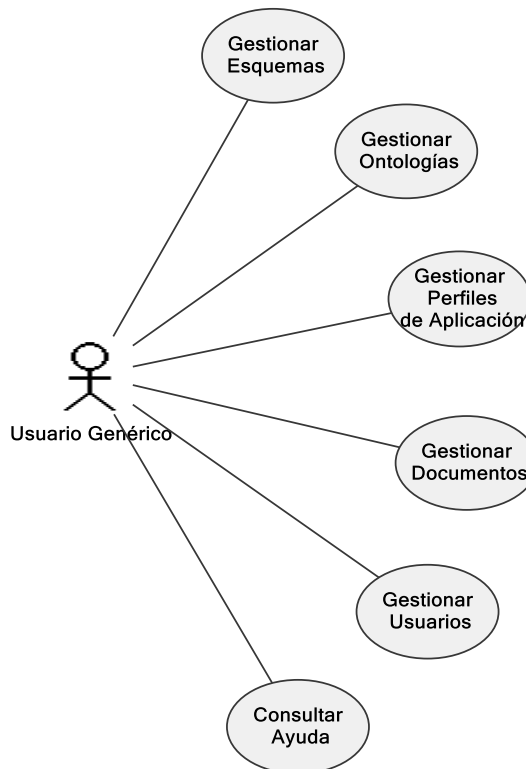


Figura III-12: Diagrama inicial de Casos de Uso

Existen varios tipos de usuarios que pueden interactuar con la aplicación. Cada tipo de usuario representa un rol en el sistema. Se definen a continuación:

- **Usuario:** Usuario de la aplicación cuya función está limitada, básicamente, a realizar consultas de tipo sintáctica, conceptual, por metadatos, por ontología y por formalismos soportados, sobre documentos y ontologías específicas o de referencia del sistema. También tendrá la posibilidad de subir sus propios esquemas para que, una vez revisados, sean incorporados al sistema para su consulta y reutilización.
- **Administrador:** Persona encargada de la gestión de usuarios en la aplicación, de la asignación de permisos y perfiles a éstos, y de la validación de los usuarios que pretendan trabajar con el sistema.
- **Experto responsable del dominio:** Persona con amplios conocimientos de un dominio en concreto. Los expertos representan y validan los conceptos de las ontologías relacionadas con su dominio.
- **Experto del dominio:** Es la persona que está al cargo de los expertos del dominio. Los conceptos más importantes sobre el dominio son validados por este experto. Es quien decide en caso de no existir consenso entre los expertos del dominio.
- **Ingeniero responsable del dominio:** Es un ingeniero en informática destacado en el dominio. Ayuda a los expertos del dominio en sus tareas. Combina el conocimiento del dominio con capacidades de representación de la Ingeniería Ontológica. Su función se basa en coordinar y actuar como puente de comunicación entre ingenieros del dominio y expertos del dominio.
- **Ingeniero del dominio:** El encargado de gestionar y mantener la integridad de la ontología de referencia.
- **Diseñador del dominio:** Usuario que abstrae en el sistema tanto a Ingenieros como a Expertos.

En la siguiente figura se incluye la representación gráfica de la jerarquía de usuarios.

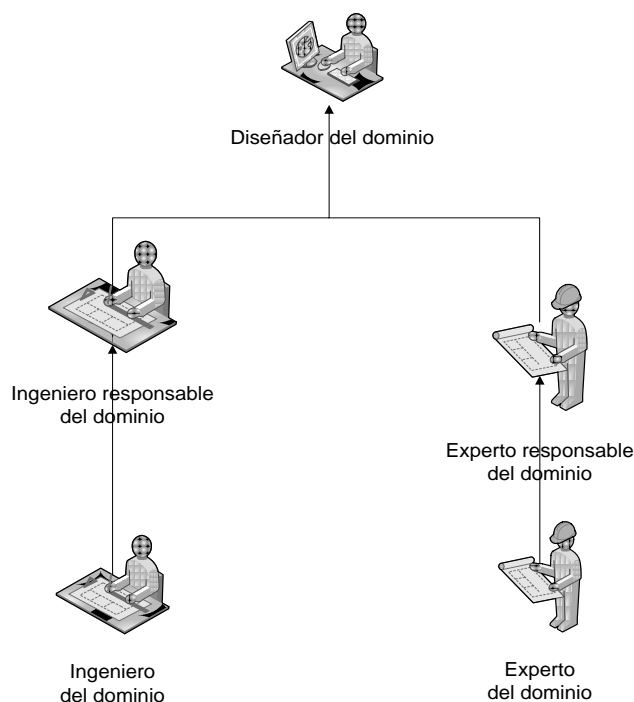


Figura III-13: Representación gráfica de la jerarquía de usuarios

Una vez obtenida la primera descripción de funcionalidad y roles del sistema, se estableció, por su complejidad y riesgo tecnológico, un ciclo de vida en espiral para el proceso de desarrollo. El sistema incluye una gran cantidad de funcionalidades que pueden ser desarrolladas de forma independiente e incremental, además, existían riesgos derivados del desconocimiento de tecnologías existentes para la gestión de ontologías. Mediante el uso de un ciclo de vida en espiral, es posible acercar el sistema a la solución de forma progresiva, permitiendo el desarrollo paralelo y la especialización del equipo de desarrollo en función del componente a desarrollar.

Una vez establecido el ciclo de vida del proceso de desarrollo, se decidió realizar una primera especificación de requisitos completa, para minimizar riesgos derivados de realizar especificaciones de requisitos incrementales tales como: aparición de nuevos requisitos, incompatibilidad de los nuevos requisitos con los existentes, etc., que pueden llevar al rediseño de partes ya desarrolladas. De este modo, se desarrolló una primera especificación de todo el sistema, con el fin de obtener una visión completa y más detallada de las necesidades que debía cubrir. Si bien la aparición de nuevos requisitos es inevitable, al menos se reduce su número y se maximiza la estabilidad de los obtenidos. El resumen de la especificación de requisitos del sistema se muestra en el siguiente apartado.

3.5.2 Especificación de requisitos de usuario

El objetivo de la fase de especificación de requisitos de usuario es elicitación del conjunto de requisitos que comprenden las capacidades que debe satisfacer el sistema a desarrollar. El conjunto de requisitos comprende las necesidades que la presente propuesta plantea de cara a ser soportada por un sistema informático.

Por su interés en el desarrollo del presente y futuros sistemas de soporte, en este apartado se presenta un resumen de los requisitos obtenidos tras la consecución de esta fase. La especificación completa puede consultarse en (Ayensa, I. et al., 2010a).

Los requisitos mostrados a continuación, en forma de casos de uso, se ha priorizado según la necesidad e importancia de los mismos, dada la extensión del proyecto. Los distintos grados de prioridad establecidos son: **alta, media y baja**. El objetivo de la asignación de prioridades a los requisitos es establecer el orden en el cual se implementarán. De este modo, el orden de implementación se establecerá de mayor a menor prioridad.

Los requisitos se han clasificado para cada tipo de usuario, es decir, por el rol que requiere dicho servicio o funcionalidad. Para facilitar su lectura e interpretación, se han incluido diagramas de casos de uso que representan los requisitos descritos textualmente.

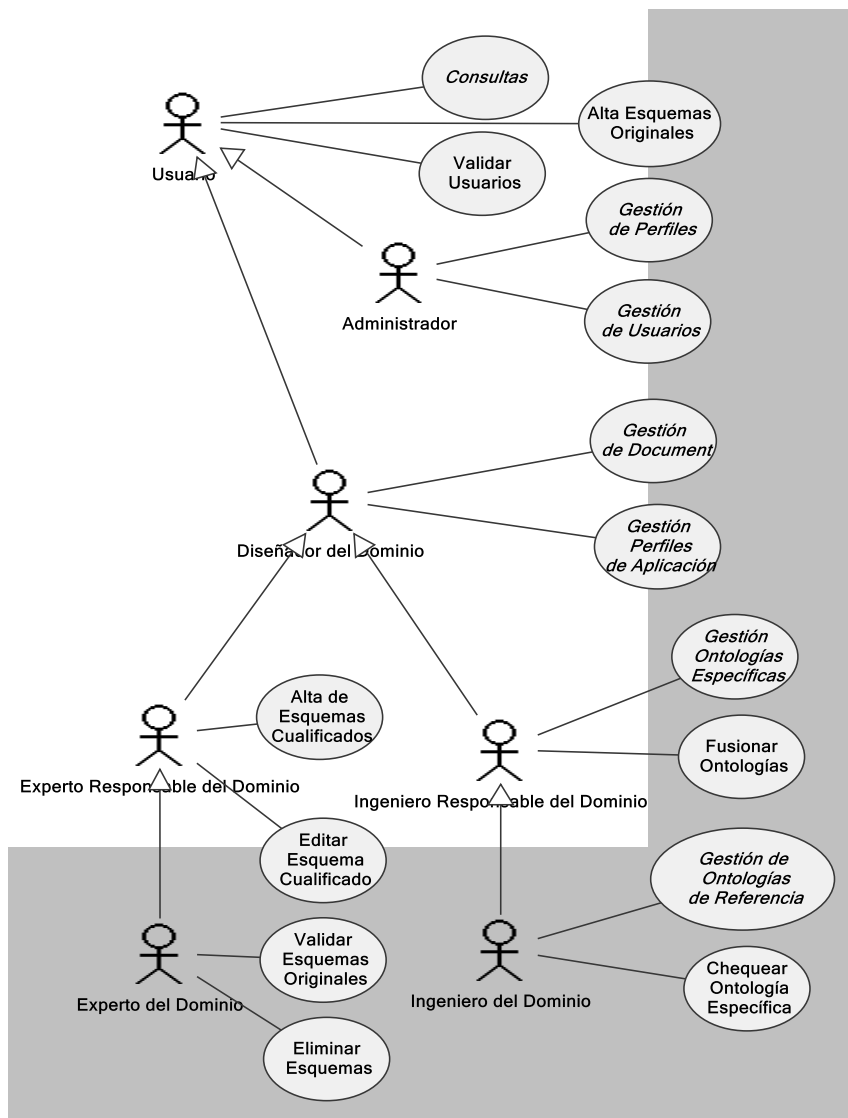


Figura III-14: Diagrama general de Casos de Uso.

3.5.2.1 Usuario

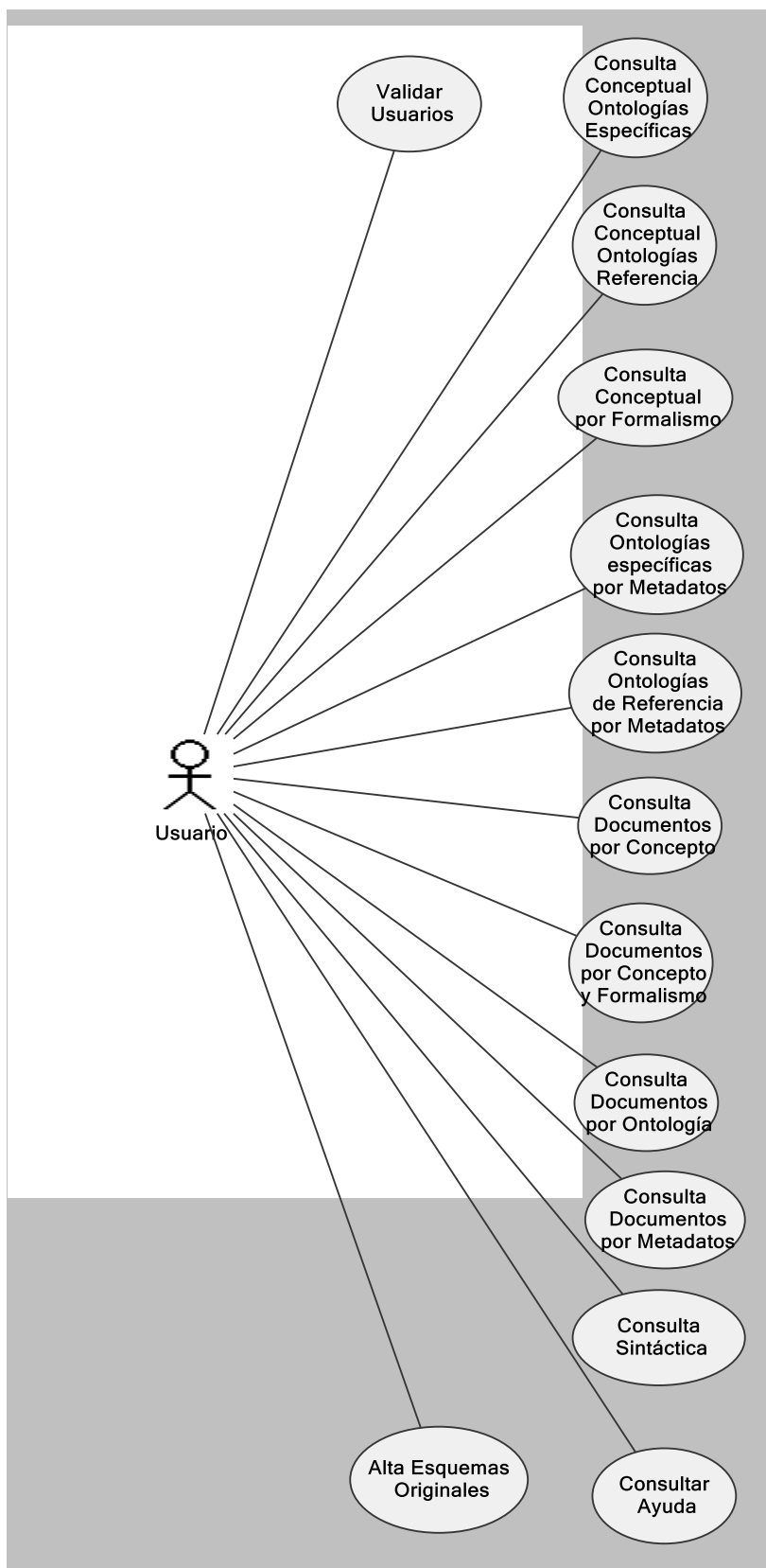


Figura III-15: Diagrama de casos de uso del rol usuario

Nombre	Consulta Conceptual
Identificador	CU01
Actores	Usuario
Objetivo	Obtener la representación del concepto que busca el usuario haciendo uso de la base ontológica. Se mostrará la definición del concepto, las relaciones con los conceptos padre e hijos y conceptos relacionados o sinónimos, además de la referencia a cada ontología específica en la que se haya encontrado.
Precondiciones	
Postcondiciones	
Escenario básico	<ol style="list-style-type: none"> 1. Introducir concepto 2. Realizar búsqueda sintáctica en ontologías específicas 3. Mostrar resultados intermedios 4. Seleccionar concepto 5. Buscar conceptos relacionados en la ontología de referencia 6. Buscar conceptos relacionados en las ontologías específicas 7. Mostrar resultados
Prioridad	Alta

Nombre	Consulta Conceptual por Formalismo
Identificador	CU02
Actores	Usuario
Objetivo	Obtener la representación del concepto que busca el usuario haciendo uso de la base ontológica y que además se encuentran en el formalismo especificado. Similar a la Consulta Conceptual pero tomando como entrada, además, el formalismo.
Precondiciones	
Postcondiciones	
Escenario básico	<ol style="list-style-type: none"> 1. Introducir concepto y formalismo 2. Realizar búsqueda sintáctica en ontologías específicas 3. Mostrar resultados intermedios 4. Seleccionar concepto 5. Buscar conceptos relacionados en la ontología de referencia 6. Buscar conceptos relacionados en las ontologías específicas 7. Mostrar resultados
Prioridad	Baja

Nombre	Consulta por Dominio
Identificador	CU03
Actores	Usuario
Objetivo	Obtener las distintas ontologías específicas cuyo dominio coincida con el introducido por el usuario en la búsqueda.
Precondiciones	
Postcondiciones	
Escenario básico	1. Introducir dominio de búsqueda. 2. Realizar la búsqueda. 3. Mostrar resultados.
Prioridad	Alta

Nombre	Consulta Ontologías específicas por Metadatos
Identificador	CU04
Actores	Usuario
Objetivo	Obtener las distintas ontologías específicas a partir de un conjunto de metadatos facilitados por el usuario en la consulta.
Precondiciones	
Postcondiciones	
Escenario básico	1. Introducir metadatos por los que se desea buscar. 2. Realizar la búsqueda. 3. Mostrar resultados.
Prioridad	Alta

Nombre	Consulta Documentos por Concepto
Identificador	CU05
Actores	Usuario
Objetivo	Poder realizar búsquedas en el sistema por un concepto. Se debe poder encontrar documentos en los que esté aparezca dicho concepto con la acepción adecuada. Tomando como base la Consulta Conceptual, finalmente se realizará una búsqueda del concepto en la base de datos de documentos.

Precondiciones

Postcondiciones

Escenario básico	<ol style="list-style-type: none"> 1. Introducir concepto 2. Realizar búsqueda sintáctica en ontologías específicas 3. Mostrar resultados intermedios 4. Seleccionar concepto 5. Buscar conceptos relacionados en la ontología de referencia 6. Buscar conceptos relacionados en las ontologías específicas 7. Seleccionar concepto 8. Buscar documentos que incluyan el concepto 9. Mostrar resultados
-------------------------	---

Prioridad	Alta
------------------	-------------

Nombre	Consulta Documentos por Concepto y Formalismo
---------------	--

Identificador	CU06
----------------------	-------------

Actores	Usuario
----------------	----------------

Objetivo	Obtener documentos en los que aparezca el concepto con la acepción adecuada y que además, se encuentren en el formalismo especificado. Similar a la Consulta de Documentos por Concepto pero añadiendo como criterio de búsqueda el formalismo.
-----------------	--

Precondiciones

Postcondiciones

Escenario básico	<ol style="list-style-type: none"> 1. Introducir concepto y formalismo 2. Realizar búsqueda sintáctica en ontologías específicas 3. Mostrar resultados intermedios 4. Seleccionar concepto 5. Buscar conceptos relacionados en la ontología de referencia 6. Buscar conceptos relacionados en las ontologías específicas 7. Seleccionar concepto 8. Buscar documentos que incluyan el concepto 9. Mostrar resultados
-------------------------	--

Prioridad	Baja
------------------	-------------

Nombre	Consulta Documentos por Ontología
---------------	--

Identificador	CU07
----------------------	-------------

Actores	Usuario
----------------	----------------

Objetivo	Obtener los documentos asociados a una ontología.
-----------------	--

Precondiciones

Postcondiciones

Escenario básico	<ol style="list-style-type: none">1. Introducir dominio de búsqueda2. Realizar la búsqueda3. Mostrar resultados4. Se selecciona la ontología de la cual se quiere extraer la información5. Se devuelven todos los documentos asociados a la ontología en cuestión
-------------------------	---

Prioridad	Alta
------------------	-------------

Nombre	Consulta Sintáctica
---------------	----------------------------

Identificador	CU08
----------------------	-------------

Actores	Usuario
----------------	----------------

Objetivo	Obtener conceptos de las ontologías específicas en base a una consulta sintáctica del término introducido por el usuario.
-----------------	--

Precondiciones

Postcondiciones

Escenario básico	<ol style="list-style-type: none">1. Introducir término de búsqueda2. Realiza la búsqueda sintáctica sobre las ontologías específicas3. Mostrar resultados
-------------------------	--

Prioridad	Alta
------------------	-------------

Nombre	Alta Esquemas Originales
---------------	---------------------------------

Identificador	CU35
----------------------	-------------

Actores	Usuario
----------------	----------------

Objetivo	El usuario podrá subir esquemas originales que serán incorporados una vez de validados.
-----------------	--

Precondiciones

Postcondiciones

Escenario básico	<ol style="list-style-type: none">1. Seleccionar el fichero o URL del esquema a subir.2. Confirmar operación.
-------------------------	--

Prioridad	Alta
------------------	-------------

Nombre	Validar Usuarios
Identificador	CU37
Actores	Usuario
Objetivo	Validar a un usuario registrado en el sistema según el/los perfiles que tenga asignados.
Precondiciones	
Postcondiciones	Usuario validado.
Escenario básico	<ol style="list-style-type: none"> 3. Introducir credenciales 4. Realizar la validación
Prioridad	Alta

Nombre	Consultar Ayuda.
Identificador	CU09
Actores	Usuario
Objetivo	El usuario podrá consultar la ayuda para resolver las dudas que le puedan surgir durante la interacción con el sistema. Se proporcionará un mecanismo de ayuda que pueda servir como guía en determinadas operaciones que se puedan realizar con la aplicación.
Precondiciones	
Postcondiciones	
Escenario básico	<ol style="list-style-type: none"> 1. Seleccionar la sección de ayuda del sistema 2. Consultar el área sobre el que desee obtener información
Prioridad	Alta

3.5.2.2 Diseñador del Dominio

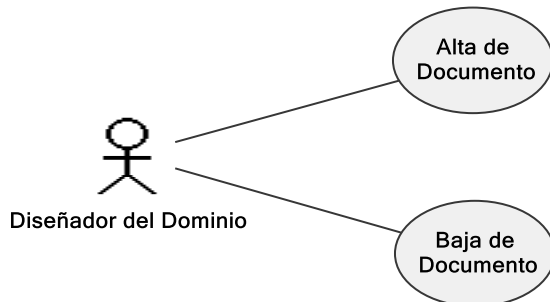


Figura III-16: Diagrama de gestión de documentos

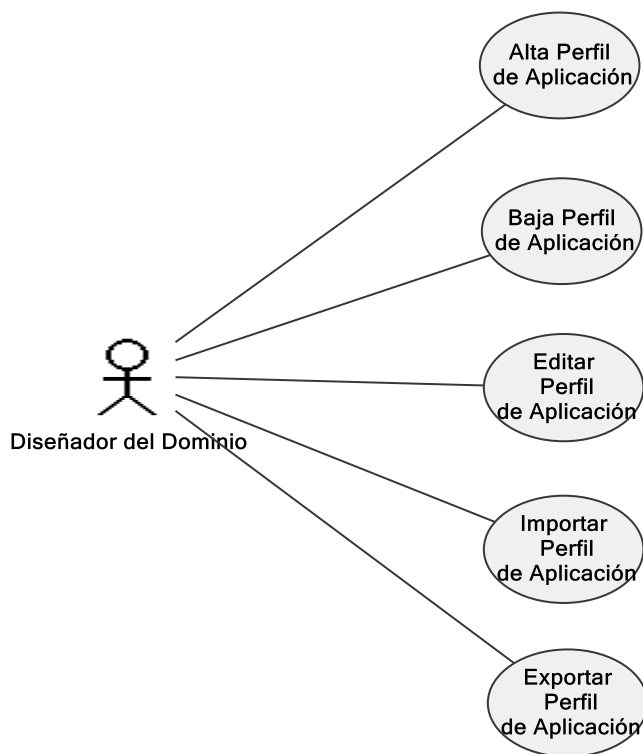


Figura III-17: Diagrama de gestión de perfiles de aplicación

Nombre	Alta de Documento.
Identificador	CU31
Actores	Diseñador del Dominio
Objetivo	Poder dar de alta nuevos documentos.
Precondiciones	
Postcondiciones	Se generarán las instancias correspondientes al documento.
Escenario básico	1. Seleccionar la ontología que describe el documento 2. Enviar petición al buscador/indizador
Prioridad	Media

Nombre	Baja de Documento.
Identificador	CU32
Actores	Diseñador del Dominio
Objetivo	Poder eliminar documentos, borrando al mismo tiempo las referencias entre el documento y los términos a los que estuviera asociado.
Precondiciones	Existe, al menos, un documento dado de alta.
Postcondiciones	Las instancias derivadas del documento son eliminadas.
Escenario básico	1. Seleccionar la ontología que describe el documento 2. Enviar petición de baja al buscador/indizador
Prioridad	Media

Nombre	Alta Perfil de Aplicación
Identificador	CU15
Actores	Diseñador del Dominio
Objetivo	Dar de alta un perfil de aplicación en el sistema.
Precondiciones	
Postcondiciones	El perfil de aplicación queda almacenado en el sistema.
Escenario básico	1. Registrar el nuevo perfil de aplicación 2. Seleccionar conceptos a incluir en el perfil de aplicación 3. Guardar cambios.

Prioridad	Media
Nombre	Baja Perfil de Aplicación
Identificador	CU16
Actores	Diseñador del Dominio
Objetivo	Dar de baja un perfil de aplicación en el sistema.
Precondiciones	Existe, al menos, un perfil de aplicación dado de alta.
Postcondiciones	El perfil de aplicación se elimina del sistema.
Escenario básico	1. Buscar perfil de aplicación 2. Borrar perfil de aplicación 3. Guardar cambios
Prioridad	Media
Nombre	Editar Perfil de Aplicación
Identificador	CU17
Actores	Diseñador del Dominio
Objetivo	Poder modificar los datos referentes a un determinado perfil de aplicación.
Precondiciones	Existe, al menos, un perfil de aplicación dado de alta
Postcondiciones	El perfil de aplicación queda actualizado con los nuevos datos.
Escenario básico	1. Buscar perfil de aplicación 2. Realizar cambios sobre dicho perfil 3. Guardar cambios
Prioridad	Media
Nombre	Importar Perfil de Aplicación
Identificador	CU18
Actores	Diseñador del Dominio
Objetivo	Poder importar un perfil de aplicación al sistema.
Precondiciones	El perfil de aplicación ha sido dado de alta.

Postcondiciones	El perfil se importa al sistema.
Escenario básico	<ol style="list-style-type: none"> 1. Buscar perfil de aplicación 2. Seleccionar fichero importación o indicar URL 3. Importar un perfil de aplicación
Prioridad	Media

Nombre	Exportar Perfil de Aplicación
Identificador	CU19
Actores	Diseñador del Dominio
Objetivo	Poder exportar un perfil de aplicación del sistema.
Precondiciones	Existe, al menos, un perfil de aplicación.
Postcondiciones	
Escenario básico	<ol style="list-style-type: none"> 1. Buscar perfil de aplicación 2. Seleccionar el perfil a exportar 3. Seleccionar el formato de exportación y ubicación del fichero
Prioridad	Media

3.5.2.3 Experto Responsable del Dominio

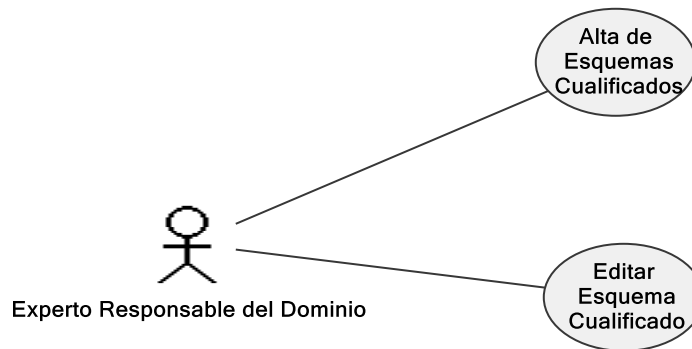


Figura III-18: Diagrama de casos de uso del rol Experto Responsable del Dominio

Nombre	Alta de Esquemas Cualificados
Identificador	CU36
Actores	Experto Responsable del Dominio
Objetivo	Poder dar de alta un esquema cualificado en el sistema.

Precondiciones	Existe un esquema original al que asociar el esquema cualificado.
Postcondiciones	
Escenario básico	<ol style="list-style-type: none"> 1. Introducir fichero o URL del esquema cualificado a incorporar 2. Seleccionar esquema original al que asociar el esquema 3. Confirmar operación
Prioridad	Alta

Nombre	Editar Esquema Cualificado
Identificador	CU37
Actores	Experto Responsable del Dominio
Objetivo	Poder modificar los datos referentes a un determinado esquema cualificado.
Precondiciones	Existe, al menos, un esquema cualificado dado de alta
Postcondiciones	El esquema cualificado queda actualizado con los nuevos datos.
Escenario básico	<ol style="list-style-type: none"> 1. Buscar esquema cualificado 2. Realizar cambios sobre dicho esquema 3. Guardar cambios
Prioridad	Media

3.5.2.4 Experto del Dominio

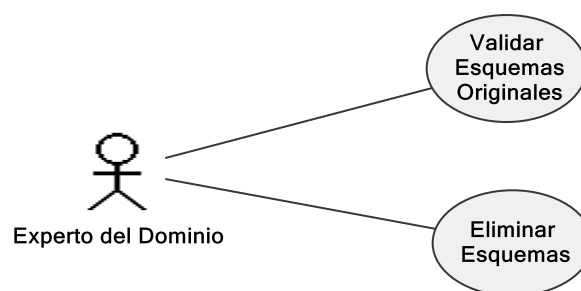


Figura III-19: Diagrama de casos de uso del rol Experto del Dominio

Nombre	Validar Esquemas Originales
Identificador	CU38
Actores	Experto del Dominio
Objetivo	Poder validar un esquema subido por un usuario.

Precondiciones	Existe un esquema original pendiente de validación.
Postcondiciones	El esquema original pasa a estar validado y queda incorporado como un esquema más en el sistema.
Escenario básico	1. Seleccionar esquema original no validado 2. Mostrar esquema seleccionado 3. Confirmar validación
Prioridad	Alta

Nombre	Eliminar Esquemas
Identificador	CU39
Actores	Experto del Dominio
Objetivo	Poder eliminar un esquema, original o cualificado.
Precondiciones	Existen esquemas dados de alta en el sistema.
Postcondiciones	El esquema queda eliminado del sistema, incluyendo los esquemas cualificados y ontologías específicas que pueda tener asociado.
Escenario básico	1. Seleccionar esquema a eliminar 2. Mostrar esquema seleccionado 3. Confirmar validación
Prioridad	Alta

3.5.2.5 Ingeniero del Dominio

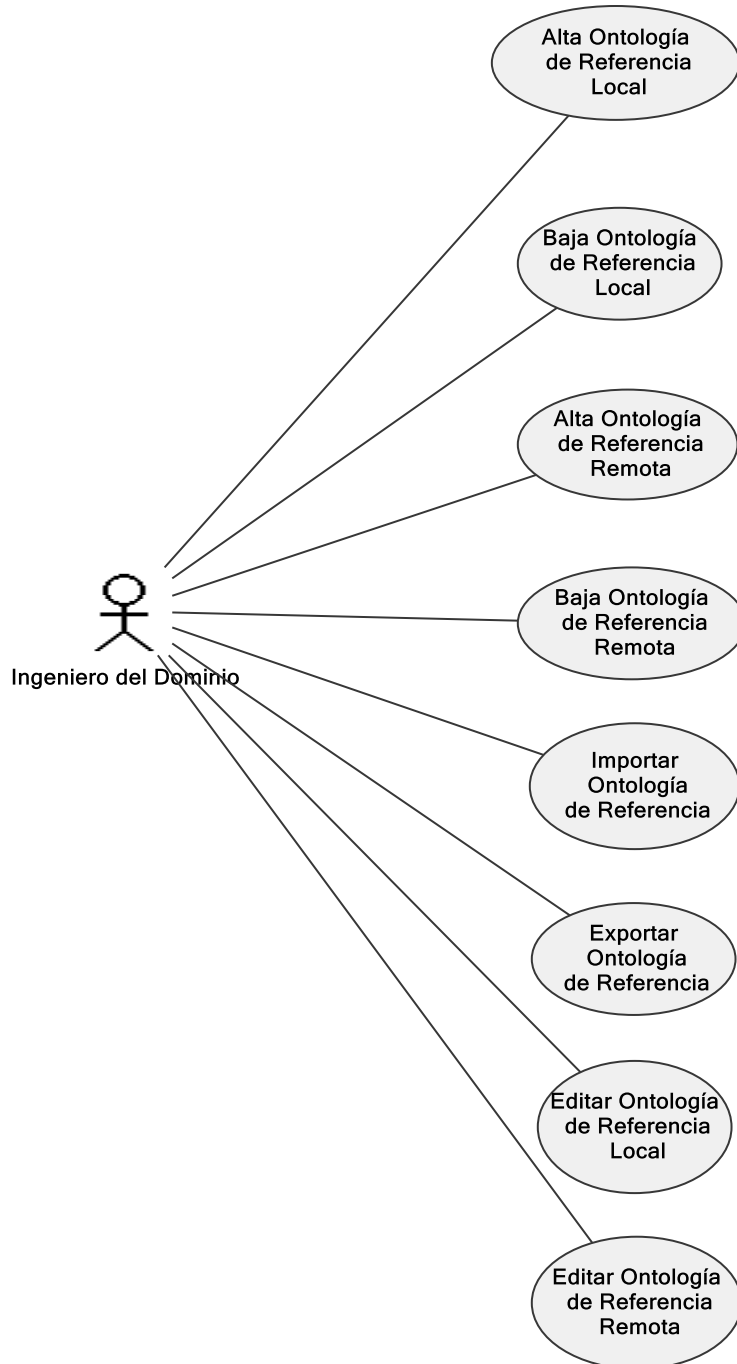


Figura III-20: Diagrama gestión de ontologías de referencia

Nombre	Alta Ontología de Referencia Local.
Identificador	CU22
Actores	Ingeniero del Dominio
Objetivo	Poder dar de alta una ontología de referencia local al sistema.
Precondiciones	
Postcondiciones	Se almacena una nueva ontología de referencia local al sistema, incluyendo sus ontologías de alineamiento asociadas.
Escenario básico	<ol style="list-style-type: none"> 1. Indicar ubicación de la ontología de referencia 2. Indicar ubicación de la ontología de alineamiento interno 3. Indicar ubicación de la ontología de alineamiento externo 4. Guardar cambios
Prioridad	Alta

Nombre	Baja Ontología de Referencia Local.
Identificador	CU23
Actores	Ingeniero del Dominio
Objetivo	Poder dar de baja una ontología de referencia local al sistema.
Precondiciones	Existe, al menos, una ontología de referencia local.
Postcondiciones	Se elimina una ontología de referencia local al sistema y sus ontologías de alineamiento asociadas.
Escenario básico	<ol style="list-style-type: none"> 1. Seleccionar la ontología a eliminar 2. Borrar la ontología seleccionada y sus ontologías de alineamiento 3. Guardar cambios
Prioridad	Alta

Nombre	Alta Ontología de Referencia Remota.
Identificador	CU24
Actores	Ingeniero del Dominio
Objetivo	Poder dar de alta una ontología de referencia remota.
Precondiciones	

Postcondiciones	Se almacena un enlace a la nueva ontología de referencia remota.
Escenario básico	<ol style="list-style-type: none"> 1. Indicar la URL de la ontología de referencia 2. Indicar la URL de la ontología de alineamiento interno 3. Indicar la URL de la ontología de alineamiento externo 4. Guardar cambios.
Prioridad	Alta
<hr/>	
Nombre	Baja Ontología de Referencia Remota.
Identificador	CU25
Actores	Ingeniero del Dominio
Objetivo	Poder dar de baja una ontología de referencia remota.
Precondiciones	Existe, al menos, una ontología de referencia remota.
Postcondiciones	Se elimina una ontología de referencia remota, es decir, la URL que hace de enlace a la ontología de referencia. También se eliminarán sus ontologías de alineamiento asociadas.
Escenario básico	<ol style="list-style-type: none"> 1. Seleccionar la ontología a eliminar 2. Borrar el enlace a la ontología y las ontologías de alineamiento asociadas 3. Guardar cambios
Prioridad	Alta
<hr/>	
Nombre	Importar Ontología de Referencia.
Identificador	CU26
Actores	Ingeniero del Dominio
Objetivo	Poder importar una ontología de referencia en el sistema.
Precondiciones	La ontología de referencia se ha dado de alta como ontología local.
Postcondiciones	Se actualiza el contenido de la ontología de referencia local al sistema.
Escenario básico	<ol style="list-style-type: none"> 1. Buscar ontología de referencia. 2. Seleccionar la fuente de la ontología a importar, por ejemplo desde “fichero”. 3. Actualizar metadatos de la nueva ontología creada.
Prioridad	Alta

Nombre	Exportar Ontología de Referencia.
Identificador	CU27
Actores	Ingeniero del Dominio
Objetivo	Poder exportar una ontología de referencia del sistema.
Precondiciones	Existe, al menos, una ontología de referencia local.
Postcondiciones	
Escenario básico	<ol style="list-style-type: none"> 1. Buscar ontología. 2. Seleccionar la ontología a exportar. 3. Seleccionar el formato de exportación y ubicación, por ejemplo, “fichero”.
Prioridad	Alta

Nombre	Editar Ontología de Referencia Local.
Identificador	CU28
Actores	Ingeniero del Dominio
Objetivo	Poder editar cualquier elemento de la ontología de referencia del sistema.
Precondiciones	Existe, al menos, una ontología de referencia local.
Postcondiciones	La ontología queda almacenada con las modificaciones realizadas.
Escenario básico	<ol style="list-style-type: none"> 1. Seleccionar la ontología de referencia. 2. Editar elementos de la ontología. 3. Guardar cambios.
Prioridad	Alta

Nombre	Editar Ontología de Referencia Remota.
Identificador	CU29
Actores	Ingeniero del Dominio
Objetivo	Poder cambiar la URL que referencia a la ontología remota o de las ontologías de alineamiento asociadas.
Precondiciones	Existe, al menos, una ontología de referencia remota.
Postcondiciones	La URL queda modificada.

Escenario básico	<ol style="list-style-type: none">1. Seleccionar la ontología de referencia.2. Realizar cambios en las URLs.3. Guardar cambios.
Prioridad	Alta
Nombre	Chequear Ontología Específica.
Identificador	CU30
Actores	Ingeniero del Dominio
Objetivo	Permitir la comprobación de los conceptos y sus descripciones incluidos en una ontología específica. En caso de ser necesario, deberá permitirse la edición de dicha ontología. La comprobación debe permitir consultar los valores de los atributos, incluyendo los significados asociados a la ontología de referencia.
Precondiciones	Existe, al menos, una ontología específica.
Postcondiciones	
Escenario básico	<ol style="list-style-type: none">1. Seleccionar la ontología específica.2. Realizar la comprobación:<ol style="list-style-type: none">a. Consultar valores de atributos.b. Consultar significados asociados.3. Chequear si todos los conceptos están alineados contra la ontología de referencia.
Prioridad	Alta

3.5.2.6 Ingeniero Responsable del Dominio

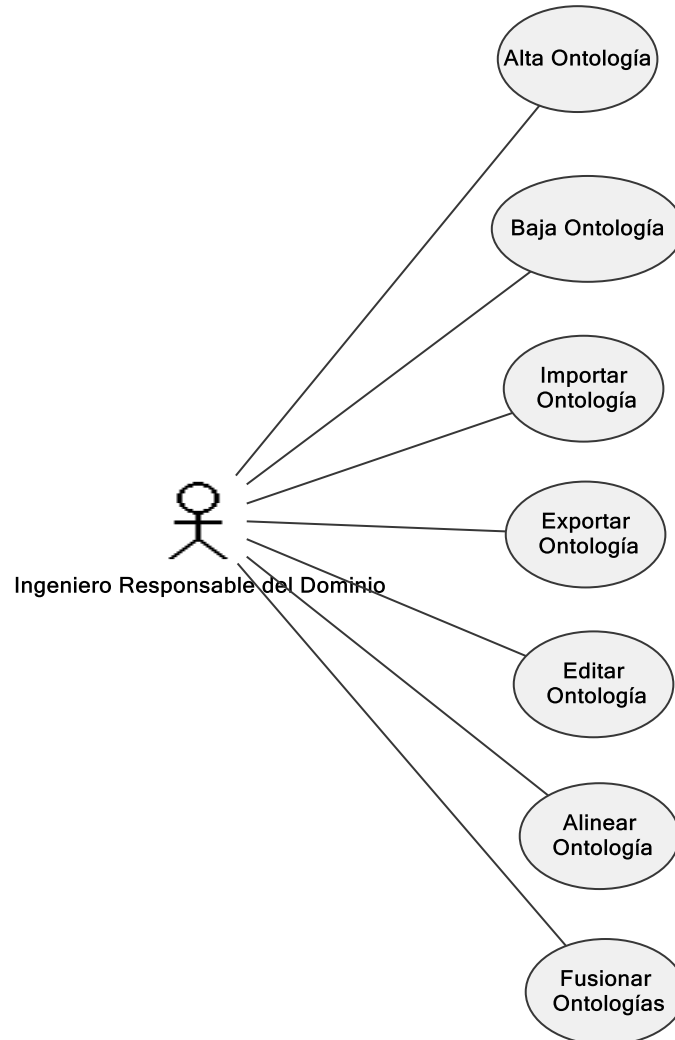


Figura III-21: Diagrama de casos de uso del rol Ingeniero Responsable del Dominio

Nombre	Alta Ontología
Identificador	CU10
Actores	Ingeniero Responsable del Dominio
Objetivo	Dar de alta una ontología específica en el sistema.
Precondiciones	Debe existir un esquema validado pendiente de asignación.
Postcondiciones	La ontología queda almacenada en el sistema y asociada a un esquema original validado.

Escenario básico	<ol style="list-style-type: none"> 1. Seleccionar el fichero o URL de la ontología 2. Seleccionar el esquema original validado al que asociar la ontología 3. Guardar cambios
-------------------------	--

Prioridad	Alta
------------------	------

Nombre	Baja Ontología
---------------	----------------

Identificador	CU11
----------------------	------

Actores	Ingeniero Responsable del Dominio
----------------	-----------------------------------

Objetivo	Dar de baja una ontología específica en el sistema.
-----------------	---

Precondiciones	Existe, al menos, una ontología dada de alta.
-----------------------	---

Postcondiciones	La ontología se elimina del sistema.
------------------------	--------------------------------------

Escenario básico	<ol style="list-style-type: none"> 1. Buscar ontología 2. Seleccionar ontología 3. Borrar la ontología 4. Guardar cambios
-------------------------	---

Prioridad	Alta
------------------	------

Nombre	Importar Ontología
---------------	--------------------

Identificador	CU12
----------------------	------

Actores	Ingeniero Responsable del Dominio
----------------	-----------------------------------

Objetivo	Poder importar una ontología específica en el sistema.
-----------------	--

Precondiciones	La ontología ha sido dada de alta
-----------------------	-----------------------------------

Postcondiciones	La ontología se crea y se importa al sistema.
------------------------	---

Escenario básico	<ol style="list-style-type: none"> 1. Buscar ontología 2. Seleccionar la fuente de la ontología a importar, por ejemplo desde fichero 3. Incluir metadatos de la nueva ontología creada
-------------------------	--

Prioridad	Alta
------------------	------

Nombre	Exportar Ontología
---------------	--------------------

Identificador	CU13
----------------------	------

Actores	Ingeniero Responsable del Dominio
Objetivo	Poder exportar una ontología específica del sistema.
Precondiciones	Existe, al menos, una ontología dada de alta.
Postcondiciones	
Escenario básico	<ol style="list-style-type: none"> 1. Buscar ontología 2. Seleccionar la ontología a exportar 3. Seleccionar el formato de exportación y ubicación, por ejemplo, “fichero”
Prioridad	Alta

Nombre	Editar Ontología
Identificador	CU14
Actores	Ingeniero Responsable del Dominio
Objetivo	Poder editar una ontología específica del sistema.
Precondiciones	Existe, al menos, una ontología dada de alta.
Postcondiciones	La ontología queda almacenada con las modificaciones realizadas.
Escenario básico	<ol style="list-style-type: none"> 1. Seleccionar una ontología 2. Editar la ontología 3. Guardar cambios
Prioridad	Alta

Nombre	Alinear Ontología
Identificador	CU20
Actores	Ingeniero Responsable del Dominio
Objetivo	Relacionar los conceptos de la ontología específica con los de la ontología de referencia. Cuando se trata de una ontología específica, las correspondencias se almacenan en la ontología de alineamiento y son editables.
Precondiciones	Existe, al menos, una ontología específica y una ontología de referencia.
Postcondiciones	La ontología específica se alinea en el sistema.

Escenario básico	<ol style="list-style-type: none"> 1. Seleccionar la ontología a alinear 2. Seleccionar la ontología de referencia 3. Realizar el proceso de correspondencia
Prioridad	Alta
<hr/>	
Nombre	Fusionar Ontologías
Identificador	CU21
Actores	Ingeniero Responsable del Dominio
Objetivo	Poder unir varias ontologías específicas en una sola a través de la de referencia.
Precondiciones	Deben existir, al menos, dos ontologías para poder realizar la fusión.
Postcondiciones	De la fusión surge una nueva ontología en el sistema.
Escenario básico	<ol style="list-style-type: none"> 1. Seleccionar las ontologías a fusionar 2. Seleccionar la ontología de referencia 3. Realizar el proceso de equivalencia de conceptos 4. Definir la información sobre la nueva ontología agregada (metadatos)
Prioridad	Baja

3.5.2.7 Administrador

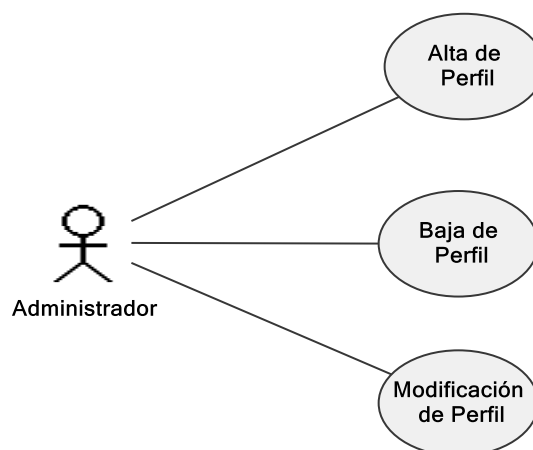


Figura III-22: Diagrama de gestión de perfiles

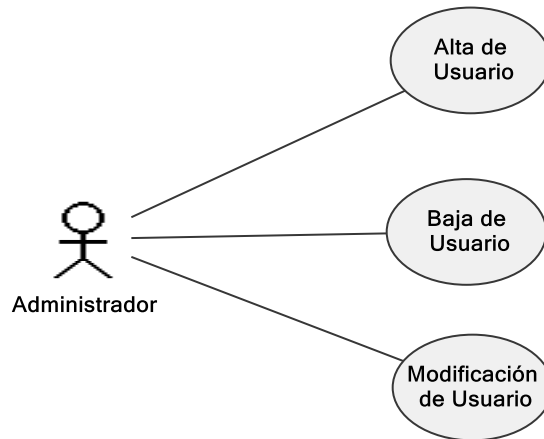


Figura III-23: Diagrama de gestión de usuarios

Nombre	Gestionar Perfiles
Identificador	CU35
Actores	Administrador
Objetivo	Gestionar los distintos perfiles, poder agregar, modificar o borrar perfiles. Igualmente, engloba la asignación de permisos a cada perfil para acceder a ciertas funcionalidades.
Precondiciones	
Postcondiciones	El sistema registra las actualizaciones realizadas sobre perfiles.
Escenario básico	<ol style="list-style-type: none"> 1. Seleccionar el perfil a crear, borrar o modificar 2. Asignarle la nueva funcionalidad o quitársela al perfil en cuestión 3. Guardar cambios
Prioridad	Media

Nombre	Gestionar Usuarios
Identificador	CU36
Actores	Administrador
Objetivo	Dar de alta a nuevos usuarios, eliminar usuarios ya registrados y/o modificar sus datos en el sistema.
Precondiciones	
Postcondiciones	Nuevo usuario almacenado o eliminado del sistema.

Escenario básico	1. Dar de alta o de baja un usuario 2. Asignar perfil al usuario 3. Guardar cambios
Prioridad	Alta

3.5.3 Especificación de requisitos software

El objetivo de la fase de especificación de requisitos software es, partiendo del conjunto de requisitos de usuario obtenido en la fase anterior, generar un conjunto de requisitos software dirigido a los desarrolladores del sistema. Con este fin: el conjunto de requisitos será descrito con mayor detalle; será revisado para eliminar inconsistencias y ambigüedades; será completado con aquellos requisitos que surjan de la especificación con mayor detalle de los requisitos de usuario; y será expresado utilizando un lenguaje técnico.

Como resultado de esta fase se ha obtenido un conjunto de requisitos software que comprenden la funcionalidad y restricciones que debe satisfacer el sistema de soporte a desarrollar. Asimismo, se realizó un diseño del interfaz del sistema, que comprende: un mapa web del sistema, el conjunto de plantillas de las distintas páginas y el diseño inicial de pantallas en base a las plantillas definidas.

Debido a su extensión, el documento no ha sido incluido en el presente apartado, aunque puede ser consultado en (Ayensa, I. et al., 2010b).

3.5.4 Diseño de la arquitectura

El objetivo de la fase de diseño de la arquitectura, o diseño arquitectónico, es establecer la arquitectura, hardware y software, que mejor satisfaga los requisitos elicitados en las fases anteriores. Se establece de este modo el conjunto de componentes a alto nivel que conformarán el sistema, así como su distribución en los recursos hardware. Junto con el diseño de la arquitectura del sistema, en esta fase se evalúan y seleccionan las tecnologías a utilizar en las fases posteriores del desarrollo del sistema. Así, se toman decisiones referentes al lenguaje de programación a utilizar, componentes, marcos, herramientas, etc.

Aunque el documento generado en esta fase puede consultarse en (Fernández-Tostado, T., 2009), a continuación se resumen las principales características de la arquitectura propuesta.

La arquitectura propuesta hace uso del patrón MVC. De este modo, los componentes desarrollados se han identificado en base a la función que proporcionan: vista, modelo y controlador. Asimismo se ha aplicado el estilo arquitectónico de capas, de modo que la distribución de los componentes se ha organizado en niveles bien definidos. Para la capa de la vista se ha utilizado la tecnología Java Server Faces, en su implementación ICEfaces (IceSoft, 2009), que ofrece una manera sencilla de generar páginas Web con alta usabilidad, mientras que para la capa de infraestructura se usará Jena (Hewlett-Packard Development Company, 2009a), para la gestión de repositorios semánticos e Hibernate (JBoss Community, 2010), como interfaz entre la base de datos y la

aplicación (*Object Relational Mapping*).

Entre la vista y el modelo se sitúa la capa de control. Es precisamente aquí donde se integran los marcos utilizados: JSF y Spring (SpringSource, 2010), para la gestión de componentes.

Por último, para la capa de persistencia se utilizará, como Sistema Gestor de Bases de Datos, PostgreSQL (PostgreSQL, 2010), que se ajusta al volumen de datos y rendimiento que se prevé para esta aplicación. En la siguiente figura se muestra el particionado en capas de la aplicación y las tecnologías elegidas para cada capa.



Figura III-24: Arquitectura de la aplicación y tecnologías utilizadas

La aplicación queda desplegada en el servidor de aplicaciones Glasfish v2 (Java.net, 2010), relacionándose cada capa de la arquitectura según el patrón MVC implementado. Así las vistas se vinculan con la capa de control a través del marco JSF, que controla las peticiones del usuario. De igual manera la capa de control está enlazada con todas las clases de acción. Dichas clases delegan la petición del usuario a la capa de negocio, que a su vez interactúa con la capa de infraestructura, donde se encuentran las clases que encapsulan la información que permite la gestión de los esquemas y ontologías a través de los marcos de Jena e Hibernate. Desde estas clases, que conforman el modelo del dominio, se accede a la Base de Datos (PostgreSQL) para operaciones de inserción, borrado y consulta.

Finalmente, destacar que el lenguaje de programación seleccionado ha sido Java (Sun Microsystems, 2010), tanto por sus características como por la tecnología de los componentes seleccionados en la arquitectura, y como Entorno Integrado de Desarrollo se ha utilizado NetBeans 5.5 (Oracle, 2010).

3.5.5 Diseño detallado, codificación y pruebas

Como parte del proyecto del Plan Nacional de I+D+i titulado “*Desarrollo de un sistema de Recuperación Conceptual mediante Niveles Semánticos en la representación de*

Esquemas de Metadatos”, y basado en el planteamiento propuesto en esta tesis, se ha generado documentación correspondiente a estas fases, disponible en (Gárate, F.J., 2010).

3.5.6 Desarrollo de infraestructuras de soporte

Dentro del proyecto del Plan Nacional de I+D+i antes mencionado, se están desarrollando también aplicaciones satélite que complementan y cubren aspectos específicos de la propuesta. Entre ellos destacan:

- Araña de búsqueda. Primeramente se ha desarrollado una araña que, apoyándose en distintos buscadores y diccionarios de términos, explora la Web localizando documentos que hacen uso de alguno de los esquemas almacenados en el sistema. Una vez localizados los recupera y almacena, opcionalmente, para su posterior análisis e indexación.
- Analizador/indexador de documentos. Junto con la araña de búsqueda, se está desarrollando un analizador/indexador que, inicialmente, valida los documentos, y a continuación los procesa para su posterior indexación en una base de datos. La información indexada incluye, para cada documento, los elementos que aparecen en él y los relaciona con el esquema que los define. De este modo se puede obtener información acerca de qué documentos incluyen qué elementos, documentos generados a partir de un esquema o, por ejemplo, qué correlación existe entre elementos de distintos esquemas.
- Generador de versiones usables de esquemas. Se está desarrollando una aplicación que, a modo de servicio web, va a permitir generar versiones amigables, esto es, interpretables por personas, de los esquemas y ontologías gestionados por el sistema. Para este fin, se ha seleccionado como tecnología el uso de *JasperReports* (JasperForge, 2009a) y *JasperServer* (JasperForge, 2009b).

Además de los desarrollos anteriores, ha sido necesario implementar diversas infraestructuras de soporte a la coordinación y desarrollo del proyecto, entre ellas destacan:

- Portal del proyecto. Se ha implementado un Sistema de Gestión de Contenidos (CMS) para albergar la web del proyecto (Semantic Metadata Search, 2010b). La creación del portal ha facilitado: la coordinación del grupo, permitiendo la publicación de documentación, organización del proyecto y del grupo; y la promoción del proyecto. Una vez completados, todos los entregables serán publicados en el portal.
- Sistema de Control de Versiones. Se ha implementado un sistema de control de versiones basado en *Subversion* (SVN) (Semantic Metadata Search, 2010c). De este modo se ha contado con los beneficios para el desarrollo colaborativo que aporta la gestión de código mediante este tipo de sistemas.
- Servidor de Aplicaciones. Se ha implementado un servidor de aplicaciones

basado en *Glassfish v2* (Semantic Metadata Search, 2010d). Esta infraestructura permite la publicación de las distintas versiones del sistema y facilita el desarrollo colaborativo a través de un punto único de publicación.

- Servidor de Persistencia. Se ha implementado un servidor de persistencia basado en *PostgreSQL*. De forma similar al anterior, de este modo se ha facilitado el desarrollo de la plataforma y simplificado las tareas de administración y mantenimiento.

3.5.7 Estado actual de la plataforma

Actualmente la plataforma se encuentra en desarrollo. En (Semantic Metadata Search, 2010c) se puede obtener la última versión del código fuente de la aplicación. En (Semantic Metadata Search, 2010d) se puede acceder a la última versión desplegada de la aplicación.

Capítulo IV: Resultados, Evaluación y Discusión

Este capítulo tiene un objetivo doble: primeramente, presentar un resumen de los resultados obtenidos durante la elaboración de la presente tesis, valorando la aportación que realiza en el contexto de la problemática que resuelve; en segundo lugar, evaluar y comparar la propuesta con otros métodos de recuperación.

En el subapartado *4.1 Resultados y discusión*, se muestran los resultados generados a lo largo de las distintas fases de desarrollo de la investigación, procedentes del análisis del contexto del problema y las consiguientes propuestas para su resolución. De este modo se han obtenido, desde requisitos, hasta propuestas metodológicas que son resumidas y discutidas en este apartado. Para facilitar su lectura, se optó por intercalar la discusión, incluyéndose a continuación de cada resultado.

Con el fin de realizar una comparativa de la propuesta con otras alternativas para la recuperación de documentos XML, en el subapartado *4.2 Evaluación y comparativa de la propuesta*, se realiza una valoración cualitativa de la propuesta. Haciendo uso del método de evaluación DESMET (Kitchenham, 1996), se incluirá la evaluación cualitativa de varios métodos candidatos que, junto con la propuesta, serán comparados. Una vez evaluados los métodos, se realizará un análisis y se discutirán los resultados.

4.1 Resultados y discusión

Puesto que los resultados han sido obtenidos en las distintas fases de la investigación, este capítulo se ha organizado siguiendo la estructura de la exposición de la propuesta.

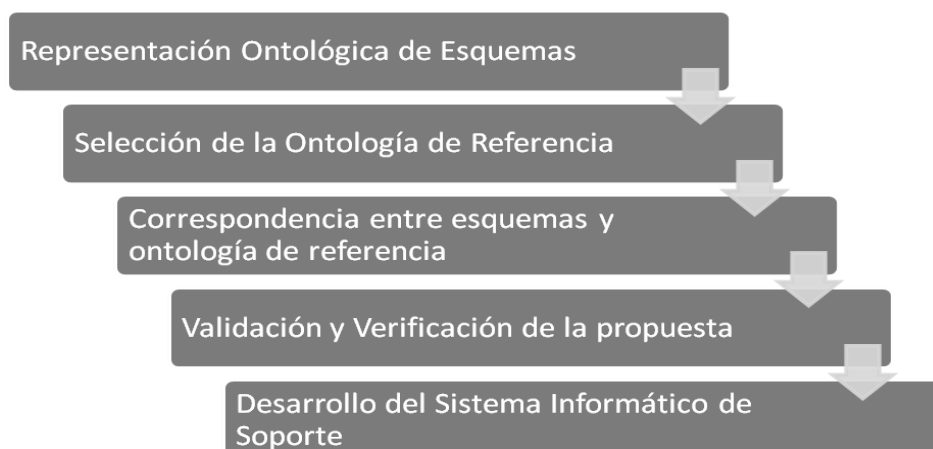


Figura IV-1: Estructura de la exposición de la propuesta y de los resultados obtenidos.

Antes de detallar los resultados de cada fase, en el siguiente esquema se resumen las directrices de los logros conseguidos, tal y como se han expuesto en la descripción de la propuesta:

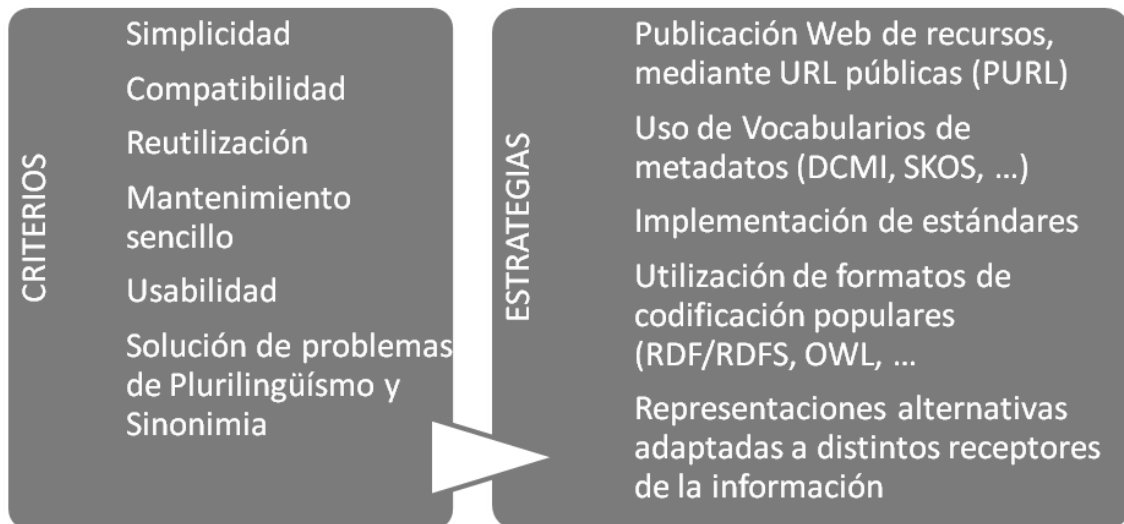


Figura IV-2: Directrices de los logros conseguidos.

4.1.1 Resultados de la representación ontológica de esquemas

Los principales logros de esta fase se resumen en la siguiente figura:

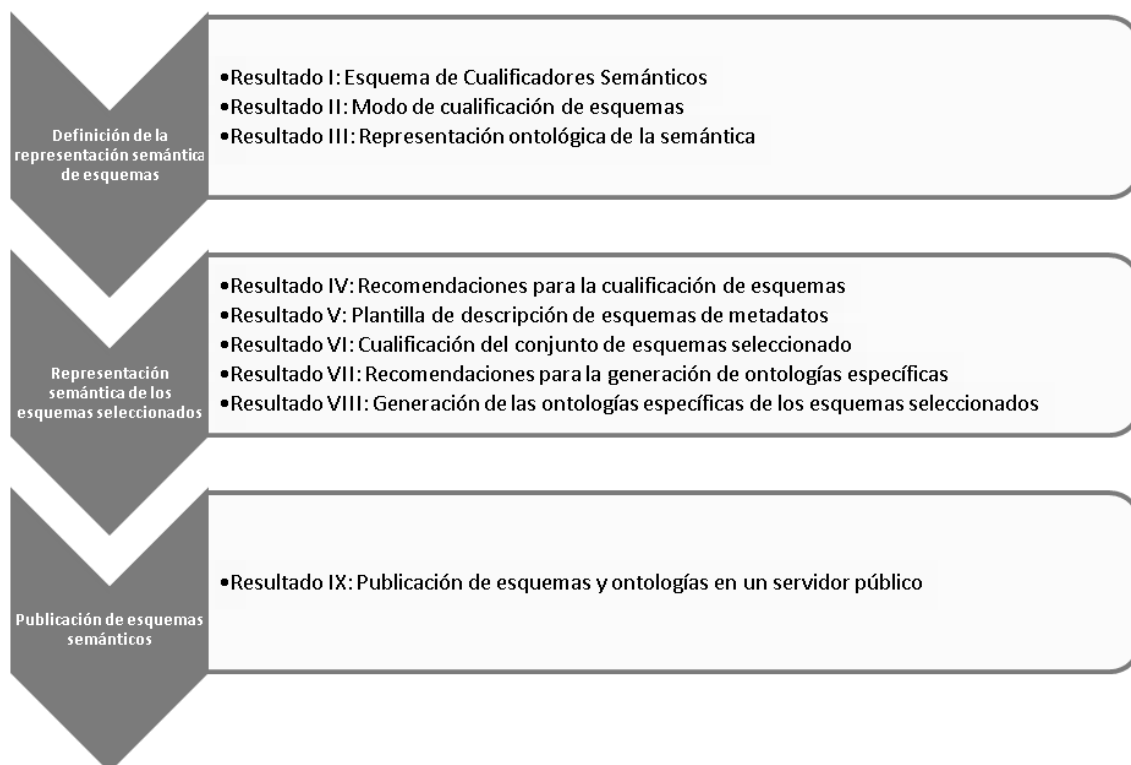


Figura IV-3: Resumen de los logros conseguidos en la fase de representación ontológica de esquemas.

4.1.1.1 Definición de la representación semántica de esquemas

Son varios los resultados obtenidos en esta primera fase de la investigación, relativos a la representación de esquemas y de su semántica.

Resultado I: Esquema de Cualificadores Semánticos: Primeramente, se ha establecido un modo de representación de esquemas que permite la inclusión de la semántica que, a su vez, es definida mediante el uso de un lenguaje capaz de expresar la semántica necesaria en fases posteriores, esto es, que permite la realización de consultas e inferencia. Como resultado se ha definido un esquema de cualificadores semánticos *Semantic Qualifiers Schema*, que alberga el cualificador semántico *hasSemantics*, el cual, como especialización del término *dc:relation*, permite incluir la definición de cada término.

En la siguiente figura se resumen los detalles del resultado:

Resultado I: Esquema de Cualificadores Semánticos

- Esquema de Cualificadores Semánticos
- Cualificador semántico *hasSemantics*
- Lenguaje de representación: RDF

Figura IV-4: Detalles del resultado I: Esquema de Cualificadores Semánticos.

Resultado II: Modo de cualificación de esquemas: También se ha establecido, y ejemplificado, el modo de cualificar esquemas, proponiendo el uso de la especialización de los términos originales y manteniendo su nombre y tipo. En el método propuesto se han seguido las recomendaciones sobre espacios de nombres, nombrado y publicación presentes en estándares y recomendaciones internacionales, así como la conveniencia de generar versiones interpretables por personas y computadoras. Finalmente, se ha seleccionado RDF como lenguaje de definición, en base a su popularidad, capacidad semántica y compatibilidad con el entorno Web.

En la siguiente figura se resumen los detalles del resultado:

Resultado II: Modo de cualificación de esquemas

- Especialización de los términos originales
- Compatibilidad manteniendo: lenguaje, nombre y tipo del término
- Uso de estándares y recomendaciones internacionales
- Versión para personas y computadoras
- Lenguaje de representación computadoras: RDF
- Lenguaje de representación: RDF

Figura IV-5: Detalles del resultado II: Modo de cualificación de esquemas.

La valoración de estos resultados debe comenzar destacando el hecho de haber sido necesaria la definición de nuevos cualificadores que permitan la inclusión de semántica en los esquemas, de forma estructurada, extensible y compatible. Esta definición ha pasado por la evaluación de los métodos de extensión de esquemas y ha tenido como principal requisito la compatibilidad de método con las versiones originales de los esquemas. Así, la ausencia de cualificadores especializados en la inclusión de semántica ha llevado a la definición de un nuevo tipo de cualificador, albergado en un esquema desarrollado a tal efecto. La definición del cualificador como especialización del término *dc:relation*, facilita su compatibilidad con un mayor número de aplicaciones, de modo que éstas puedan hacer uso del término tratándolo como el término del cual proviene, al tiempo que evita la necesidad de modificar el esquema original, en este caso, *DCMI metadata element set*. La definición del esquema específico, para albergar este nuevo cualificador, está contemplada por el DCMI como mecanismo de extensión y, por tanto, garantiza la compatibilidad con el esquema original. Del mismo modo, proporciona un método extensible a través de la inclusión de nuevos cualificadores. Siguiendo también las recomendaciones del DCMI, el esquema ha sido publicado mediante una URL permanente, y el cualificador, que es accesible mediante una URI, sigue las recomendaciones de nombrado para cualificadores.

Sobre el modo de incluir la semántica en los términos, haciendo uso del cualificador propuesto, destacar que se ha seleccionado aquella que maximiza la extensibilidad, compatibilidad y reutilización. Así, la creación de un nuevo esquema con elementos que especializan los términos originales

- Permite la independencia del esquema original, evitando su modificación.
- Facilita la compatibilidad con el esquema propuesto, dado que las aplicaciones tienen la posibilidad de tratar los términos cualificados como el término original que especializan. Además, mantiene el nombre del elemento original, su tipo y su lenguaje de definición.
- Facilita la reutilización haciendo uso de los términos originales y definiéndose de forma independiente al esquema cualificado.
- Proporciona una representación homogénea de los esquemas, lo que facilita su gestión.

Respecto al lenguaje seleccionado para la representación del esquema y los cualificadores, comentar que la decisión se debe a:

- Las capacidades semánticas que ofrece RDF, suficientes para soportar la propuesta.
- Su popularidad y compatibilidad con el procesamiento por computadoras.
- Su grado de estandarización.
- Su compatibilidad con el entorno Web, entorno operacional de la propuesta.

Resultado III: Representación ontológica de la semántica: El siguiente resultado obtenido en esta fase atiende a la necesidad de representar la semántica de los elementos. De este modo establece, y ejemplifica, el uso de *ontologías específicas* que permitan albergar las definiciones de los términos de los esquemas. Dichas ontologías serán instancias de una ontología padre, denominada *SEMSE*, que define los conceptos, relaciones y restricciones necesarios para expresar la semántica de los esquemas. Haciendo uso de lenguajes como OWL, RDFS y SKOS, atiende también a aspectos como plurilingüismo y sinonimia. De nuevo se siguen las recomendaciones sobre espacios de nombres y publicación mediante URLs persistentes.

En la siguiente figura se resumen los detalles del resultado:

Resultado III: Representación ontológica de la semántica

- Definición de la metaontología SEMSE
- Definición de las ontologías específicas instancias de SEMSE
- Lenguajes de representación: OWL, RDFS y SKOS
- Soporte a plurilingüismo y sinonimia
- Uso de estándares y recomendaciones internacionales

Figura IV-6: Detalles del resultado III: Representación ontológica de la semántica.

La valoración de estos resultados debe comenzar por analizar la propuesta de una representación independiente de la semántica. La separación de la semántica del esquema aporta múltiples ventajas

- Permite utilizar un formalismo específico, en este caso una ontología, para la representación semántica, lo cual favorece la extensibilidad y mantenibilidad de la solución.
- Permite el uso de lenguajes con mayor capacidad semántica sin necesidad de alterar el esquema, en línea con el uso de un recurso especializado en la representación semántica.
- La representación especializada permite sacar todo el partido de este formalismo, abordándose aspectos como sinonimia, plurilingüismo o restricciones de los términos.

La desventaja en este caso proviene de la complejidad que implica mantener un recurso más, pero queda compensada con las ventajas anteriormente expuestas.

Respecto a la ontología padre, *SEMSE*, su definición hace uso de lenguajes con alta expresividad semántica OWL, RDFS y SKOS, compatibles con el entorno Web. Esto permite la definición de conceptos, relaciones y restricciones, necesarios para la representación de la semántica de esquemas, el tratamiento de aspectos como la sinonimia o el plurilingüismo, al tiempo que proporcionan soporte a consultas e inferencia.

Respecto a las ontologías específicas, su uso:

- Facilita la mantenibilidad del esquema, al permanecer independiente de la representación de su semántica.
- Es compatible con cualquier esquema.
- Permite cubrir aspectos como la sinonimia y el plurilingüismo.
- Permite el uso de lenguajes con alta capacidad semántica, sin alterar el esquema original.

Para finalizar la valoración de este resultado, destacar que, tanto la ontología *SEMSE* como el proceso de generación de las ontologías específicas, facilitan la reutilización en cuanto a que:

- En su definición reutilizan ontologías y vocabularios existentes.
- Hacen uso de los conceptos de la ontología *SEMSE* en la generación de las ontologías específicas.
- Se encuentran publicadas vía Web para su consulta y reúso.

De nuevos inconvenientes provienen de la complejidad extra de la generación y mantenimiento de un nuevo recurso, no obstante, las ventajas expuestas anteriormente hacen patente la necesidad de una representación especializada e independiente de la semántica de los esquemas.

4.1.1.2 Representación semántica de los esquemas seleccionados

También se han obtenido varios resultados en esta fase, que son resumidos a continuación.

Resultado IV: Recomendaciones para la cualificación de esquemas: En primer lugar se ha elaborado, y ejemplificado, un conjunto de recomendaciones para la cualificación de esquemas. Estas recomendaciones, validadas en publicaciones nacionales e internacionales como: la *Fifth International Semantic Web Conference* (Palacios, V. et al., 2006a), la *Conferência Ibérica de Sistemas e Tecnologias de Informação* (Palacios, V. et al., 2006b) o la *International Conference on Dublin Core and Metadata Applications* (Palacios, V. et al., 2006c), establecen un conjunto de pasos para cualificar cualquier esquema.

En la siguiente figura se resumen los detalles del resultado:

Resultado IV: Recomendaciones para la cualificación de esquemas

- Validación en publicaciones nacionales e internacionales
- Definición de la metodología para la cualificación de esquemas
- Generación a partir del esquema original: proceso independiente

Figura IV-7: Detalles del resultado IV: Recomendaciones para la cualificación de esquemas.

Resultado V: Plantilla de descripción de esquemas de metadatos: Además de las recomendaciones, se incluye una plantilla para la definición de las dos versiones del esquema: interpretable por personas, por ejemplo en HTML, e interpretable por computadoras, en RDF. La plantilla de la versión interpretable por personas ha sido incluida en el *Anexo A: Plantilla de descripción de Esquemas de Metadatos*.

En la siguiente figura se resumen los detalles del resultado:

Resultado V: Plantilla de descripción de esquemas de metadatos

- Basada en recomendaciones del DCMI y W3C
- Representación interpretable por personas
- Representación interpretable por computadoras: RDF

Figura IV-8: Detalles del resultado V: Plantilla de descripción de esquemas de metadatos.

La definición de las recomendaciones de cualificación obedece a la necesidad de establecer un procedimiento que permita cualificar cualquier esquema de metadatos. Es el resultado de aunar distintas recomendaciones, partiendo de las propuestas por el DCMI, para establecer un conjunto de pasos, claramente definidos, que permiten obtener las dos versiones del esquema. De este modo, se ejemplifica la generación de la versión interpretable por computadoras y se elabora, y aplica, una plantilla para la versión interpretable por personas.

Resultado VI: Cualificación del conjunto de esquemas seleccionado: Una vez definido el procedimiento de cualificación, el siguiente resultado obtenido es la cualificación de un conjunto de esquemas seleccionado previamente.

En la siguiente figura se resumen los detalles del resultado:

Resultado VI: Cualificación del conjunto de esquemas seleccionado

- Aplicación de la metodología propuesta
- 6 esquemas cualificados
- 289 elementos cualificados
- Representación compatible y homogénea
- Mantenimiento del lenguaje de representación original
- Mejora en la representación de esquemas escasamente formalizados
- Generador automático de la representación interpretable por personas (en desarrollo)

Figura IV-9: Detalles del resultado VI: Cualificación del conjunto de esquemas seleccionado.

Concretamente, se han generado los esquemas cualificados resumidos en la siguiente tabla:

Esquema original	Esquema cualificado	Elementos	Espacio de nombres
Dcelements	sqdcelements	15	http://purl.org/semse/sqdcelements
Vcard-rdf	sqvcard	59	http://purl.org/semse/sqvcard
FOAF	sqfoaf	74	http://purl.org/semse/sqfoaf
DOAC	sqdoac	33	http://purl.org/semse/sqdoac
DOAP	sqdoap	50	http://purl.org/semse/sqdoap
PIM	sqpim	58	http://purl.org/semse/sqpim

Tabla IV-1: Resumen de los esquemas cualificados generados

Se han generado versiones cualificadas en formato RDF de 6 esquemas, que suponen un total de 289 elementos. Respecto al proceso, destacar que se trata de una tarea sencilla que apenas ha consumido tiempo. Concretamente, no requiere conocimientos sobre el esquema en cuestión y, básicamente, consiste en transcribir el esquema original al formato de cualificación propuesto, de hecho, se ha automatizado parcialmente. En este punto es importante destacar las ventajas que la representación aporta frente a los esquemas originales. Los esquemas generados no sólo permiten incluir semántica, manteniendo la compatibilidad con la versión original del esquema, además proporcionan una representación homogénea de los esquemas, resolviendo incluso problemas de definición, como en el caso de DOAC. Sobre la versión interpretable por

personas, se ha iniciado el desarrollo de un generador automático de estas versiones que permitirá, vía Web, obtener versiones amigables partiendo de las versiones desarrolladas en RDF.

Resultado VII: Recomendaciones para la generación de ontologías específicas: Con las versiones cualificadas generadas, el siguiente resultado ha sido el conjunto de recomendaciones para la generación de las representaciones ontológicas de la semántica de los esquemas, las ontologías específicas.

En la siguiente figura se resumen los detalles del resultado:

Resultado VII: Recomendaciones para la generación de ontologías específicas

- Metodología para la generación de ontologías específicas
- Generación a partir del esquema original: proceso independiente

Figura IV-10: Detalles del resultado VII: Recomendaciones para la generación de ontologías específicas.

Resultado VIII: Generación de las ontologías específicas de los esquemas seleccionados: Establecido el conjunto de pasos para su generación, se han generado las ontologías específicas correspondientes a cada esquema del conjunto de esquemas seleccionado previamente. De este modo, se han generado ontologías específicas con soporte a los lenguajes inglés y español, que comprenden 289 definiciones y 578 etiquetas.

En la siguiente figura se resumen los detalles del resultado:

Resultado VIII: Generación de las ontologías específicas de los esquemas seleccionados

- Aplicación de la metodología propuesta
- 6 ontologías específicas generadas
- 289 conceptos generados: clases y propiedades
- 578 etiquetas en total
- Representación especializada en aspectos semánticos: lenguaje OWL-Full
- Mejora en la representación de esquemas escasamente formalizados
- Corrección y adición de semánticas ambiguas o inexistentes
- Soporte a plurilingüismo: adición de 289 definiciones y 289 etiquetas en español
- Soporte a sinonimia: adición de 102 sinónimos en inglés y español
- Generador automático de la representación interpretable por personas (en desarrollo)

Figura IV-11: Detalles del resultado VIII: Generación de las ontologías específicas de

los esquemas seleccionados.

De forma similar al proceso de cualificación, ha sido necesario definir el procedimiento para la generación de las ontologías específicas. Cabe destacar que, al igual que los esquemas cualificados, estas ontologías son generadas a partir del esquema original. Este aspecto es importante puesto que no establece dependencias en el proceso de generación de las dos representaciones, lo que permite su desarrollo en paralelo y de forma independiente. El procedimiento se ha aplicado a los 6 esquemas seleccionados, generándose representaciones semánticas de los 289 elementos que incluyen, repartidas entre clases y propiedades.

El proceso consiste, inicialmente, en la transcripción del esquema original al formato definido para las ontologías específicas, esto es, haciendo uso de los conceptos y propiedades definidos en la ontología SEMSE. Una vez generada la primera versión de la ontología, ha sido preciso realizar una revisión de las definiciones incluidas en los esquemas originales. El porqué de esta revisión se debe a la escasa formalización de algunos de ellos, por ejemplo PIM donde, en muchos casos, los términos carecían de definición. Como norma se han mantenido las definiciones del esquema original, únicamente se han corregido algunas especialmente ambiguas y se han añadido las que faltaban. En cualquier caso, siempre que se ha hecho una modificación en la semántica de un elemento, la nueva definición se ha etiquetado como [SEMSE] para su posterior identificación. Con la semántica correcta, el siguiente paso es el tratamiento de la sinonimia a través de la adición de las posibles etiquetas alternativas del término. Aquí la norma ha sido incluir sólo aquellos casos especialmente relevantes, principalmente sinónimos bien conocidos, como *telephone* y *phone*. El paso final ha consistido en tratar el plurilingüismo mediante la adición de definiciones y etiquetas en otros lenguajes. Concretamente, se ha añadido soporte al idioma español, lo que ha supuesto añadir la traducción de las definiciones de los 289 elementos, así como sus etiquetas y sinónimos, hasta un total de 578 etiquetas y 102 sinónimos.

El proceso no entraña mayor complejidad, aunque sí es necesario un conocimiento del esquema original para la revisión de las definiciones de los términos. No obstante, en la mayoría de los esquemas, correctamente formalizados, no ha sido preciso realizar alteraciones en la semántica de los elementos. Por otro lado, la definición correcta de los términos es una aportación más de la propuesta, dado que, para su uso y reutilización, todos los términos precisan de una semántica bien definida y, de este modo, se está completando y mejorando la formalización del esquema original.

Respecto a las ontologías generadas, se resumen en la siguiente tabla:

Esquema original	Esquema cualificado	Elementos	Espacio de nombres
Dcelements	dcelementsso	15	http://purl.org/semse/dcelementsso
Vcard-rdf	vcardso	59	http://purl.org/semse/vcardso
FOAF	foafso	74	http://purl.org/semse/foafso
DOAC	doacso	33	http://purl.org/semse/doacso

DOAP	doapso	50	http://purl.org/semse/doapso
PIM	pimso	58	http://purl.org/semse/pimso

Tabla IV-2: Resumen de las ontologías específicas generadas.

Destacar que, en este caso, sólo se incluye la versión en OWL, puesto que son recursos dirigidos al procesamiento automático por parte de computadoras. No obstante, el generador automático de versiones, antes mencionado, permitirá también el procesamiento de las ontologías específicas y, por tanto, obtener versiones interpretables por personas de las mismas.

4.1.1.3 Publicación de esquemas semánticos

Resultado IX: Publicación de esquemas y ontologías en un servidor público: La aportación de esta fase ha consistido en la publicación en Web de todos los artefactos generados en las fases anteriores. De este modo, se han publicado el esquema de cualificadores semánticos, la ontología SEMSE, los esquemas cualificados y las ontologías específicas, siguiendo las políticas y recomendaciones propuestas por el DCMI y el W3C.

En la siguiente figura se resumen los detalles del resultado:

Resultado IX: Publicación de esquemas y ontologías en un servidor público

- Uso de estándares y recomendaciones del DCMI y el W3C
- Generación de reglas de acceso a los recursos publicados
- Publicación del esquema de cualificadores semánticos
- Publicación de la ontología SEMSE
- Publicación de 6 esquemas cualificados
- Publicación de 6 ontologías específicas

Figura IV-12: Detalles del resultado IX: Publicación de esquemas y ontologías en un servidor público.

Haciendo uso de un método de publicación ampliable, se ha garantizado la accesibilidad de los recursos para su uso y reutilización, tanto por parte del sistema de soporte, como por otros sistemas o personas. El método de publicación es sencillo, básicamente la copia de los ficheros en ubicaciones establecidas y la adición de reglas para permitir su acceso. La estructura en el servidor permite la adición futura de nuevos recursos, al tiempo que no implica altos requisitos de computación, al tratarse de un simple servidor Web.

4.1.2 Resultados de la selección de la ontología de referencia

Los principales logros de esta fase se resumen en la siguiente figura:

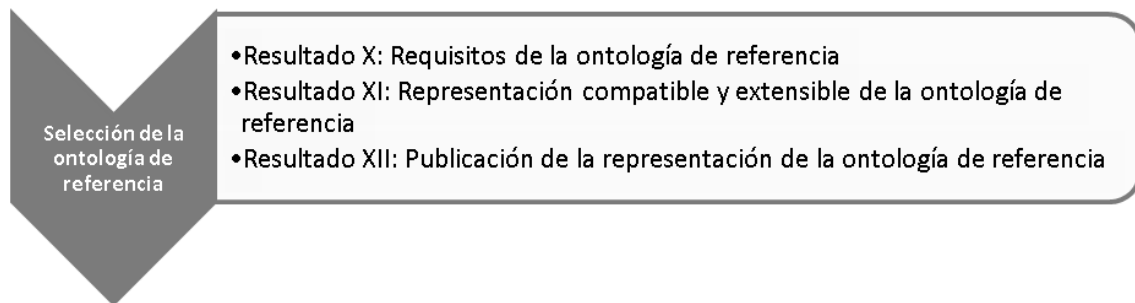


Figura IV-13: Resumen de los logros conseguidos en la fase de selección de la ontología de referencia.

Resultado X: Requisitos de la ontología de referencia: Se han establecido los requisitos que debe cumplir la ontología de referencia en el contexto de la propuesta y se ha seleccionado la ontología más adecuada en base a ellos.

En la siguiente figura se resumen los detalles del resultado:

Resultado X: Requisitos de la ontología de referencia

- Establecimiento del conjunto de requisitos de la ontología de referencia
- Evaluación y selección de la ontología de referencia, en base a los requisitos establecidos

Figura IV-14: Detalles del resultado X: Requisitos de la ontología de referencia.

Tanto los requisitos generados como el proceso de selección realizado, se desarrollan en el marco de la propuesta, no obstante, son extrapolables a aplicaciones con requisitos similares, esto es, sistemas que quieran hacer uso de una, o varias, ontologías genéricas en un entorno Web. De este modo, la exposición de la problemática y sus soluciones expuestas, pueden servir como guía y ejemplo en la evaluación de ontologías genéricas de sistemas similares.

Resultado XI: Representación compatible y extensible de la ontología de referencia: De modo similar, la representación compatible de la ontología de referencia se basa en el conjunto de requisitos anterior, proponiendo un mecanismo de representación y extensión de la ontología original, articulado mediante una ontología de aplicación.

En la siguiente figura se resumen los detalles del resultado:

Resultado XI: Representación compatible y extensible de la ontología de referencia

- Basado en los requisitos anteriores y en criterios de calidad: extensibilidad, compatibilidad y reutilización
- Definición del modo de representación de la ontología de referencia
- Definición del modo de extensión de la ontología de referencia
- Lenguaje de representación: OWL-Full

Figura IV-15: Detalles del resultado XI: Representación compatible y extensible de la ontología de referencia.

La solución propuesta es de nuevo aplicable y, si cabe, necesaria en el uso de ontologías genéricas. Se proporciona, por tanto, una solución al problema común de la extensión de este tipo de ontologías, atendiendo a criterios de:

- Extensibilidad, puesto que el diseño mantiene la independencia de la ontología de referencia y utiliza un sistema de identificación formal, lo que facilita su extensión.
- Compatibilidad, en cuanto a que se mantiene la estructura de la ontología genérica y su lenguaje de definición, lo que facilita la gestión de ambos recursos como un todo.
- Reutilización, tanto de los conceptos de la ontología de referencia como de los extendidos a través de la ontología de aplicación, que, además, serán publicados finalmente en Web.

Resultado XII: Publicación de la representación de la ontología de referencia: De modo similar al realizado para las representaciones semánticas de los esquemas, se ha procedido a la aportación en Web de los artefactos generados. De este modo, se ha publicado la ontología de aplicación, siguiendo las políticas y recomendaciones propuestas por el DCMI y el W3C.

En la siguiente figura se resumen los detalles del resultado:

Resultado XII: Publicación de la representación de la ontología de referencia

- Uso de estándares y recomendaciones del DCMI y el W3C
- Generación de reglas de acceso a los recursos publicados
- Publicación de la ontología de aplicación para la integración y extensión de la ontología de referencia

Figura IV-16: Detalles del resultado XII: Publicación de la representación de la ontología de referencia.

4.1.3 Resultados de la correspondencia entre esquemas y la ontología de referencia

Los principales logros obtenidos en esta fase se resumen en la siguiente figura:

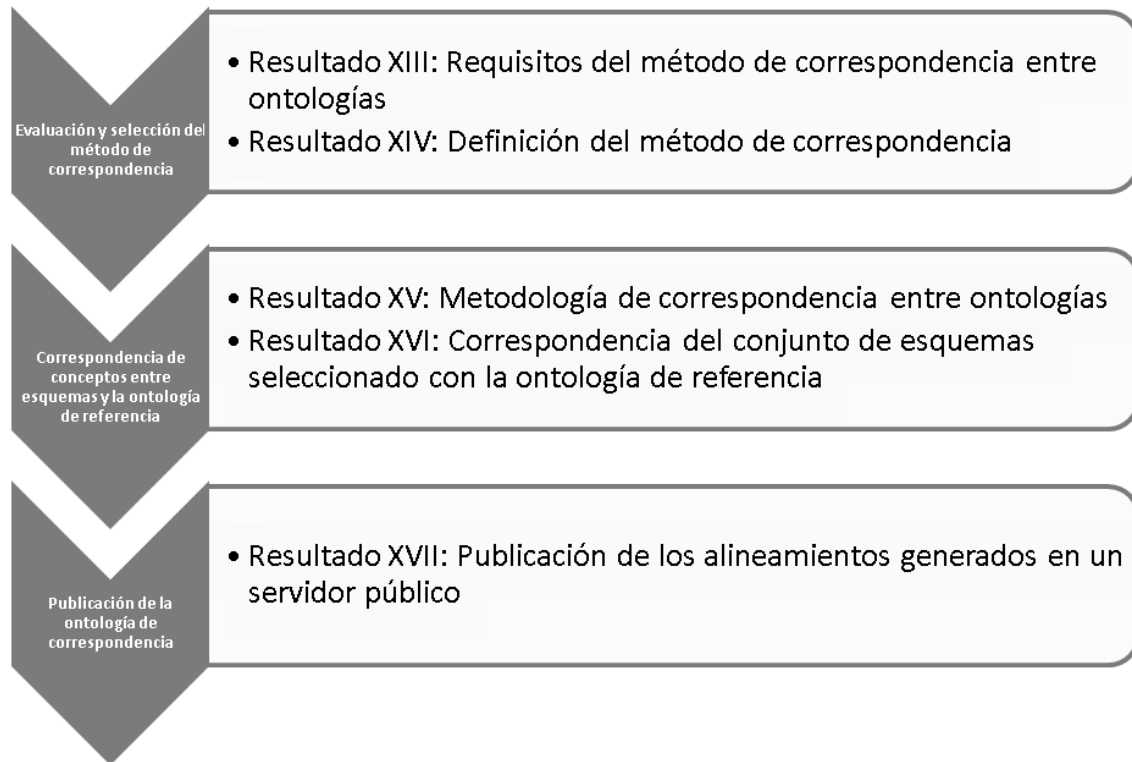


Figura IV-17: Resumen de los logros conseguidos en la fase de correspondencia entre esquemas y la ontología de referencia.

4.1.3.1 Evaluación y selección del método de correspondencia

Resultado XIII: Requisitos del método de correspondencia entre ontologías: Primeramente se han establecido los requisitos que debe cumplir el método de correspondencia. Aunque los requisitos se establecen en el contexto de la propuesta, nuevamente, son extrapolables a otros sistemas que precisen generar alineamientos entre ontologías en el entorno Web. Se proporciona así, un conjunto de necesidades a tener en cuenta en la valoración de las posibles soluciones al problema común de correspondencia entre ontologías. Asimismo, las soluciones y decisiones planteadas pueden servir de ejemplo en la solución de problemas similares.

En la siguiente figura se resumen los detalles del resultado:

Resultado XIII: Requisitos del método de correspondencia entre ontologías

- Definición de los requisitos del método de correspondencia
- Fundamentados en criterios de calidad: extensibilidad, accesibilidad, mantenibilidad y reutilización

Figura IV-18: Detalles del resultado XIII: Requisitos del método de correspondencia entre ontologías.

Resultado XIV: Definición del método de correspondencia: Atendiendo a los requisitos definidos, se propone un método de correspondencia, articulado mediante una ontología de alineamiento, independiente, definida en lenguaje OWL. Asimismo se establece la estructura de las correspondencias, haciendo uso de la ontología SEMSE para su definición.

En la siguiente figura se resumen los detalles del resultado:

Resultado XIV: Definición del método de correspondencia

- Definición del modo de correspondencia basado en los requisitos anteriores
- Definición de una ontología de alineamiento independiente
- Definición de la estructura de las correspondencias en la ontología SEMSE
- Lenguaje de representación: OWL-Full
- Solución extensible, accesible, eficiente, mantenible y reutilizable

Figura IV-19: Detalles del resultado XIV: Definición del método de correspondencia.

La solución propuesta establece un ejemplo de resolución al problema de la correspondencia entre ontologías Web. Proporciona así, una forma de resolver un problema común:

- Extensible, al permitir la adición de nuevas correspondencias internas y externas, así como el uso de múltiples ontologías de referencia.
- Accesible, al utilizar un sistema de identificación formal y un lenguaje compatible en el entorno Web.
- Eficiente y mantenible, en cuanto que el procesamiento de las correspondencias se realiza de forma independiente. A esto hay que añadir que mantiene la independencia de las ontologías a corresponder, evitando la necesidad de

modificarlas, y permite la correspondencia entre dos elementos cualquiera, independientemente de su tipo.

- Reutilizable, en cuanto a que ha sido publicada en Web, permitiéndose su reutilización, tanto a nivel de alineamiento como a nivel de correspondencia.

Respecto a la definición de las correspondencias, destacar que la solución permite establecer nuevas relaciones de correspondencia, a las ya incluidas en la propuesta. De este modo, incluyéndose en la ontología SEMSE y, por tanto, no siendo necesario la modificación de ningún otro componente, sería posible hacer uso de correspondencias más complejas en cuanto a semántica, direccionalidad, uso de valores de confianza, etc.

4.1.3.2 Correspondencia de conceptos entre esquemas y la ontología de referencia

Resultado XV: Metodología de correspondencia entre ontologías: El primer resultado de esta fase es el desarrollo de una metodología de correspondencia entre ontologías. Partiendo de cada caso analizado, se han inducido los pasos necesarios, que incluyen la casuística encontrada en el proceso de generación de correspondencias. La metodología se ha generado, refinado y validado mediante la correspondencia entre los esquemas seleccionados y la ontología de referencia. El análisis de cada caso ha tenido en cuenta aspectos estructurales y semánticos, e incluye el proceso de extensión de la ontología de referencia en caso de ser necesario definir nuevos términos.

En la siguiente figura se resumen los detalles del resultado:

Resultado XV: Metodología de correspondencia entre ontologías

- Desarrollo mediante un proceso inductivo a partir de los casos encontrados en la correspondencia de los esquemas seleccionados y la ontología de referencia
- 289 casos procesados
- Análisis sintáctico y semántico de cada caso
- Establecimiento del método de extensión de la ontología de referencia
- Refinamiento iterativo de la metodología
- Correspondencia manual a nivel de esquema: minimización de errores y compensación del esfuerzo con la cobertura de documentos para cada esquema

Figura IV-20: Detalles del resultado XV: Metodología de correspondencia entre ontologías.

Sobre este primer resultado es importante indicar la aportación que supone el análisis, resolución y ejemplificación de cada caso encontrado en el proceso de correspondencia.. Tratándose de una metodología inducida, no es posible garantizar la aparición de un

caso no contemplado. No obstante, en su elaboración se han tratado casi 300 casos de correspondencia provenientes de esquemas heterogéneos en cuanto a ámbito y formalización, esto proporciona una muestra lo suficientemente representativa como para validar la metodología a un nivel aceptable.

El proceso de generación de la metodología se ha desarrollado de forma iterativa. De este modo, se han generado hipótesis a medida que se iban analizando casos, que han sido refinadas, replanteadas o ampliadas, en función de los nuevos casos encontrados. Estos cambios han provocado la necesidad de reestructurar y refinar las correspondencias ya realizadas, en función de las modificaciones de la metodología, aunque lo cierto es que la correspondencia de los últimos esquemas no ha implicado cambios en la metodología, lo cual apoya la validez del método propuesto.

El análisis de cada caso se ha basado en: aspectos terminológicos, como los formatos de los valores de las propiedades; aspectos estructurales, en cuanto al tipo de los elementos a relacionar, la ubicación de los conceptos dentro de la ontología de referencia, o los rangos y dominios de las propiedades a corresponder; y en aspectos semánticos, analizando los significados de cada elemento o relaciones entre elementos, tales como agregación, generalización, etc. La resolución de cada caso se ha establecido en función de la existencia del término correspondiente en la ontología de referencia. En su caso, ha sido preciso definir el método de extensión del recurso, a través de la ontología de aplicación, mediante la adición, especialización, generalización y agregación de conceptos.

Sobre la aplicabilidad de la metodología de correspondencia, se base en un método manual, aplicable a cualquier esquema y a cualquier ontología de referencia, salvando las posibles diferencias en cuanto a su formalismo de representación. El método propuesto es manual, debido a que uno de los objetivos era minimizar posibles errores de correspondencia que pudieran introducir ruido en la verificación y validación de la propuesta. Además, al tratarse de un método cuya entrada es a nivel de esquema, y no de instancia, permite obtener los beneficios de alinear un reducido número de esquemas que cubren un amplio número de instancias. Como ejemplo, sólo la incorporación de los esquemas de Dublin Core y FOAF cubren alrededor de 4 millones de documentos, lo cual apoya la tesis de que el esfuerzo en la correspondencia manual queda compensado por la cobertura de documentos, máxime cuando se minimizan los errores de correspondencia, cuyo impacto sería muy elevado dado el volumen de instancias.

Resultado XVI: Correspondencia del conjunto de esquemas seleccionado con la ontología de referencia: El segundo resultado obtenido en esta fase es el alineamiento del conjunto de esquemas seleccionado con la ontología de referencia, de este modo, se han alineado los 289 elementos, procedentes de los 6 esquemas seleccionados, contra los conceptos y propiedades de la ontología PROTON.

En la siguiente figura se resumen los detalles del resultado:

Resultado XVI: Correspondencia del conjunto de esquemas seleccionado con la ontología de referencia

- Aplicación de la metodología propuesta
- 6 ontologías específicas alineadas contra la ontología de referencia
- 289 correspondencias generadas, almacenadas en la ontología de alineamiento
- Adición de 154 conceptos nuevos, almacenados en la ontología de aplicación
- Proceso incremental: la correspondencia se apoya en alineamientos existentes y conceptos que extienden la ontología de referencia
- Facilitado por: *crosswalks* y solapamientos entre los esquemas
- Validación por ingenieros reponsables del dominio e ingenieros de dominio

Figura IV-21: Detalles del resultado XVI: Correspondencia del conjunto de esquemas seleccionado con la ontología de referencia.

El proceso de correspondencia ha generado 289 correspondencias almacenadas en la ontología de alineamiento y 154 elementos nuevos, almacenados en la ontología de aplicación, que extienden la versión original de la ontología PROTON. En cuanto al desarrollo del proceso de correspondencia, se ha aplicado la metodología propuesta abordando cada esquema de forma secuencial. Es importante destacar la incrementalidad del proceso, de este modo, la correspondencia de un esquema hace uso y se beneficia de:

- Los alineamientos existentes, como solución a casos comunes entre los esquemas ya alineados y el nuevo a incluir.
- Los nuevos conceptos incluidos como extensión de la ontología de referencia, al representar clases y propiedades reutilizables en la generación de nuevas correspondencias.

Esta incrementalidad ha facilitado el proceso de correspondencia, a medida que se añadían nuevos esquemas y mayor era el conjunto ya alineado. Sólo en casos concretos ha sido preciso generalizar algún concepto de la ontología de referencia o de la ontología de aplicación, de modo que incluyera al nuevo concepto además de los ya cubiertos. En este punto cabe destacar la ayuda que los *crosswalks*, o alineamientos uno a uno, pueden prestar al proceso. Contar con un alineamiento entre el esquema a añadir y un esquema ya incluido, puede facilitar la correspondencia, en función del grado de solapamiento de ambos esquemas. En líneas generales, es una buena práctica recopilar este tipo de recursos como paso previo al proceso de correspondencia. Como es lógico, el conocimiento del esquema y la experiencia en el proceso de correspondencia son

aspectos importantes para obtener un alineamiento de calidad, de ahí que se recomiende la validación del resultado a través del consenso entre los ingenieros responsables del dominio y los ingenieros de dominio.

4.1.3.3 Publicación de la ontología de correspondencia

Resultado XVII: Publicación de los alineamientos generados en un servidor público: De modo similar al realizado para las representaciones semánticas de los esquemas, se ha procedido a la aportación en Web de los artefactos generados. De este modo, se ha publicado la ontología de alineamiento, siguiendo las políticas y recomendaciones propuestas por el DCMI y el W3C.

En la siguiente figura se resumen los detalles del resultado:

Resultado XVII: Publicación de los alineamientos generados en un servidor público

- Uso de estándares y recomendaciones del DCMI y el W3C
- Generación de reglas de acceso a los recursos publicados
- Publicación de la ontología de alineamiento que alberga las correspondencias entre ontologías específicas y la ontología de referencia

Figura IV-22: Detalles del resultado XVII: Publicación de los alineamientos generados en un servidor público.

Esta publicación es un hito importante, dado que no es habitual encontrar este tipo de recursos compartidos en abierto. Si bien existen herramientas, métodos, e incluso servidores, para la gestión de alineamientos, estos recursos suelen tener un carácter privado, como elemento nuclear en la integración de fuentes heterogéneas de conocimiento. Esta publicación tiene como objetivo poner a disposición de sistemas y usuarios los alineamientos generados, para su uso, reutilización y enriquecimiento, a través de la retroalimentación recibida por parte de dichos agentes.

4.1.4 Resultados de la validación y verificación de la propuesta

Los principales logros de esta fase se resumen en la siguiente figura:

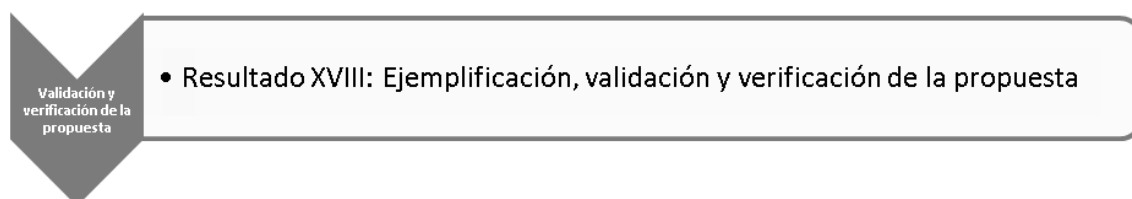


Figura IV-23: Resumen de los logros conseguidos en la fase de validación y verificación de la propuesta.

Resultado XVIII: Ejemplificación, validación y verificación de la propuesta: El

objetivo de los resultados obtenidos en esta fase es el de validar y verificar la propuesta. A través de varios ejemplos de consulta sobre un conjunto de esquemas reales y con un conjunto de elementos significativo, se ilustra cómo los distintos componentes colaboran en la obtención del resultado, cómo los resultados satisfacen la búsqueda planteada, así como la calidad de los resultados, en base a los criterios de evaluación típicos en la recuperación de información, precisión y exhaustividad. Los resultados son, además, comparados con los obtenidos en un proceso de consulta clásico, esto es, una consulta sintáctica sobre un corpus indizado previamente, y con los obtenidos por uno de los recuperadores semánticos más populares: *Watson*.

En la siguiente figura se resumen los detalles del resultado:

Resultado XVIII: Ejemplificación, validación y verificación de la propuesta

- Ejemplificación del funcionamiento de la propuesta
- Validación de la propuesta mediante la satisfacción de requisitos: interoperatividad, reutilización, recuperación conceptual y contextual
- Verificación de la propuesta mediante indicadores de calidad: precisión y exhaustividad
- Mejores resultados en una comparación cuantitativa con búsquedas sintácticas y recuperadores semánticos
- Análisis de resultados y propuesta de mejoras

Figura IV-24: Detalles del resultado XVIII: Ejemplificación, validación y verificación de la propuesta.

La evaluación de los resultados muestra, primeramente, cómo la propuesta satisface los requisitos para los que se diseñó:

- Mejora la interoperatividad entre esquemas de metadatos, puesto que el resultado incluye tanto el concepto buscado como los conceptos afines al mismo, independientemente del esquema en el que se encuentren y habiéndose seleccionado un conjunto de esquemas de ámbito heterogéneo.
- Posibilita la reutilización de esquemas, por el motivo señalado en el punto anterior y porque permite realizar búsquedas conceptuales. Estas características permiten al usuario obtener un conjunto de resultados, del cual puede seleccionar el concepto que más le interese. Además, a diferencia de los recuperadores sintácticos, el proceso de selección se realiza en base al significado del concepto.
- Permite obtener un conjunto de resultados basado en las relaciones entre conceptos, incluyendo las relaciones establecidas en la ontología de alineamiento y las definidas en la ontología de referencia, o su extensión, la ontología de aplicación.

- Permite realizar recuperación conceptual, puesto que las consultas se basan en el significado de los conceptos, así como las relaciones establecidas entre éstos, el resultado se obtiene mediante una consulta fundamentada en la semántica y no en la sintaxis.

Además de la validación, a partir de los resultados se verifica que la propuesta realiza su función con calidad. Los valores obtenidos para los criterios de calidad, tal y como se resumen en las tablas *III-13* a *III-15*, muestran cómo se obtiene la máxima precisión y exhaustividad, tanto en las consultas conceptuales como en las contextuales. Además, se puede comprobar cómo mejoran notablemente los resultados obtenidos mediante una búsqueda sintáctica y el buscador semántico *Watson*. Destacar también cómo, en el caso de la consulta contextual, es posible ampliar los resultados devueltos incluyendo los derivados del contexto del concepto, lo cual aporta valor al resultado, al tiempo que proporciona un resultado más amplio, derivado de las relaciones establecidas en la ontología de referencia subyacente y sus correspondencias conceptuales con los esquemas.

La evaluación de otros buscadores semánticos pone de manifiesto algunas carencias que presentan este tipo de recuperadores. La imposibilidad de que el usuario utilice como criterio de búsqueda el significado del concepto, la de restringir el dominio de búsqueda o la de diferenciar entre conceptos importados de otros esquemas de los definidos en el propio esquema. Estos motivos hacen que, en la mayoría de los casos, las consultas ofrezcan unos resultados muy semejantes a los obtenidos por una búsqueda sintáctica. A esto hay que añadir la escasa usabilidad en la presentación de los resultados al usuario, lo que dificulta su procesamiento y reduce su utilidad.

Las consultas definidas, muestran las posibilidades que ofrece la base ontológica desarrollada, en cuanto a la recuperación conceptual de información. De este modo se ilustra cómo podría hacerse uso de este recurso, planteándose la posibilidad de definir consultas más complejas, tanto en la definición de los términos de búsqueda como en el proceso de consulta. De modo similar, se plantea la posibilidad de ordenar el resultado en base a información extra albergada en metadatos. Esto permitiría ordenar el resultado en base a la relevancia, establecida en función de la popularidad del esquema, su novedad, su estabilidad, etc.

Finalmente, destacar también el carácter colaborativo de las consultas planteadas. En línea con los fundamentos de la Web 2.0, el usuario tiene un papel activo en el proceso de consulta, refinando los resultados parciales a través de la selección de los conceptos que le interesan, en base a su significado. Como efecto de la participación del usuario, se obtiene un conjunto de resultados con la máxima relevancia, objetivo último de todo proceso de recuperación de información; al mismo tiempo se obtiene información que podría permitir el análisis del comportamiento de los usuarios en las consultas.

4.1.5 Resultados del proceso de desarrollo del sistema informático de soporte

Los resultados del proceso de desarrollo, como es habitual en la Ingeniería del Software, comprenden el conjunto de entregables y artefactos generados como resultado de cada

una de las fases del proceso. Así, se pueden resumir en:

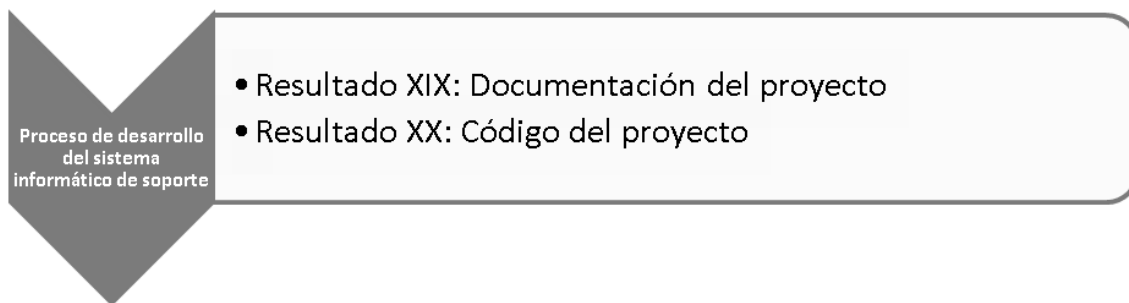


Figura IV-25: Resumen de los logros conseguidos en la fase de desarrollo del proyecto.

Resultado XIX: Documentación del proyecto: En la siguiente figura se resumen los detalles del resultado:

Resultado XIX: Documentación del proyecto

- Elaboración del documento de requisitos de usuario
- Elaboración del documento de requisitos software
- Elaboración de una evaluación de repositorios semánticos
- Elaboración de un estudio tecnológico y del diseño de la arquitectura de la plataforma de soporte
- Elaboración del documento de diseño detallado, codificación y pruebas

Figura IV-26: Detalles del resultado XIX: Documentación del proyecto.

A continuación se describen brevemente los documentos generados:

- Documento de Requisitos de Usuario (Ayensa, I. et al., 2010a): que recoge el conjunto de requisitos proporcionado por los usuarios de la aplicación, esto es, la funcionalidad que la aplicación debe satisfacer, desde el punto de vista del usuario final.
- Documento de Requisitos Software (Ayensa, I. et al., 2010b): que comprende el conjunto de requisitos anterior, después de un proceso de refinamiento y especificación técnica, desarrollados y dirigidos a informáticos. De este modo se establece la funcionalidad a satisfacer desde un punto de vista técnico, como base para su desarrollo. Incluye también otros aspectos propios de la aplicación como son el diseño de las páginas o el mapa web del sistema.
- Evaluación de repositorios semánticos (Corbera, S., 2007): estudio realizado con el objetivo de evaluar las opciones tecnológicas existentes para la edición y gestión de ontologías. A través de este estudio se pudo valorar, comparar y seleccionar la solución que mejor se adaptaba a los requisitos del sistema. Por su interés con la presente propuesta, en el *Anexo B: Resultados de la evaluación de*

repositorios semánticos se incluye la tabla resumen de la evaluación de repositorios. En (Corbera, S., 2007) se puede consultar la evaluación completa, incluyendo la realizada para editores de ontologías.

- Estudio tecnológico y diseño de la arquitectura (Fernández-Tostado, T., 2009): este estudio incluye una evaluación y valoración de las tecnologías de desarrollo existentes, de cara a seleccionar aquella más adecuada a los requisitos del sistema. Junto con la selección tecnológica se definió también la arquitectura, software y hardware, más adecuada, en base a los requisitos y a criterios de calidad como: extensibilidad, mantenibilidad o adaptabilidad.
- Diseño detallado, codificación y pruebas (Gárate, F.J., 2010): aunque el desarrollo no forma parte de la presente propuesta, pueden consultarse detalles sobre las últimas fases del proceso de desarrollo en su estado actual.

Resultado XX: Código del proyecto: Basado en el planteamiento teórico de esta tesis, el proyecto “*Desarrollo de un sistema de Recuperación Conceptual mediante Niveles Semánticos en la representación de Esquemas de Metadatos*”, dentro del Plan Nacional de I+D+i, ha generado código correspondiente al sistema principal, *Sistema de Gestión del Conocimiento basado en Ontologías (SGCO)*, y a la araña de búsqueda, disponible en (Semantic Metadata Search, 2010c). También se encuentran en desarrollo otras aplicaciones que complementan la propuesta como: el buscador/indizador de documentos o el sistema para la generación de esquemas interpretables por personas.

En el proceso de gestión y soporte del proyecto se han desplegado: un portal del proyecto (Semantic Metadata Search, 2010b), un sistema de control de versiones (Semantic Metadata Search, 2010c), un servidor de aplicaciones (Semantic Metadata Search, 2010d) y un servidor de persistencia.

En la siguiente figura se resumen los detalles del resultado:

Resultado XX: Código del proyecto

- Generación del código del *Sistema de Gestión del Conocimiento basado en Ontologías (SGCO)* (en desarrollo)
- Generación del código del Buscador/indizador de documentos (en desarrollo)
- Implementación del Generador de esquemas interpretables por personas (en desarrollo)
- Despliegue de la infraestructura de soporte: portal del proyecto, sistema de control de versiones, servidor de aplicaciones y servidor de persistencia

Figura IV-27: Detalles del resultado XX: Código del proyecto.

Si bien la capacidad de reutilización es inversamente proporcional al nivel de especificidad de los artefactos generados en un proceso de desarrollo, los resultados generados aportan, en distintos niveles de abstracción, ejemplos extrapolables al

desarrollo de aplicaciones similares.

- El documento de requisitos de usuario proporciona una visión de las capacidades que un sistema de gestión de conocimiento debería proporcionar. Esto, sumado a su alto nivel de abstracción, por tratarse de uno de los primeros entregables de un proyecto, lo convierten en un documento de gran interés para la reutilización en desarrollos similares.
- El conjunto de requisitos software proporciona una visión técnica de la funcionalidad y restricciones de la aplicación, extrapolable a otros sistemas.
- El estudio de evaluación de repositorios semánticos, ofrece una visión general de las posibilidades tecnológicas para la edición y gestión de ontologías, valora dichas posibilidades y las compara, aportando una valiosa herramienta de cara a la selección de la tecnología más adecuada para una aplicación basada en ontologías.
- El estudio tecnológico y el diseño de la arquitectura proporcionan una visión general, valoración y comparación sobre las tecnologías de desarrollo actuales, esto lo convierte en un recurso de gran interés a la hora de tomar una decisión sobre la tecnología a utilizar. Además, a través de la definición de la arquitectura, ejemplifica una solución de alta calidad que hace uso de las tecnologías mejor valoradas por los desarrolladores, al tiempo que muestra la integración de los componentes propuestos. Como recurso escaso en este ámbito, se trata de un recurso de gran valor para un arquitecto de software.
- La documentación relativa al diseño detallado, codificación y pruebas es la menos susceptible de reutilización, si bien ejemplifica el desarrollo del sistema en estas fases.
- El código generado y la aplicación, en su estado actual, están accesibles para su descarga y consulta. Sujeto a una licencia libre, cualquier desarrollador tiene la posibilidad de descargar, consultar, modificar y distribuir el código del proyecto.
- Respecto a la infraestructura de soporte, destacar la labor del portal del proyecto de cara a la publicación de los artefactos desarrollados, y de otros sistemas satélite, responsables de la descarga y ejecución del recurso, así como de su desarrollo.

4.2 Evaluación y comparativa de la propuesta

En este apartado se presenta una evaluación de la propuesta con el objetivo de realizar una comparación con otros métodos de recuperación de documentos XML. A través de un proceso de evaluación cualitativa de los distintos métodos, un análisis de los resultados y una discusión de los mismos. Se pretende posicionar el enfoque propuesto frente a otras alternativas actuales, dirigidas a la representación y recuperación de documentos XML.

4.2.1 Presentación del método de evaluación

Para poder realizar la comparación entre las distintas propuestas, es preciso establecer un método de evaluación. Concretamente, en este caso se ha seleccionado el método de evaluación DESMET (Kitchenham, 1996). DESMET es un método ampliamente utilizado en técnicas de evaluación, que facilita la planificación, definición y ejecución de procesos de evaluación fiables e imparciales, aplicables a métodos y herramientas de Ingeniería del Software.

Respecto al tipo de evaluación, DESMET cubre métodos cuantitativos y cualitativos. En este apartado se aplicará DESMET a la evaluación cualitativa de los requisitos expuestos a lo largo de la tesis, mostrando las ventajas de la propuesta para la organización y recuperación de elementos semánticos. Si bien en un futuro, cuando se incorporen estas propuestas a una plataforma informática se podrá automatizar la obtención y el análisis cuantitativo de resultados.

Siguiendo la terminología de DESMET, se realizará un *análisis de características (feature analysis)*, esto es, una evaluación basada en la identificación de requisitos para una determinada tarea, y la correspondencia de éstos con las características que proporciona un conjunto de métodos, dirigidas a soportar los requisitos establecidos. Esto es, evaluar el nivel de soporte que los métodos proporcionan al conjunto de requisitos establecido. Una vez establecidos, el evaluador valorará el nivel de satisfacción que cada método proporciona al conjunto de requisitos, analizará y presentará los resultados.

Las principales tareas de un análisis de características son:

1. Selección de un conjunto de candidatos, bien métodos o bien herramientas, a evaluar.
2. Decidir el conjunto de características o propiedades a evaluar de cada candidato.
3. Priorizar las características de acuerdo con los requisitos de los usuarios del método o herramienta.
4. Acordar un sistema de valoración aplicable a todas las características.
5. Realizar la evaluación para determinar el grado de cumplimiento que los candidatos proporcionan al conjunto de características establecido.
6. Analizar e interpretar los resultados.
7. Presentar los resultados.

Mediante este conjunto de tareas se establece, primeramente, el conjunto de candidatos a evaluar. A continuación se define un marco común de evaluación, expresado en términos de un conjunto de propiedades, cualidades, características, atributos, etc., aplicables al conjunto de candidatos. Estas características son priorizadas en función de su necesidad, por ejemplo: obligatorias, deseables y opcionales, según sean requeridas por los usuarios. Una vez definido el marco comparativo, se evalúa el nivel de soporte

que cada candidato proporciona al conjunto de características, asignando un valor a cada una de ellas según un sistema de valoración acordado. Una vez puntuados los métodos, se lleva a cabo su comparativa y el análisis de los resultados.

En el siguiente apartado, *4.2.2 Definición del marco comparativo*, se llevará a cabo la definición del marco comparativo de la evaluación. Seguidamente, el apartado *4.2.3 Evaluación de métodos de recuperación de documentos XML*, incluirá el proceso de evaluación de los candidatos seleccionados y, finalmente, en el apartado *4.2.4 Presentación, análisis y discusión de resultados*, se presentarán y discutirán los resultados obtenidos.

4.2.2 Definición del marco comparativo

Siguiendo el método DESMET, la primera tarea a realizar es el establecimiento del conjunto de candidatos a evaluar.

Puesto que en esta tesis se propone un sistema de recuperación conceptual de esquemas y documentos XML, la comparativa se realizará en el ámbito de la recuperación de documentos, esto es, métodos de recuperación de esquemas y documentos XML. Tal y como se presentaban en el apartado *2.4 Recuperación de documentos XML*, se evaluará la propuesta frente a métodos de recuperación alternativos, a saber: motores XML, directorios de DTDs/Esquemas, motores de metadatos, directorios de metadatos, motores de ontologías y directorios de ontologías.

Respecto al conjunto de candidatos seleccionado destacar, primeramente, que la propuesta se evaluará en base al planteamiento teórico expuesto en la tesis y a la plataforma informática de soporte que se ha diseñado. El resto de candidatos se evaluarán de forma genérica, esto es, sin entrar en sistemas o desarrollos concretos debido a su elevado número. No obstante, en caso de interés, se comentarán aspectos relativos a posibles implementaciones alternativas de un mismo método.

Una vez definido el conjunto de candidatos, el siguiente paso es establecer el conjunto de características a evaluar para cada uno de los métodos seleccionados. Según se propone en DESMET, existen dos tipos de características:

- **Características simples:** que pueden estar presentes o ausentes. Son valoradas utilizando una escala nominal: *SÍ* o *NO*.
- **Características compuestas:** donde se valora el grado de soporte proporcionado por el método evaluado. Son cuantificadas utilizando una escala ordinal.

El conjunto de características a evaluar para cada método, se ha clasificado en tres grandes grupos:

- **Gestión de Esquemas:** que comprende características relativas a la gestión de los esquemas.
- **Gestión Semántica:** que comprende características relacionadas con los significados de los conceptos y su gestión.

- **Consulta:** que comprende características concernientes al proceso de consulta y la gestión de los resultados obtenidos.

Obtenidas a partir de los objetivos de la presente propuesta, a continuación se describen las características a evaluar, clasificadas por grupo. Para cada una de ellas se incluye una descripción y su tipo, de acuerdo con el método DESMET:

- **Gestión de Esquemas**

- *Interoperatividad:* Capacidad de establecer relaciones entre conceptos de los esquemas. Es una característica de tipo simple.
- *Formalización:* Capacidad de completar o mejorar la formalización de un esquema como parte de su gestión. Es una característica de tipo simple.
- *Interactividad:* Posibilidad de que el usuario tenga una participación activa en la Gestión de Esquemas del sistema, en línea con las directrices Web 2.0. Es una capacidad de tipo compuesto.

- **Gestión Semántica**

- *Desambiguación:* Capacidad de eliminar la ambigüedad estructural y semántica de un conceptos. Es una capacidad de tipo simple.
- *Marco semántico:* Ámbito en el que se gestiona la semántica de los conceptos y que puede ser: ninguno, local al esquema, local con relaciones entre esquemas o global con relaciones entre esquemas mediante un recurso compartido, por ejemplo una ontología. Es una capacidad de tipo compuesto.
- *Plurilingüismo:* Capacidad de soportar múltiples idiomas. Es una característica de tipo simple.
- *Sinonimia:* Capacidad de resolver aspectos derivados de conceptos distintos con un mismo significado. Es una característica de tipo compuesto.
- *Ámbito:* Dominio en el que se define la semántica de los esquemas a gestionar y que puede ser: homogéneo o heterogéneo. Es una característica de tipo simple.
- *Extensibilidad:* Capacidad de ampliación de la representación de la semántica de los esquemas. Es una característica de tipo compuesto.
- *Reusabilidad:* Capacidad de reutilización de la representación de la semántica de los esquemas. Es una característica de tipo compuesto.
- *Modificabilidad:* Capacidad de modificación de la representación de la semántica de los esquemas. Es una característica de tipo compuesto.
- *Lenguaje:* Capacidad de representación del lenguaje utilizado en la formalización de la semántica. Es una característica de tipo compuesto.

- **Consulta**

- *Semántica*: Capacidad de expresar el significado concreto del concepto en el proceso de consulta. Es una característica de tipo simple.
- *Consulta conceptual*: Capacidad de realizar consultas en base a los significados de los conceptos. Es una característica de tipo compuesto.
- *Consulta contextual*: Capacidad de obtener resultados derivados de las relaciones existentes entre conceptos. Es una característica de tipo compuesto.
- *Recuperación de documentos*: Capacidad de obtener documentos semánticos derivados de esquemas, además de los propios esquemas. Es una característica de tipo simple.

Una vez establecidas las características a evaluar, el siguiente paso del método es priorizarlas en función de su importancia. De acuerdo con los requisitos de la propuesta y su planteamiento teórico, la importancia establecida a cada característica se resume en la siguiente tabla:

Característica	Importancia	Justificación
Interoperatividad	Obligatorio	Aspecto fundamental para lograr la interoperatividad entre esquemas
Formalización	Obligatorio	Aspecto fundamental para la reutilización y desambiguación de esquemas
Interactividad	Deseable	Aspecto que aprovecha la aportación de los usuarios
Desambiguación	Obligatorio	Aspecto fundamental para posibilitar la recuperación conceptual
Marco semántico	Obligatorio	Aspecto fundamental para posibilitar la recuperación conceptual
Plurilingüismo	Deseable	Aspecto deseable para la consulta y reutilización de esquemas
Sinonimia	Obligatorio	Aspecto fundamental para la consulta conceptual
Ámbito	Obligatorio	Aspecto fundamental para la aplicación del método en diversos dominios de conocimiento
Extensibilidad	Deseable	Aspecto deseable en la gestión de la semántica de los esquemas
Reusabilidad	Deseable	Aspecto que permite rentabilizar el esfuerzo realizado en la representación semántica, mediante el uso en futuras aplicaciones
Modificabilidad	Deseable	Aspecto deseable en la gestión de la semántica de los esquemas
Lenguaje	Opcional	Aspecto que puede mejorar la representación y gestión de la semántica del esquema
Semántica	Obligatorio	Aspecto fundamental para la consulta conceptual
Consulta conceptual	Obligatorio	Aspecto fundamental para la reutilización y uso de esquemas

Consulta contextual	Obligatorio	Aspecto fundamental para la reutilización y uso de esquemas
Recuperación de documentos	Opcional	Aspecto que aporta valor añadido a los resultados de la consulta

Tabla IV-3: Importancia de cada característica y breve justificación

Una vez priorizadas las características, el siguiente paso consiste en establecer un sistema de valoración. El sistema de valoración tiene que establecer una escala ordinal para las características y para los niveles de importancia asignados a ellas.

Comenzando por las características simples, cuyos posibles valores son *SÍ* o *NO*, su sistema de valoración se resume en la siguiente tabla:

Características simples	Valor
NO	0
SÍ	5

Tabla IV-4: Sistema de valoración de características simples

Como se muestra en el sistema de valoración, el valor *SÍ* recibe la mayor ponderación y el valor *NO* recibe la menor. El porqué de estos valores es la equiparación de los pesos de estas características con los de las características compuestas, descritos a continuación.

Respecto a las características compuestas, aquellas que toman valores ordinales, su sistema de valoración se resume en la siguiente tabla:

Características compuestas	Valor
Ninguno	0
Bajo	1
Medio	3
Alto	5

Tabla IV-5: Sistema de valoración de características compuestas

Como se muestra en el sistema de valoración, se han asignado pesos entre 0 y 5. El valor más bajo, *Ninguno*, se corresponde con la ausencia de soporte a la característica en cuestión. El valor *Bajo* implica que la característica proporciona algún soporte, sin considerarlo un aspecto de interés. El valor *Medio* implica que la característica proporciona un soporte parcial, considerando el aspecto de interés. El valor *Alto* implica que la característica proporciona un soporte completo, considerando el aspecto como fundamental. Destacar el caso particular de la característica *marco semántico*, en este caso se ha realizado una correspondencia entre los valores de la característica y el sistema de valoración, estableciéndose: *ninguno*=0; *local al esquema*=1; *local con relaciones entre esquemas*=3 y *global con relaciones entre esquemas mediante recurso compartido*=5. Esta correspondencia está alineada con la definición de una representación semántica global haciendo uso de ontologías, propuesta en la presente tesis.

Para finalizar el sistema de valoración, es preciso asignar pesos a los valores de importancia asociados a cada característica. En la siguiente tabla se resumen los pesos asignados a los valores de importancia:

Importancia	Valor
Opcional	3
Deseable	6
Obligatorio	10

Tabla IV-6: Sistema de valoración de la importancia de las características

Una vez establecido el sistema de valoración, el siguiente paso será la evaluación de cada uno de los métodos candidatos. Esta valoración se presenta en el siguiente apartado.

4.2.3 Evaluación de métodos de recuperación de documentos XML

El proceso de evaluación consiste en asignar un valor a cada característica, que represente el soporte que el método evaluado proporciona a una dicha característica. El valor a asignar dependerá del tipo de característica, simple o compuesta, y se encontrará dentro del rango definido por el sistema de valoración expuesto en el apartado anterior.

La evaluación de cada método se ha realizado a partir del análisis de sus ejemplos representativos, incluidos en el apartado 2.4 *Recuperación de documentos XML*. Para cada valor asignado se incluirá una breve justificación con el objetivo de clarificar el criterio de dicha asignación. Destacar que, por defecto, se ha tomado el valor más alto entre los ejemplos representativos. De este modo, cada método ha obtenido el mejor valor entre los sistemas que lo representan.

A continuación se resume, para cada característica, los valores obtenidos por cada método candidato:

- **Interoperatividad:** No soportan esta característica los motores XML, los directorios de esquemas ni los directorios de metadatos. Los registros de metadatos sí incorporan relaciones uno a uno entre esquemas. Los motores de ontologías, en algún caso concreto como *Watson*, analizan las relaciones entre conceptos. Los directorios de ontologías permiten incluir correspondencias uno a uno entre ontologías. Los tres primeros candidatos reciben el valor *NO* y los tres últimos, incluyendo la presente propuesta, reciben el valor *SÍ*.
- **Formalización:** Ninguno de los candidatos, salvo la propuesta, soporta esta característica. Concretamente, hacen uso del esquema tal y como lo define la entidad responsable de su creación y mantenimiento. Todos los candidatos reciben el valor *NO*, a excepción de la propuesta que recibe el valor *SÍ*.
- **Interactividad:** Los motores XML soportan esta característica de forma indirecta, quedará a criterio de la organización que los implante proporcionar al usuario la posibilidad de tener una participación activa. Por este motivo recibe el valor *Medio*. Los directorios de esquemas, los registros de metadatos y los directorios de ontologías sí suelen proporcionar la posibilidad de que el usuario

incorpore recursos al sistema facilitando, en algunos casos, herramientas para su gestión. Reciben, junto con la propuesta, el valor *Alto*. Los directorios de metadatos, al igual que los motores de ontologías, suelen estar cerrados a la intervención del usuario, salvo en los procesos de consulta. Reciben, por tanto, el valor *Bajo*.

- **Desambiguación:** Al hacer uso del esquema original, ninguno de los candidatos, salvo la propuesta, abordan la desambiguación estructural y semántica. Esto es, la representación sintáctica y semántica desambiguada, para cada concepto y propiedad incluidos en el esquema. Todos ellos reciben el valor *NO* a excepción de la propuesta, que recibe el valor *SÍ*.
- **Marco semántico:** Los motores XML, los directorios de esquemas y los directorios de metadatos no hacen uso de la semántica asociada al concepto, más que como parte de la definición del mismo. Los motores de metadatos, los motores de ontologías y los directorios de ontologías hacen uso de la semántica local al esquema, incluyendo relaciones con otros esquemas, habitualmente uno a uno. Los tres primeros candidatos reciben el valor *Ninguno* y los tres siguientes *Medio*. La propuesta, al hacer uso de una representación semántica global mediante un recurso compartido, recibe el valor *Alto*.
- **Plurilingüismo:** Ninguno de los candidatos, salvo alguna iniciativa concreta desarrollada en este ámbito y la propia propuesta, soportan esta característica. Reciben, por tanto, la calificación *NO*. La propuesta recibe el valor *SÍ*.
- **Sinonimia:** No es una característica contemplada en los motores XML, los directorios de esquemas y los directorios de metadatos. Se considera cubierta parcialmente en el resto de los candidatos, al soportar la definición de correspondencias entre conceptos, uno a uno. La propuesta, mediante el uso de una representación global de la semántica, cubre completamente la característica. Los tres primeros candidatos reciben el valor *Ninguno* y los tres siguientes *Medio* y la propuesta *Alto*.
- **Ámbito:** El ámbito de los candidatos es, habitualmente, heterogéneo. No obstante, en el caso de los motores de metadatos y los directorios de metadatos, suele estar restringido a uno o pocos esquemas. Estos dos candidatos reciben el valor *Bajo* y el resto el valor *Alto*.
- **Extensibilidad:** Los motores XML, los directorios de esquemas y los directorios de metadatos no incluyen una representación específica de la semántica asociada al concepto, más que la definición del mismo. Los motores de metadatos, los motores de ontologías y los directorios de ontologías hacen uso de la semántica local al esquema, incluyendo relaciones con otros esquemas, habitualmente uno a uno. Este tipo de relaciones dificulta la extensibilidad, en cuanto que es necesario establecer la correspondencia de cada concepto con todos los demás. La propuesta incluye la posibilidad de extender la representación semántica, resolviendo la correspondencia entre el concepto y la ontología de referencia. Los tres primeros candidatos reciben el valor *Bajo*, los tres siguientes *Medio*, y la propuesta recibe el valor *Alto*.

- **Reusabilidad:** De los candidatos que hacen uso de la semántica del esquema, motores de metadatos, motores de ontologías y directorios de ontologías, sólo los últimos contemplan la publicación de los alineamientos uno a uno para su reúso. La propuesta permite el acceso libre a la representación semántica completa, incluyendo los esquemas, los alineamientos, la ontología de referencia y su extensión. Los candidatos reciben el valor *Ninguno*, a excepción de los directorios de ontologías, que reciben el valor *Medio*, y la propuesta, que recibe el valor *Alto*.
- **Modificabilidad:** La capacidad de modificación de la representación semántica queda restringida a los métodos que hacen uso de ella: motores de metadatos, motores de ontologías, directorios de ontologías y la propuesta. Los tres primeros, al hacer uso de correspondencias uno a uno, se enfrentan a un problema de mayor dificultad, que implica la revisión y actualización de todas las correspondencias con el resto de esquemas. En la propuesta, el uso de correspondencias entre el esquema y la ontología de referencia, simplifica esta tarea, proporcionando mejor soporte a la característica. Por estos motivos, los motores de metadatos, los motores de ontologías y los directorios de ontologías reciben el valor *Medio* y la propuesta recibe el valor *Alto*. El resto de candidatos recibe el valor *Ninguno*.
- **Lenguaje:** Los candidatos que no hacen uso de una representación de la semántica y los motores de metadatos, utilizan el lenguaje de definición del esquema, habitualmente XML o RDF. Los motores de ontologías, los directorios de ontologías y la propuesta hacen uso de lenguajes con mayor capacidad semántica, como OWL-Full. Los primeros reciben el valor *Medio*, los restantes el valor *Alto*.
- **Semántica:** Salvo la propuesta, el resto de candidatos no permite especificar el significado concreto del concepto a recuperar, por ejemplo en el caso de un término polisémico. Todos los candidatos reciben el valor *NO* a excepción de la propuesta, que recibe el valor *SÍ*.
- **Consulta conceptual:** Los motores XML, los directorios de esquemas y los directorios de metadatos basan la recuperación en la búsqueda sintáctica de etiquetas y atributos. Los motores de metadatos, los motores de ontologías y los directorios de ontologías amplían la búsqueda, incluyendo los significados y las relaciones entre conceptos. La propuesta incluye los significados y las relaciones entre conceptos en un ámbito global, al tiempo que permite establecer la semántica concreta del concepto a recuperar. Por estos motivos, los tres primeros candidatos reciben el valor *Bajo*, los tres siguientes *Medio*, y la propuesta recibe el valor *Alto*.
- **Consulta contextual:** Los motores XML, los directorios de esquemas y los directorios de metadatos no amplían los resultados en base al contexto del concepto, al no tratar las relaciones entre términos. Los motores de metadatos, los motores de ontologías y los directorios de ontologías amplían los resultados en base a correspondencias uno a uno. La propuesta permite ampliar el contexto con las relaciones globales entre conceptos y las relaciones semánticas

establecidas entre los conceptos de la ontología de referencia, por ejemplo la generalización. Por estos motivos, los tres primeros candidatos reciben el valor *Ninguno*, los tres siguientes *Medio*, y la propuesta recibe el valor *Alto*.

- **Recuperación de documentos:** Los directorios, de esquemas, metadatos y ontologías, permiten recuperar esquemas sin incluir documentos. A diferencia de los anteriores, los motores, XML, de metadatos y de ontologías, sí permiten recuperar documento, en un ámbito más o menos restringido. La propuesta también permite la recuperación de documentos, generados a partir de los esquemas incluidos en el sistema, y recopilados previamente por un recuperador/indizador Web. De este modo, los directorios reciben el valor *NO* y el resto de métodos el valor *SÍ*.

A modo de resumen, en la siguiente tabla se incluyen los valores obtenidos por cada método durante la evaluación de cada característica:

Característica	Motores XML	Directorios Esquemas	Motores Metadatos	Directorios Metadatos	Motores Ontologías	Directorios Ontologías	Propuesta
Interoperatividad	NO	NO	SÍ	NO	SÍ	SÍ	SÍ
Formalización	NO	NO	NO	NO	NO	NO	SÍ
Interactividad	Medio	Alto	Alto	Bajo	Bajo	Alto	Alto
Desambiguación	NO	NO	NO	NO	NO	NO	SÍ
Marco semántico	Ninguno	Ninguno	Medio	Ninguno	Medio	Medio	Alto
Plurilingüismo	NO	NO	NO	NO	NO	NO	SÍ
Sinonimia	Ninguno	Ninguno	Medio	Ninguno	Medio	Medio	Alto
Ámbito	Alto	Alto	Bajo	Bajo	Alto	Alto	Alto
Extensibilidad	Bajo	Bajo	Medio	Bajo	Medio	Medio	Alto
Reusabilidad	Ninguno	Ninguno	Ninguno	Ninguno	Ninguno	Medio	Alto
Modificabilidad	Ninguno	Ninguno	Medio	Ninguno	Medio	Medio	Alto
Lenguaje	Medio	Medio	Medio	Medio	Alto	Alto	Alto
Semántica	NO	NO	NO	NO	NO	NO	SÍ
Consulta conceptual	Bajo	Bajo	Medio	Bajo	Medio	Medio	Alto
Consulta contextual	Ninguno	Ninguno	Medio	Ninguno	Medio	Medio	Alto
Recuperación de documentos	SÍ	NO	SÍ	NO	SÍ	NO	SÍ

Tabla IV-7: Resumen de la valoración de los métodos candidatos.

Una vez evaluados los métodos, en el siguiente apartado se presentarán y discutirán los resultados obtenidos.

4.2.4 Presentación, análisis y discusión de resultados

Partiendo del marco de evaluación definido y de la valoración de cada método, incluidos en los apartados anteriores, se ha procedido al cálculo de los resultados del proceso.

El proceso de cálculo se ha realizado del siguiente modo: para cada característica se ha obtenido el producto del valor asignado en la evaluación por el factor de importancia

establecido para dicha característica. Por ejemplo, los motores XML tienen un valor de 18 en la característica de interactividad, resultado de multiplicar 3 puntos obtenidos en la evaluación por 6 puntos, correspondientes a la importancia (deseable) de la característica.

Una vez calculados los valores ponderados de cada método, y para facilitar su interpretación, se han calculado los porcentajes agregados por categoría. Concretamente, para cada una de las categorías definidas: Gestión de Esquemas, Gestión Semántica y Consulta, se han sumado los valores de sus características y se ha calculado el porcentaje de dicho sumatorio sobre el valor máximo posible, que se correspondería con el 100%. Por ejemplo, los motores XML tienen un porcentaje en la Gestión de Esquemas igual a 13,8%, valor que proviene de la suma de los valores ponderados para las características: Interoperatividad, Formalización e Interactividad, frente al valor máximo que podría obtenerse, en este caso igual a 130.

A continuación se muestran, agrupados por categoría, los gráficos correspondientes a la evaluación de cada método.

En la siguiente figura se muestran los resultados de la evaluación de los métodos para la categoría Gestión de Esquemas.

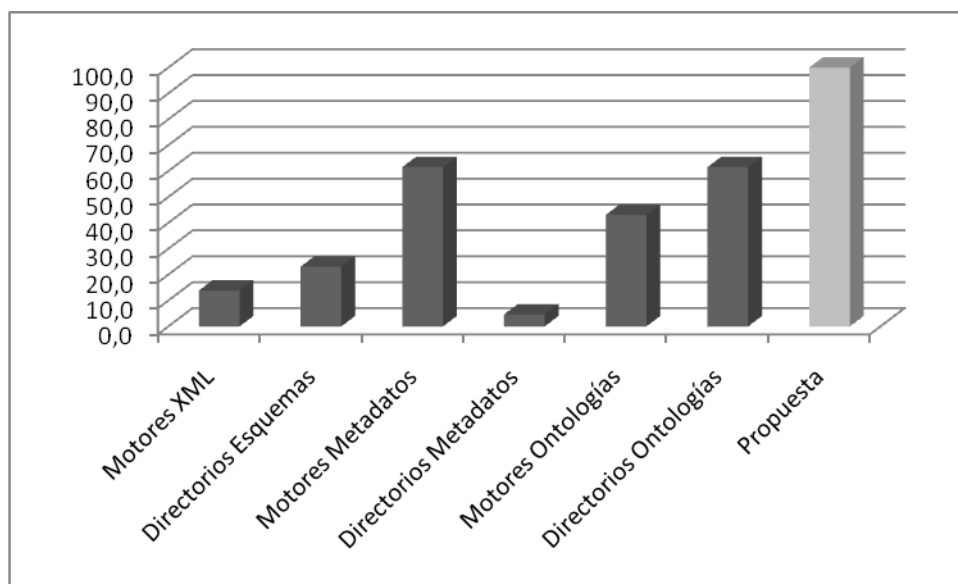


Figura IV-28. Resultados de la evaluación de los métodos candidatos para la categoría Gestión de Esquemas.

Como se puede observar en la figura, para la categoría de Gestión de Esquemas, la propuesta obtiene un resultado notablemente superior frente al resto de métodos evaluados. El resultado de la propuesta es lógico, teniendo en cuenta que los aspectos evaluados son abordados en el planteamiento con un carácter fundamental. Los motores de metadatos y los directorios de ontologías obtienen buenos resultados, principalmente gracias a que promueven la participación del usuario y a que soportan la definición de relaciones entre elementos de los esquemas. Los motores de ontologías se encuentran por debajo de los anteriores, debido a su menor capacidad de interactividad con el usuario. Los métodos restantes obtienen valores sensiblemente inferiores a causa de la

falta de soporte al establecimiento de correspondencias entre elementos y una menor interactividad con el usuario.

En la siguiente figura se muestran los resultados de la evaluación de los métodos para la categoría Gestión Semántica.

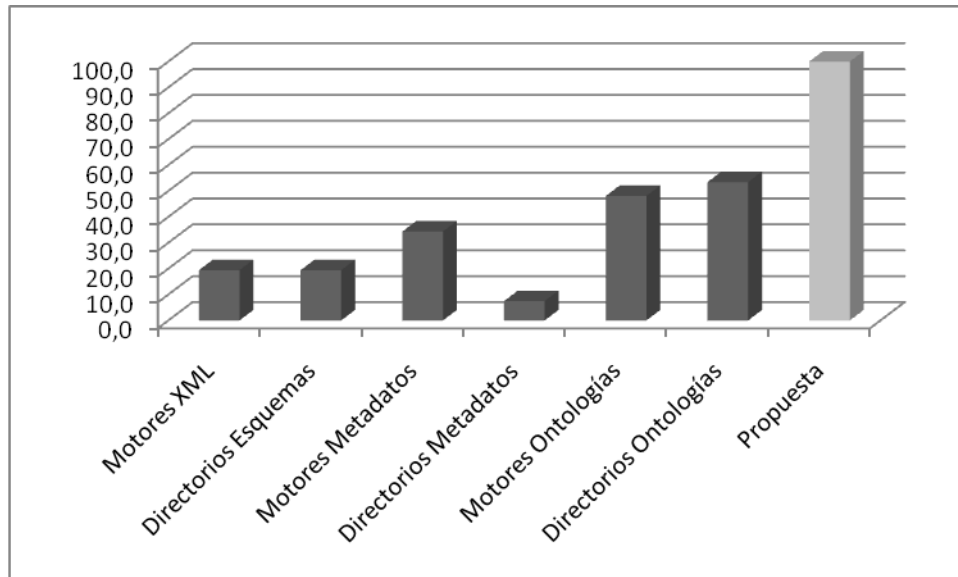


Figura IV-29. Resultados de la evaluación de los métodos candidatos para la categoría Gestión Semántica.

Como se puede observar en la figura, para la categoría de Gestión Semántica, la propuesta obtiene, de nuevo, un resultado notablemente superior frente al resto de métodos evaluados. El resultado de la propuesta se debe a que cubre los aspectos evaluados, con un carácter fundamental, obteniendo así la máxima valoración en todos ellos. En orden descendente, los siguientes valores son obtenidos por los métodos que hacen uso de representaciones ontológicas, lógicamente, puesto que la categoría incluye características relacionadas con la gestión de la semántica. Así, los directorios de ontologías y los motores de ontologías obtienen buenos resultados, gracias a: la gestión de relaciones entre conceptos; su ámbito de aplicación, heterogéneo en cuanto al dominio del conocimiento; la modificabilidad de la solución y el lenguaje de representación semántica utilizado. El siguiente valor es el obtenido por los motores de metadatos, cuyo descenso está motivado por su ámbito, más restringido, y por el uso de lenguajes con menor capacidad semántica de representación. Los siguientes resultados, en orden descendente, son los correspondientes a los motores XML y a los directorios de esquemas. En ambos casos, el resultado se debe a carencias en la gestión de la semántica y menor capacidad de extensibilidad y modificabilidad. Los directorios de metadatos obtienen el valor más bajo debido, principalmente, a su ámbito de aplicación, más restringido.

En la siguiente figura se muestran los resultados de la evaluación de los métodos para la categoría Consulta.

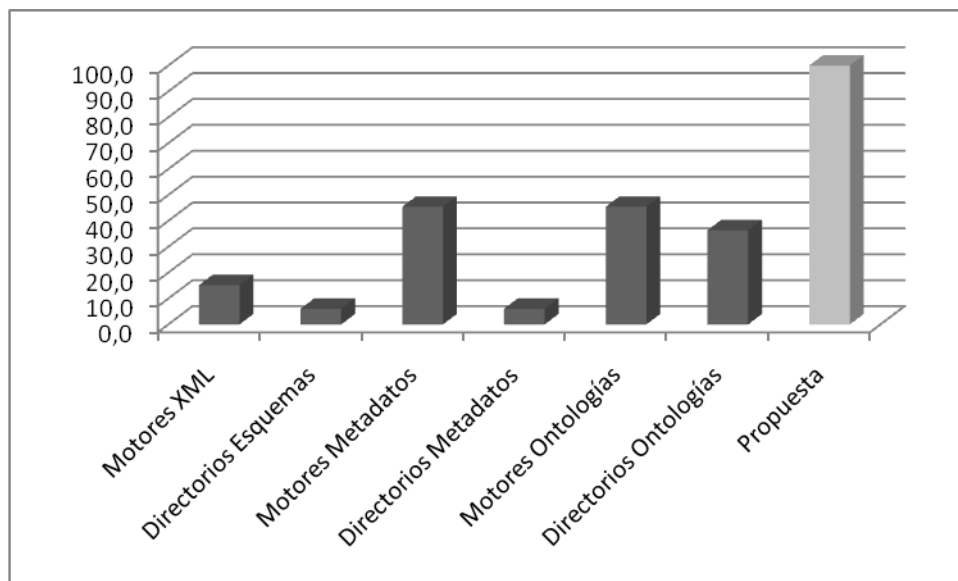


Figura IV-30. Resultados de la evaluación de los métodos candidatos para la categoría Consulta.

Como se puede observar en la figura, para la categoría de Consulta, la propuesta vuelve a obtener el mejor resultado respecto al resto de métodos evaluados. De nuevo, los aspectos evaluados son cubiertos completamente, al formar parte de los objetivos establecidos en su planteamiento. En orden descendente, los siguientes valores son obtenidos por los motores de ontologías y los motores de metadatos debido, principalmente, a la capacidad de realizar consultas conceptuales contextualizadas, así como a la posibilidad de obtener documentos semánticos. El siguiente valor se corresponde con los directorios de ontologías, cuyo descenso se debe a la imposibilidad de obtener documentos asociados a los esquemas. El descenso en los valores restantes está causado por la carencia de contextualización de los resultados y un uso local de la semántica de los esquemas. Concretamente, los motores XML obtienen un valor algo superior gracias a la posibilidad de obtener documentos asociados a los esquemas. Las carencias en esta última característica provocan que los directorios de esquemas y los directorios de metadatos obtengan los valores más bajos.

Para finalizar la comparativa, se ha calculado la evaluación completa de cada método. Con el fin de poder comparar los candidatos desde un punto de vista general, se ha calculado el valor de cada método como el porcentaje del candidato frente al máximo posible en la ponderación. En la siguiente figura se muestran los resultados finales de la comparativa.

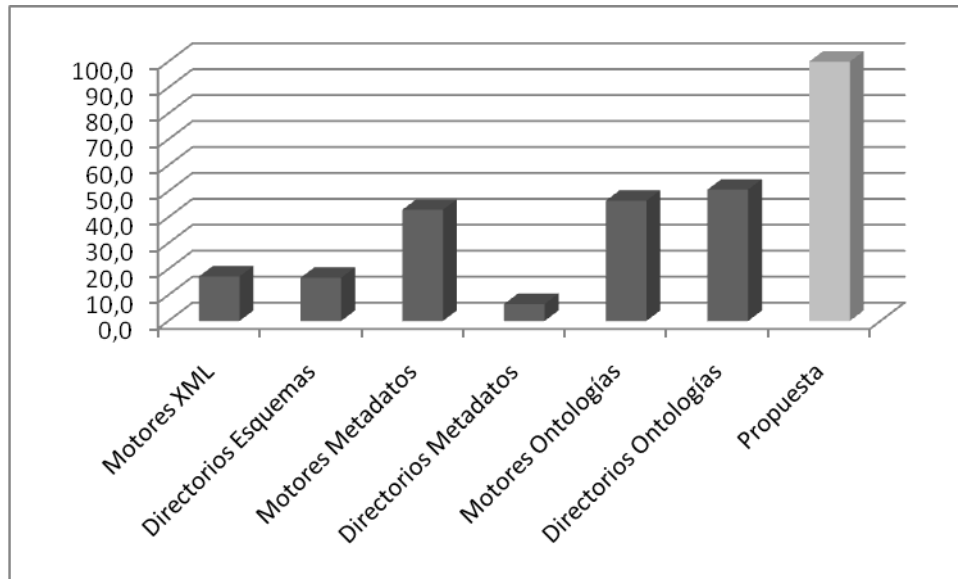


Figura IV-31. Comparativa final entre los métodos candidatos.

Tal y como muestra la figura, la propuesta obtiene un resultado notablemente superior en la comparativa final entre los métodos candidatos. El valor obtenido proviene de los resultados parciales, igualmente superiores, conseguidos en las tres categorías evaluadas. El siguiente valor, sensiblemente inferior, es el obtenido por los directorios de ontologías. En este caso, como resultado de una evaluación positiva en el soporte a las categorías de gestión de esquemas y gestión semántica. Siguiendo en orden descendente, los motores de ontologías han obtenido el siguiente valor. Este resultado se debe a la evaluación positiva del método en las consultas y en la gestión de la semántica. En el caso de los motores de metadatos, la valoración obtenida proviene de compensar la gestión semántica con una buena gestión de esquemas y consultas. Los resultados de los motores XML y los directorios de esquemas, sensiblemente inferiores a los anteriores, son debido a carencias en la gestión semántica y consultas. En último lugar se encuentran los directorios de metadatos, debido a carencias en el soporte a todas las categorías evaluadas.

Capítulo V: Conclusiones

Los principales problemas que presentan actualmente los esquemas de metadatos, se resumen en la necesidad de que el usuario tenga un conocimiento previo del esquema para poder utilizarlo. De este modo, tanto las personas como las computadoras, precisan conocer la existencia y localización del esquema, así como su estructura, recomendaciones de uso y, lo más importante, el significado de sus términos. Aunque algunos de estos requisitos pueden ser satisfechos mediante búsquedas, registros de metadatos y documentación de los esquemas, el aspecto semántico de los términos sigue siendo un reto de primer orden. La principal dificultad que presentan las definiciones de los términos es el ámbito en el cual se definen, típicamente restringido al ámbito de una entidad u organización, en el mejor de los casos. La carencia de una semántica compartida en la definición de los elementos conlleva dificultades en la reutilización de esquemas y problemas de interoperatividad entre los mismos, además de imposibilitar la recuperación conceptual de términos, relegándose ésta a las comunes búsquedas sintácticas. El resultado suele ser el desarrollo de nuevos esquemas, en muchos casos *ad-hoc* y escasamente formalizados, frente a lo que sería deseable, la reutilización de los ya existentes.

Para paliar estos problemas, este trabajo presenta una propuesta consistente en dotar a los esquemas de metadatos de un plano semántico, permitiendo así la asignación de semántica y desambiguación de los términos, mediante una estructura conceptual flexible. A través de una ontología de referencia, se consigue un mapa conceptual y relacional que incluye y permite equivalencias léxicas entre los términos de los documentos y la estructura ontológica general. De este modo, primeramente se define un modo de representar esquemas que permita la gestión de su semántica y cubra aspectos relacionados como sinonimias y plurilingüismo. Seguidamente se selecciona la ontología de referencia para, a continuación, establecer una metodología de correspondencia entre los conceptos incluidos en los esquemas y los incluidos en la ontología de referencia. Finalmente, se lleva a cabo el proceso de correspondencia de un conjunto de esquemas seleccionado. Tal y como se plantea en la propuesta, la ontología de referencia realiza las funciones de diccionario general mientras que los esquemas de metadatos hacen las funciones de diccionarios especializados, superando las limitaciones de los aspectos fraseológicos y las irregularidades en cuanto a la información en las definiciones. Además de servir como vocabulario controlado y proporcionar significado a los elementos, a través de la ontología de referencia se consigue un mapa conceptual y relacional que incluye y permite equivalencias léxicas entre los términos de los esquemas y la estructura ontológica general.

Las principales ventajas de la propuesta parten del hecho de no presuponer ningún conocimiento de la estructura del esquema por parte del usuario, al darle la posibilidad de realizar no sólo consultas conceptuales sobre un repertorio de esquemas, sino también de incluir sus propios esquemas y documentos en dicho repositorio. Para

lograrlo, se ha desarrollado tanto la metodología como los recursos para la interconexión de los términos incluidos en los esquemas. De este modo, la propuesta incluye tanto el marco teórico, como la metodología para la cualificación semántica de esquemas y su correspondencia con la ontología de referencia, todo ello verificado y validado por ejemplos de aplicación. Además, se incluye la especificación del sistema informático que soportará el proceso de interconexión y gestión, tanto de esquemas como de ontologías.

Respecto a los objetivos planteados, la propuesta logra:

- Desambiguar la representación y definición de los términos incluidos en esquemas, mediante la asignación de una semántica compartida, lo que permite mejorar la interoperatividad de esquemas de metadatos heterogéneos, a través de la definición de una semántica común. El consenso de las definiciones se alcanza mediante la colaboración de expertos e ingenieros y es soportado por un recurso Web colaborativo que permite al usuario, además, incluir y gestionar modelos.
- El uso de una semántica común mejora la reutilización de esquemas, al eliminar la necesidad de un conocimiento previo de los mismos y sus elementos. Además, se elimina la ambigüedad en las definiciones, a la vez que se mejora la interpretabilidad tanto por personas como por computadoras.
- Se posibilita la recuperación de información más completa, haciendo uso del contexto que proporciona la ontología de referencia. De este modo se permite la navegación entre conceptos incluidos en la ontología, a través de las relaciones establecidas en ella.
- Se posibilita la recuperación conceptual de documentos semánticos, incluyendo búsquedas por el contenido conceptual, así como búsquedas por esquemas y vocabularios de metadatos con un concepto determinado.
- A través de la representación semántica de los esquemas, se elimina la ambigüedad semántica de los elementos y se formaliza su representación. Además, se proporcionan nuevas capacidades de gestión basadas en la semántica, por ejemplo, inferencia.
- La propuesta está sirviendo como base teórica para el desarrollo un sistema informático que la soporta, y que incluye como objetivos: recopilar y almacenar un Corpus para compartir los esquemas más populares; incorporar metadatos en la representación ontológica de los esquemas que serán utilizados tanto en la búsqueda como por un algoritmo de posicionamiento; permitir al usuario incorporar sus propios esquemas, que en última instancia serán validados por un sistema de votación de expertos; desarrollar un motor de recuperación que permita la recuperación semántica de documentos semánticos y Esquemas asociados, mediante una interfaz Web de acceso libre. Se han publicado todos los entregables generados en el desarrollo hasta el momento.

Respecto a las características de la solución propuesta, cumple los siguientes requisitos expuestos en los objetivos iniciales y que suponen una mejora sustantiva a los problemas indicados inicialmente:

- Es aplicable a cualquier conjunto de esquemas.
- Es compatible con sistemas que no soporten la representación semántica del esquema, al posibilitar la aplicación de principios de simplificación que permitan el uso del esquema original.
- En el diseño de la solución, se han seguido las recomendaciones propuestas por las principales entidades de estandarización en este ámbito, como el DCMI o el W3C.
- Se permite la reutilización de elementos en el futuro, mediante definiciones consensuadas, no ambiguas, comprensibles y accesibles.
- Se han generado metodologías, plantillas y procedimientos que permiten la gestión de los esquemas semánticamente cualificados, y que comprende desde su generación, *3.1.1 Definición de la representación semántica de esquemas*, hasta su alineamiento contra la ontología de referencia, *3.1.3.2 Representación ontológica de la semántica de los esquemas seleccionados*.
- Se soportan criterios de modularidad, extensibilidad, refinamiento y plurilingüismo.
- Se han aplicado las metodologías, procedimientos y plantillas propuestos, con el fin de ejemplificar, validar y verificar la propuesta.
- Se permite la desambiguación de los términos que incluyen múltiples significados dependientes del contexto, la desambiguación de términos distintos con el mismo significado y los términos definidos en múltiples idiomas.
- Se permite el uso de múltiples ontologías simultáneas como ontología de referencia. De este modo, un elemento puede tener asignada semántica proveniente de distintas ontologías.
- Se facilita el uso y reutilización de los artefactos generados, mediante el uso de sistemas de identificación formal y su publicación en Web.

Con lo expuesto, la propuesta proporciona una posible solución a la incorporación de semántica en recursos Web, de forma extensible, compatible y reutilizable, que sienta las bases para la desambiguación de futuros formalismos de representación de conocimiento.

Capítulo VI: Trabajos Futuros

Como primera propuesta para futuros trabajos, estaría la experimentación y evaluación cuantitativa de los requisitos. Para ello sería preciso el desarrollo de la plataforma de soporte diseñada. Actualmente el desarrollo se encuentra en curso, como parte de un proyecto del Plan Nacional de I+D+i.

Además de la experimentación, el sistema de soporte proporcionará información útil para la evaluación de las interacciones de los usuarios con el sistema.

En el campo de los esquemas de metadatos, el sistema informático permitiría evaluar la interoperatividad entre esquemas semánticamente cualificados. Esto podría suponer un avance importante en el ámbito de la interoperatividad de esquemas.

La reutilización de la base ontológica generada en la elaboración de estudios y el desarrollo de aplicaciones, permitiría reaprovechar el esfuerzo invertido en esta propuesta, al tiempo que podría proporcionar una realimentación a la misma. En relación con su mantenimiento, podría desarrollarse un sistema supervisado que facilitara la incorporación y actualización de esquemas y ontologías.

La ampliación del método propuesto a otros formalismos de representación, como bases de datos, diagramas de clases, etc., permitiría obtener las ventajas de la cualificación semántica en modelos más complejos de representación del conocimiento.

En aspectos más concretos del planteamiento, sería de gran interés evaluar el uso de algoritmos de correspondencia automáticos o semiautomáticos, con el fin de reducir el esfuerzo de generación de alineamientos o valorar la calidad de las correspondencias generadas por dichos algoritmos. En esta línea, también podrían definirse correspondencias más complejas o hacer uso de plataformas para su gestión. En este ámbito se encontraría también la definición de búsquedas más complejas a las propuestas, que permitieran la especificación de criterios complejos y sacaran más partido a la base ontológica generada, así como el uso de algoritmos de posicionamiento avanzados, aplicados también a la recuperación.

En relación con el desarrollo de la plataforma destacar que, al hacer uso de tecnologías punteras en el ámbito de la informática y la gestión de ontologías, sería recomendable realizar evaluaciones periódicas del estado de la cuestión en dichos ámbitos. El resultado de estas evaluaciones podría revertir en una mejora continua de la calidad del sistema informático de soporte, dentro de su fase de mantenimiento.

Capítulo VII: Bibliografía

- Adie, C. (2007). IS Support for Web 2.0 Services: the Next Steps. Disponible en: <http://www.vp.is.ed.ac.uk/content/1/c4/12/25/Web2.0ActionPlan20070531.pdf>. [Consultado en junio de 2010].
- American Library Association (2010). American Library Association. Disponible en: <http://www.ala.org/>. [Consultado en junio de 2010].
- American National Standards Institute. National Information Standards Organization (1992). Application Service Definition and Protocol Specification for Open Systems Interconnection. American National Standards Institute. National Information Standards Organization. Disponible en: <http://www.cni.org/pub/NISO/docs/Z39.50-1992/>. [Consultado en junio de 2010].
- Amsterdam University (1999). EuroWordnet: Building a multilingual database with wordnets for several European languages. Disponible en: <http://www.illc.uva.nl/EuroWordNet/>. [Consultado en junio de 2010].
- Ankolekar, A.; Krötzsch, M.; Tran, T.; Vrandečić, D. (2007). The Two Cultures: Mashing up Web 2.0 and the Semantic Web. *16th International World Wide Web Conference, 8-12 May. Banff, Alberta, Canada*. New York, USA: ACM Press, pp. 10. ISBN: 978-1-59593-654-7. Disponible en: <http://www2007.org/papers/paper777.pdf>. [Consultado en junio de 2010].
- Apache Software Foundation (2007). Apache Xindice. Disponible en: <http://xml.apache.org/xindice/>. [Consultado en junio de 2010].
- Aquin, M.; Sabou, M.; Džbor, M.; Baldassarre, C.; Angeletou, S.; Motta, E. (2007). WATSON : A Gateway for the Semantic Web. *Poster session. European Semantic Web Conference, 3-7 Jun. Innsbruck, Austria*. Disponible en: http://watson.kmi.open.ac.uk/DownloadsAndPublications_files/iswc07.pdf. [Consultado en junio de 2010].
- Ariadne Training (2005). *UML Applied - Second Edition. Object Oriented Analysis and Design*. Ariadne Training Limited, pp. 235. Disponible en: [http://www.ariadnetraining.co.uk/resources/UML Applied Second Edition.pdf](http://www.ariadnetraining.co.uk/resources/UML%20Applied%20Second%20Edition.pdf). [Consultado en junio de 2010].
- Ayensa, I.; Calvo, J.; Gárate, F.J.; López, D. (2010). Sistema de Gestión de Conocimiento Basado en Ontologías: Documento de Requisitos Software. Disponible en: http://semse.kr.inf.uc3m.es/SEMSE/index.php?option=com_docman&task=doc_download&gid=16&Itemid=39. [Consultado en junio de 2010].

Ayensa, I.; Calvo, J.; Gárate, F.J.; López, D. (2010). Sistema de Gestión de Conocimiento Basado en Ontologías: Documento de Requisitos de Usuario. Disponible en:

http://semse.kr.inf.uc3m.es/SEMSE/index.php?option=com_docman&task=doc_download&gid=15&Itemid=39. [Consultado en junio de 2010].

Baeza, R.; Ribeiro, B. (1999). *Modern information retrieval*. New York, USA: ACM Press, pp. 506. ISBN: 0-201-39829-X.

Bao, Z.; Ling, T.W.; Chen, B.; Lu, J. (2009). Effective xml keyword search with relevance oriented ranking. *Proceedings of the 2009 IEEE International Conference on Data Engineering*. Washington, USA: IEEE Computer Society, pp. 517-528. ISBN: 978-0-7695-3545-6.

Bechhofer, S.; Harmerlen Van, F.; Hendler, J.; Horrocks, I.; McGuinness, D.L.; Stein, L.A.; Patel-Schneider, P.F. (2004). OWL Web Ontology Language Reference. Disponible en: <http://www.w3.org/TR/owl-ref/>. [Consultado en junio de 2010].

Berners-Lee, T.; Hendler, J.; Lassila, O. (2001). The Semantic Web. *Scientific American*, **284**(5), pp. 34-43. ISSN: 0036-8733.

Berners-Lee, T. (1998). A roadmap to the Semantic Web. Disponible en: <http://www.w3.org/DesignIssues/Semantic.html>. [Consultado en junio de 2010].

Berners-Lee, T. (2007). Personal Information Markup. Disponible en: <http://www.w3c.org/2000/10/swap/pim/contact>. [Consultado en junio de 2010].

Berners-Lee, T. (1994). Universal Resource Identifiers in WWW. Disponible en: <http://www.ietf.org/rfc/rfc1630.txt>. [Consultado en junio de 2010].

Berrueta, D.; Phipps, J. (2008). Best Practice Recipes for Publishing RDF Vocabularies. Disponible en: <http://www.w3.org/TR/swbp-vocab-pub/>. [Consultado en junio de 2010].

Bitext Innovations (2010). Bitext.com - Tecnologías de Lenguaje Natural. Disponible en: <http://www.bitext.com/>. [Consultado en junio de 2010].

Bonilla, S. (2007). Web Semántica y Agentes Metarrepresentacionales basados en Marcadores Discursivos. *Hipertext.net*, (5), pp. 10. ISSN: 1695-5498. Disponible en: <http://www.hipertext.net/web/pag272.htm>. [Consultado en junio de 2010].

Botts, N. (2008). Learning About Learning Objects. Disponible en: <http://www.learning-objects.net/>. [Consultado en junio de 2010].

Braude, E.J. (2000). *Software Engineering: An Object-Oriented Perspective*. New York, USA: John Wiley & Sons Inc, pp. 535. ISBN: 0-471-32208-5.

Bremen University (2010). Generalized Upper Model 3.0. Disponible en: <http://www.ontospace.uni-bremen.de/ontology/gum.html>. [Consultado en junio de 2010].

Brickley, D.; Miller, L. (2010). FOAF Vocabulary Specification 0.97. Disponible en: <http://xmlns.com/foaf/spec>. [Consultado en junio de 2010].

- Brickley, D. (2009). Term-centric Semantic Web Vocabulary Annotations. Disponible en: <http://www.w3.org/2003/06/sw-vocab-status/note>. [Consultado en junio de 2010].
- Buscaldi, D.; Guerrini, G.; Mesiti, M.; Rosso, P. (2003). Tag semantics for the retrieval of XML documents. *Proceedings of the 1st international symposium on Information and communication technologies, 24-26 Sep. Dublin, Ireland*. Dublin, Ireland: Trinity College Dublin, pp. 273-278. ISBN: 1-59593-243-7.
- Buzztrend (2010). Swotti Opiniones Terra. Disponible en: <http://opiniones.terra.es/>. [Consultado en junio de 2010].
- CES (2003). MetaNet: Página principal. *Centre for Educational Sociology. University of Edinburgh*. Disponible en: <http://www.epros.ed.ac.uk/metanet/>. [Consultado en junio de 2010].
- Castano, S.; Antonellis, V.D. (2001). Global viewing of heterogeneous data sources. *IEEE Transactions on Knowledge and Data Engineering*, **13**(2), pp. 277-297. ISSN: 1041-4347.
- Cheng, G.; Wu, H.; Ge, W.; Qu, Y. (2008). Searching Semantic Web Objects Based on Class Hierarchies. *World Wide Web Internet And Web Information Systems*. Amsterdam, The Netherlands: Springer-Verlag, pp. 199-226. ISSN: 1386-145X.
- Choi, N.; Song, I.; Han, H. (2006). A survey on ontology mapping. *ACM SIGMOD Record*, **35**(3), pp. 34-41. ISSN: 0163-5808.
- Christophides, V.; Karvounarakis, G.; Plexousakis, D.; Scholl, M.; Tourtounis, S. (2003). Optimizing taxonomic semantic web queries using labelling schemes. *Web Semantics: Science, Services and Agents on the World Wide Web*, **1**(2), pp. 207-228 Elsevier. ISSN: 1570-8268. Disponible en: <http://www.ics.forth.gr/isl/publications/paperlink/journalofwebsemantics2004.pdf>. [Consultado en junio de 2010].
- Cimiano, P.; Handschuh, S.; Staab, S. (2004). Towards the Self-annotating Web. *Proceedings of the 13th international conference on World Wide Web, 17-20 May. New York, USA*. New York, USA: ACM Press, pp. 462-471. ISBN: 1-58113-844-X.
- Ciravegna, F.; Dingli, A.; Petrelli, D.; Wilks, Y. (2002). User-System Cooperation in Document Annotation Based on Information Extraction. *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, EKAW02, 1-4 Oct. Siguenza, Spain*. London, UK: Springer-Verlag, pp. 122-137. ISBN: 3-540-44268-5. Disponible en: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.519>. [Consultado en junio de 2010].
- Codina, L.; Rovira, C. (2006). La Web Semántica. *Tendencias en documentación digital*. Gijón, Spain: TREA, (1), pp. 9-54. ISBN: 84-9704-270-0.

- Corbera, S. (2007). Estudio sobre sistemas de gestión de conocimiento para la web semántica. Disponible en: http://semse.kr.inf.uc3m.es/SEMSE/index.php?option=com_docman&task=doc_download&gid=14&Itemid=39. [Consultado en junio de 2010].
- Corcho, O.; Fernández-López, M.; Gómez-Pérez, A. (2003). Methodologies, tools and languages for building ontologies: where is their meeting point? *Data & Knowledge Engineering*, **46**(1), pp. 41-64. ISSN: 0169-023X.
- Cuevas, G.; Amescua, A.; Cerrada, J.A.; San Feliu, T.; Calvo-Manzano, J.; Arcilla, M.; García, M. (2003). *Gestión del Proceso Software*. Editorial Universitaria Ramón Areces, pp. 472. ISBN: 84-8004-546-9.
- Cycorp (2010). Cycorp: Página principal. Disponible en: <http://www.cyc.com/>. [Consultado en junio de 2010].
- Cycorp (2009). OpenCyc.org: Formalized Common Knowledge. Disponible en: <http://www.opencyc.org/>. [Consultado en junio de 2010].
- Cycorp (2009). ResearchCyc: Formalized Common Knowledge. Disponible en: <http://research.cyc.com/>. [Consultado en junio de 2010].
- DAML.org (2004). DAML Ontology Library. Disponible en: <http://www.daml.org/ontologies/>. [Consultado en junio de 2010].
- DTSC (1998). HotMeta. Disponible en: <http://www.dstc.edu.au/1024.html>. [Consultado en junio de 2010].
- Daconta, M.C.; Obrst, L.J.; Smith, K.T. (2003). *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*. Indianapolis, USA: John Wiley & Sons Inc, pp. 281. ISBN: 0-471-43257-1.
- Dawson, F.; Howes, T. (1998). rfc 2426. Disponible en: <http://www.apps.ietf.org/rfc/rfc2426.html>. [Consultado en junio de 2010].
- Dawson, F.; Howes, T. (1998). vCard MIME Directory Profile. Disponible en: <ftp://ftp.isi.edu/in-notes/rfc2426.txt>. [Consultado en junio de 2010].
- Day, M. (2002). Metadata: Mapping between metadata formats. *UKOLN: UK Office for Library and Information Networking*. Disponible en: <http://www.ukoln.ac.uk/metadata/>. [Consultado en junio de 2010].
- Defense Advanced Research Projects Agency (2002). DAML+OIL (March 2001). Disponible en: <http://www.daml.org/2001/03/daml+oil-index.html>. [Consultado en junio de 2010].
- Defense Information Systems Agency (2010). DoD Metadata Registry and Clearinghouse. Disponible en: <https://metadata.dod.mil/mdr/homepage.htm>. [Consultado en junio de 2010].

Derksen, E.; Fankhauser, P.; Howland, E.; Huck, G.; Macherius, I.; Murata, M.; Resnik, M.; Schöning, H. (1999). XQL (XML Query Language). Disponible en: <http://www.ibiblio.org/xql/xql-proposal.html>. [Consultado en junio de 2010].

Digital Digital Enterprise Research Institute (2010). SIOC Project: Semantically-Interlinked Online Communities. NUI Galway. Disponible en: <http://sioc-project.org/>. [Consultado en junio de 2010].

Dill, S. et al. (2003). SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. *Proceedings of the 12th international conference on World Wide Web, 20-24 May. Budapest, Hungary*. New York, USA: ACM Press, pp. 23. ISBN: 1-58113-680-3. Disponible en: <http://www2003.org/cdrom/papers/refereed/p831/p831-dill.html>. [Consultado en junio de 2010].

Ding, L.; Zhou, L.; Finin, T.; Joshi, A. (2005). How the Semantic Web is Being Used: An Analysis of FOAF Documents. *Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05), 2-6 Ene. Hilton Waikoloa Village, Island of Hawaii*. Washington, USA: IEEE Computer Society, pp. 113. ISBN: 0-7695-2268-8-4.

Do, H.H.; Melnik, S.; Rahm, E. (2002). Comparison of schema matching evaluations. *NODE 2002 Web and Database-Related Workshops on Web, Web-Services, and Database Systems, 7-10 Oct. Erfurt, Germany*. London, UK: Springer-Verlag, p. 221–237. ISBN: 3-540-00745-8. Disponible en: <http://www.springerlink.com/index/4wbxb1jbw8w4ek9v.pdf>. [Consultado en junio de 2010].

Do, H.; Rahm, E. (2002). COMA: a system for flexible combination of schema matching approaches. *Proceedings of the 28th international conference on Very Large Data Bases, 20-23 Ago. Hong Kong, China*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 610-621. ISBN: 1-558-60869-9. Disponible en: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.19.9866&rep=rep1&type=pdf>. [Consultado en junio de 2010].

Doan, A.; Domingos, P.; Halevy, A.Y.; Sellis, T.; Mehrotra, S. (2001). Reconciling schemas of disparate data sources: a machine-learning approach. *ACM SIGMOD Record*, **30**(2), pp. 509-520. ISSN: 0163-5808. Disponible en: http://portal.acm.org/ft_gateway.cfm?id=375731&type=pdf&coll=GUIDE&dl=GUIDE&CFID=90690476&CFTOKEN=81577390. [Consultado en junio de 2010].

Docsoft (2010). Element Search Appliance. Disponible en: <http://www.docsoft.com/search/>. [Consultado en junio de 2010].

Dublin Core Metadata Initiative (2001). Namespace Policy for the Dublin Core Metadata Initiative (DCMI). Disponible en: <http://dublincore.org/documents/dcmi-namespace/>. [Consultado en junio de 2010].

Dublin Core Metadata Initiative (2003). Dublin Core Namespace. Disponible en: <http://dublincore.org/2003/03/24/dces>. [Consultado en junio de 2010].

- Dublin Core Metadata Initiative (2004). DCMI Abstract Model. Disponible en: <http://dublincore.org/documents/abstract-model/>. [Consultado en junio de 2010].
- Dublin Core Metadata Initiative (2008). Dublin Core Metadata Element Set v1.1. Disponible en: <http://dublincore.org/documents/dces/>. [Consultado en junio de 2010].
- Dublin Core Metadata Initiative (2008). Dublin Core Metadata Registry. Disponible en: <http://dublincore.org/dcregistry/>. [Consultado en junio de 2010].
- Dublin Core Metadata Initiative (2010). Dublin Core Metadata Initiative (DCMI). Disponible en: <http://dublincore.org/>. [Consultado en junio de 2010].
- Dumbill, E. (2008). DOAP Description of a Project. Disponible en: <http://trac.usefulinc.com/doap/>. [Consultado en junio de 2010].
- Duval, E.; Hodgins, W.; Sutton, S.; Weibel, S.L. (2002). Metadata Principles and Practicalities. *D-Lib Magazine*, 8(4). ISSN: 1082-9873. Disponible en: <http://www.dlib.org/dlib/april02/weibel/04weibel.html>. [Consultado en junio de 2010].
- Díaz Ortuño, P.M. (2003). Problemática y tendencias en la arquitectura de metadatos Web. *Anales de Documentación*, (6), pp. 35-58 Servicio de publicaciones. Universidad de Murcia. ISSN: 1575-2437. Disponible en: <http://revistas.um.es/analesdoc/article/viewFile/2051/2041>. [Consultado en junio de 2010].
- E-XMLMedia (2002). E-XMLMedia Products - Repository - XML XQuery Database. Disponible en: <http://e-xmlmedia.com/prod/repository.htm>. [Consultado en junio de 2010].
- EAD (2009). Encoded Archival Description. Disponible en: <http://www.loc.gov/ead/>. [Consultado en junio de 2010].
- EBiquity (2010). Owlir: Information Retrieval On The Semantic Web. Disponible en: <http://ebiquity.umbc.edu/project/html/id/20/>. [Consultado en junio de 2010].
- Ehrig, M.; Staab, S. (2004). QOM -- Quick Ontology Mapping. *Proceedings of the Semantic Web ISWC 2004. Third International Semantic Web Conference, 7-11 Nov. Hiroshima, Japan*. Lecture notes in computer science. Berlin, Germany: Springer-Verlag, pp. 683-697. ISBN: 3-540-29754-5. Disponible en: www.aifb.uni-karlsruhe.de/WBS/meh/.../ehrig04qomISWC.pdf. [Consultado en junio de 2010].
- Ehrig, M.; Sure, Y. (2004). Ontology Mapping - An Integrated Approach. *The Semantic Web: Research and Applications*. Berlin, Germany: Springer-Verlag, pp. 76-91. ISBN: 978-3-540-21999-6. Disponible en: <http://www.springerlink.com/content/t3m6yjdeb99ft7ak>. [Consultado en junio de 2010].
- European Bioinformatics Institute (2010). SBO: Systems Biology Ontology. Disponible en: <http://www.ebi.ac.uk/sbo/>. [Consultado en junio de 2010].
- European Space Agency (2010). ESA Portal. Disponible en: <http://www.esa.int/esaCP/index.html>. [Consultado en junio de 2010].

Euzenat, J.; Shvaiko, P. (2007). *Ontology Matching*. Berlin, Germany: Springer-Verlag, pp. 334. ISBN: 978-3-540-49611-3.

Euzenat, J.; Valtchev, P. (2004). Similarity-based ontology alignment in OWL-Lite. *Proceedings of 16th European Conference on Artificial Intelligence (ECAI), 22-27 Ago., Valencia, Spain*. Amsterdam, The Netherlands: IOS Press, pp. 333-337. ISBN: 1-58603-452-9. Disponible en: <http://www.iro.umontreal.ca/~owlola/pdf/align-ECAI04-FSub.pdf>. [Consultado en junio de 2010].

Evri Inc. (2010). Fuel your fascination - News. Disponible en: <http://www.evri.com/>. [Consultado en junio de 2010].

FOAF-project (2010). The friend of a friend (FOAF) project. Disponible en: <http://www.foaf-project.org/>. [Consultado en junio de 2010].

Fernández-Tostado, T. (2009). Estudio Tecnológico y Diseño Arquitectónico de un Sistema de Gestión de Esquemas Semánticos basados en Ontologías. Disponible en: http://semse.kr.inf.uc3m.es/SEMSE/index.php?option=com_docman&task=doc_download&gid=17&Itemid=39. [Consultado en junio de 2010].

Forschungszentrum Informatik - Institut für Angewandte Informatik und Formale Beschreibungsverfahren (AIFB) (2001). KAON: The Karlsruhe ONtology and Semantic Web tool suite. Disponible en: <http://kaon.semanticweb.org/>. [Consultado en junio de 2010].

Franz Inc. (2010). AllegroGraph RDFStore Web 3.0's Database. Disponible en: <http://www.franz.com/agraph/allegrograph/>. [Consultado en junio de 2010].

Fuentes, D.; Gómez, J.; García, C. (2008). CoolWikiNews: More than meets the eye in the XXI century journalism. *Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications*. New York, USA: Information Science Reference, pp. 348. ISBN: 978-1-59904-877-2.

Fuentes-Lorenzo, D.; Morato, J.; Gómez, J.M. (2009). Knowledge management in biomedical libraries: A semantic web approach. *Information Systems Frontiers*, **11**(4), pp. 471-480. ISSN: 1387-3326. Disponible en: <http://www.springerlink.com/content/x76v51q1856875t1/fulltext.pdf>. [Consultado en junio de 2010].

Fuhr, N.; Großjohann, K. (2000). XIRQL: An Extension of XQL for Information Retrieval. Disponible en: <http://www.haifa.il.ibm.com/sigir00-xml/final-papers/KaiGross/sigir00.html>. [Consultado en junio de 2010].

Fumero, A.; Roca, G. (2007). *Web 2.0*. Fundación Orange, pp. 136. Disponible en: http://www.fundacionauna.com/areas/25_publicaciones/WEB_DEF_COMPLETO.pdf. [Consultado en junio de 2010].

Gárate, F.J. (2010). Desarrollo de un Sistema de Gestión para la cualificación semántica de esquemas. Disponible en: http://semse.kr.inf.uc3m.es/SEMSE/index.php?option=com_docman&task=doc_download&gid=18&Itemid=39. [Consultado en junio de 2010].

Giunchiglia, F.; Shvaiko, P.; Yatskevich, M. (2004). S-Match: an Algorithm and an Implementation of Semantic Matching. *The Semantic Web: Research and Applications*. Berlin, Germany: Springer-Verlag, pp. 61-75. ISBN: 978-3-540-21999-6. Disponible en: <http://www.springerlink.com/content/vhu9mfhql6dveu94/fulltext.pdf>. [Consultado en junio de 2010].

Godby, C.J.; Young, J.A.; Childress, E. (2004). A Repository of Metadata Crosswalks. *D-Lib Magazine*, **10**(12). ISSN: 1082-9873. Disponible en: <http://www.dlib.org/dlib/december04/godby/12godby.html>. [Consultado en junio de 2010].

GOLD Community (2009). GOLD Community Website: GOLD. Disponible en: <http://www.linguistics-ontology.org/gold.html>. [Consultado en junio de 2010].

Gómez, R. (2003). La evaluación en recuperación de la información. *Hipertext.net*, (1). ISSN: 1695-5498. Disponible en: <http://www.hipertext.net/web/pag238.htm>. [Consultado en junio de 2010].

Gómez-Pérez, A.; Fernández-López, M.; Corcho, O. (2004). *Ontological engineering : with examples from the areas of knowledge management, e-commerce and the Semantic Web*. London, UK: Springer-Verlag, pp. 403. ISBN: 1-852-33551-3.

Gómez-Pérez, A.; Pareja-Lora, A.; Aguado, G.; Alvarez, I.; Plaza, R. (2002). OntoTag: A Semantic Web Page Linguistic Annotation Model. *Proceedings of the ECAI 2002 Workshop on Semantic Authoring, Annotation and Knowledge Markup, 22-26 Jul. Lyon. France*. Menlo Park, CA, USA: AAAI Press, p. 5. ISBN: 1-58113-449-5. Disponible en: <http://www.cs.vassar.edu/~ide/events/ParejaLora.pdf>. [Consultado en junio de 2010].

Google (2010). Google Squared. Disponible en: <http://www.google.com/squared>. [Consultado en junio de 2010].

Grau, B.; Horrocks, I.; Motik, B.; Parsia, B.; Patelschneider, P.; Sattler, U. (2008). OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web*, **6**(4), pp. 309-322. ISSN: 1570-8268. Disponible en: http://www.sciencedirect.com/science?_ob=MIimg&_imagekey=B758F-4TP1FC8-1-8K&_cdi=12925&_user=143961&_pii=S1570826808000413&_orig=search&_coverDate=11%2F30%2F2008&_sk=999939995&view=c&wchp=dGLzVlb-zSkzV&md5=a1b69e054ca9206c9f3369dfc0cf4742&ie=/sdarticle.pdf. [Consultado en junio de 2010].

Graves, A. (2005). Representación de Wordnet en RDF. Disponible en: <http://www.dcc.uchile.cl/~agraves/wordnet/index.html>. [Consultado en junio de 2010].

Green, S.J. (1998). Automated link generation: Can we do better than term repetition? *Computer Networks and ISDN Systems*, **30**(1-7), pp. 75-84. ISSN: 0169-7552.

Gruber, T. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, **5**(2), pp. 199-220. ISSN: 1042-8143.

- Guarino, N. (1998). Formal Ontology and Information Systems. *International Conference on Formal Ontology Information Systems, 6-8 Jun. Trento, Italy*. Amsterdam, The Netherlands: IOS Press, pp. 3-15. ISBN: 9051993994. Disponible en: <http://www.loa-cnr.it/Papers/FOIS98.pdf>. [Consultado en junio de 2010].
- Guo, L.; Shao, F.; Botev, C.; Shanmugasundaram, J. (2003). XRANK: Ranked Keyword Search over XML Documents. *Proceedings of the 2003 ACM SIGMOD international conference on Management of data, 9-12 Jun. San Diego, CA, USA*. New York, USA: ACM Press, pp. 16-27. ISBN: 1-58113-634-X. Disponible en: <http://www.cs.cornell.edu/People/jai/papers/XRank.pdf>. [Consultado en junio de 2010].
- Hakimpour, F.; Geppert, A. (2001). Resolving semantic heterogeneity in schema integration: an Ontology Based approach. *Proceedings of the international conference on Formal Ontology in Information Systems, 17-19 Oct. Ogunquit, Maine, USA*. New York, USA: ACM Press, pp. 297-308. ISBN: 1-58113-377-4. Disponible en: http://portal.acm.org/ft_gateway.cfm?id=505196&type=pdf&coll=GUIDE&dl=GUIDE&CFID=90697605&CFTOKEN=64014670. [Consultado en junio de 2010].
- Handsuh, S.; Staab, S.; Ciravegna, F. (2002). SCREAM: Semi-automatic CREAtion of Metadata. *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web, 1-4 Oct. Madrid, Spain*. London, UK: Springer-Verlag, pp. 358-372. ISBN: 3-540-44268-5.
- Harvard University (2003). RSS 2.0 Specification. Disponible en: <http://cyber.law.harvard.edu/rss/rss.html>. [Consultado en junio de 2010].
- Heery, R.M.; Patel, M. (2000). Application profiles: mixing and matching metadata schemas. *ARIADNE*, (25). Disponible en: <http://www.ariadne.ac.uk/issue25/app-profiles/>. [Consultado en junio de 2010].
- Heijst, G.v.; Schreiber, A.T.; Wielinga, B.J. (1997). Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies*, **46**(2), pp. 183-292. ISSN: 1071-5819.
- Hewett Research (2009). SVCC 2009 Semantic Web Intro. Disponible en: <http://www.hewettresearch.com/svcc2009/>. [Consultado en junio de 2010].
- Hewlett-Packard Development Company (2009). Jena: A Semantic Web Framework for Java. Disponible en: <http://jena.sourceforge.net/>. [Consultado en junio de 2010].
- Hewlett-Packard Development Company (2009). joseki - a sparql server for jena. Disponible en: <http://www.joseki.org/>. [Consultado en junio de 2010].
- Hewlett-Packard Labs (2003). Semantic Blogging Demonstrator. Disponible en: <http://www.semanticblogging.org/blojsom-hp/semnav.html>. [Consultado en junio de 2010].
- Hewlett-Packard Labs (2009). Semantic Web Research at HP Labs. Disponible en: <http://www.hpl.hp.com/semweb/>. [Consultado en junio de 2010].

Hodge, G. (2001). *Metadata Made Simpler*. Bethesda, MD, USA: NISO Press, pp. 16. ISBN: 1-880124-50-5. Disponible en: http://www.ncsi.iisc.ernet.in/raja/is214/214-2001-2002/Metadata_Bklt.pdf. [Consultado en junio de 2010].

Howarth, L.C. (2000). Creating a Metadata-Enabled Framework for Resource Discovery in Knowledge Bases. *Annual CAIS/ACSI conference. Dimensions of a Global Information Science, 27-29 May. Quebec, Canada*. Edmonton, Alberta, Canada: Canadian Journal of Information and Library Science, pp. 15. ISBN: 2-7605-0993-1. Disponible en: <http://www.slis.ualberta.ca/cais2000/howarth.htm>. [Consultado en junio de 2010].

Hu, W.; Qu, Y. (2008). Falcon-AO: A practical ontology matching system. *Web Semantics: Science, Services and Agents on the World Wide Web*, **6**(3), pp. 237-239. ISSN: 1570-8268. Disponible en: <http://portal.acm.org/citation.cfm?id=1412999>. [Consultado en junio de 2010].

Hunter, J. (2001). MetaNet - A Metadata Term Thesaurus to Enable Semantic Interoperability between Metadata Domains. *Journal of Digital Information*, **1**(8) A&M University, Texas. Disponible en: <http://journals.tdl.org/jodi/article/viewArticle/33/34>. [Consultado en junio de 2010].

Hyun Hee, K. (2005). ONTOWEB : Implementing an ontology-based web retrieval system : Knowledge management in Asia. *Journal of the American Society for Information Science and Technology*, **56**(11), pp. 1167-1176. ISSN: 1532-2882.

IBM (2004). alphaWorks : IBM Integrated Ontology Development Toolkit : Overview. Disponible en: <http://www.alphaworks.ibm.com/tech/semanticstk/>. [Consultado en junio de 2010].

IceSoft (2009). ICEfaces - Open Source Ajax, J2EE Ajax, JSF Java Framework. Disponible en: <http://www.icefaces.org/>. [Consultado en junio de 2010].

Indecs (2009). Interoperability of data in e-commerce systems. Disponible en: <http://www.interparty.org/what-was-INDECS.html>. [Consultado en junio de 2010].

Institut National de Recherche en Informatique et Automatique (2010). Alignment api. Disponible en: <http://alignapi.gforge.inria.fr/>. [Consultado en junio de 2010].

Intellidimension (2007). Semantic Web Search. Disponible en: <http://www.semanticwebsearch.com/>. [Consultado en junio de 2010].

International Council of Museums (2006). The CIDOC CRM. Disponible en: <http://cidoc.ics.forth.gr/>. [Consultado en junio de 2010].

International Organization for Standardization (2010). Framework for Metamodel Interoperability. Disponible en: <http://metadata-stds.org/19763/index.html>. [Consultado en junio de 2010].

International Organization for Standardization (1976). Guidelines for the establishment and development of multilingual thesauri. Disponible en: <http://unesdoc.unesco.org/images/0002/000205/020561EB.pdf>. [Consultado en junio de 2010].

International Organization for Standardization (2005). Metadata Registries (MDR) - Information Technology. Disponible en: <http://metadata-stds.org/11179/index.html>. [Consultado en junio de 2010].

International Organization for Standardization (2009). Metadata Registry Interoperability and Bindings (MDRIB). Disponible en: <http://metadata-stds.org/20944/index.html>. [Consultado en junio de 2010].

International Organization for Standardization (2004). Multimedia Content Description Interface (MPEG-7) - Information Technology. Disponible en: <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>. [Consultado en junio de 2010].

International Organization for Standardization (2002). Topic Maps. Information Technology. Document Description and Processing Languages (second edition). Disponible en: http://www1.y12.doe.gov/capabilities/sgml/sc34/document/0322_files/iso13250-2nd-ed-v2.pdf. [Consultado en junio de 2010].

International Organization for Standardization (2001). Unified Modeling Language (UML). Version 1.4.2. Information technology. Disponible en: <http://www.omg.org/spec/UML/1.4/>. [Consultado en junio de 2010].

IXIASOFT (2008). XML Database, XML search engine, XML content management. Disponible en: <http://www.ixiasoft.com/>. [Consultado en junio de 2010].

JBoss Community (2010). Hibernate - JBoss Community. Disponible en: <http://www.hibernate.org/>. [Consultado en junio de 2010].

Jackson, J. (2009). Tim Berners-Lee: Machine-readable Web still a ways off. *Government Computer News*, pp. 3. Disponible en: <http://gcn.com/articles/2009/10/30/berners-lee-semantic-web.aspx>. [Consultado en junio de 2010].

Jasper, R.; Uschold, M. (1999). A Framework for Understanding and Classifying Ontology Applications. *Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5) in the different communities to overcome terminological*, 2 Ago. Stockholm, Sweden. Amsterdam, The Netherlands: CEUR Publications and University of Amsterdam, pp. 1-12. ISBN: 90-5470-085-8.

JasperForge (2009). JasperForge: JasperReports Información del Proyecto. Disponible en: <http://jasperforge.org/projects/jasperreports>. [Consultado en junio de 2010].

JasperForge (2009). JasperForge: JasperServer Información del Proyecto. Disponible en: <http://jasperforge.org/projects/jasperserver>. [Consultado en junio de 2010].

Java.net (2010). Glassfish: Open Source Application Server. Disponible en: <https://glassfish.dev.java.net/>. [Consultado en junio de 2010].

Kahan, J.; Koivunen, M.; Prud'Hommeaux, E.; Swick, R. (2001). Annotea: An Open RDF Infrastructure for Shared Web Annotations. *Proceedings of the 10th international conference on World Wide Web*. New York, USA: ACM Press, pp. 623-632. ISBN: 1-58113-348-0. Disponible en: http://portal.acm.org/ft_gateway.cfm?id=372166&type=pdf&coll=GUIDE&dl=GUIDE&CFID=92736257&CFTOKEN=71919243. [Consultado en junio de 2010].

Kalfoglou, Y.; Schorlemmer, M. (2003). Ontology mapping: the state of the art. *The Knowledge Engineering Review*, **18**(1), pp. 1-31. ISSN: 0269-8889.

Kalyanpur, A.; Hendler, J.; Parsia, B.; Goldbeck, J. (2005). SMORE - Semantic Markup, Ontology, and RDF Editor. *Maryland Information and Network Dynamics Lab Semantic Web Agents Projects*. Disponible en: <http://www.mindswap.org/papers/SMORE.pdf>. [Consultado en junio de 2010].

Kent, A.; Berry, M.M.; Leuhrs, F.U.; Perry, J.W. (1955). Machine literature searching VIII. Operational criteria for designing information retrieval systems. *American Documentation*, **6**(2), pp. 93-101. ISSN: 0096-946X.

Kitchenham, B. (1996). DESMET: A method for evaluating Software Engineering methods and tools. Disponible en: <http://www.osel.co.uk/desmet.pdf>. [Consultado en junio de 2010].

Knowledge System Laboratory - Stanford University (2001). Stanford ksl network services. Disponible en: <http://www-ksl-svc.stanford.edu/>. [Consultado en junio de 2010].

Kogut, P.; Holmes, W. (2001). AeroDAML: Applying Information Extraction to Generate DAML Annotations from Web Pages. *First International Conference on Knowledge Capture (K-CAP 2001). Workshop on Knowledge Markup and Semantic Annotation, 21 Oct., Victoria, BC, Canada*. CEUR Workshop Proceedings, pp. 3. ISSN: 1613-0073.

Laboratory for Applied Ontology (2006). DOLCE. *Laboratory for Applied Ontology. Institute of Cognitive Science and Technology*. Disponible en: <http://www.loa-cnr.it/DOLCE.html>. [Consultado en junio de 2010].

Laboratory for Applied Ontology (2003). Dolce: OntoWordNet Project. Disponible en: <http://www.loa-cnr.it/DOLCE.html#OntoWordNet>. [Consultado en junio de 2010].

Lamarca, M.J. (2002). DAML-OIL. *Hipertexto: El nuevo concepto de documento en la cultura de la imagen*. Disponible en: <http://www.hipertexto.info/documentos/daml.htm>. [Consultado en junio de 2010].

Lassila, O.; McGuinness, D. (2001). The role of frame-based representation on the semantic web. *Linköping Electronic Articles in Computer and Information Science*, **6**(5), p. 2001. ISSN: 1401-9841. Disponible en: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.4297&rep=rep1&type=pdf>. [Consultado en junio de 2010].

Library of Congress (2005). Metadata Authority Description Schema (MADS). Disponible en: <http://www.loc.gov/standards/mads/>. [Consultado en junio de 2010].

Library of Congress (2005). Metadata Object Description Schema (MODS). Disponible en: <http://www.loc.gov/standards/mods/>. [Consultado en junio de 2010].

Linhalis, F.; Mattos Fortes, R.P.; Abreu Moreira, D. (2009). OntoMap: an ontology-based architecture to perform the semantic mapping between an interlingua and software components. *Knowledge and Information Systems*, **22**(3), pp. 319-345. ISSN: 0219-1377. Disponible en: <http://www.springerlink.com/content/f15418251g15w834/fulltext.pdf>. [Consultado en junio de 2010].

Madhavan, J.; Bernstein, P.A.; Rahm, E.; Apers, P.M.; Atzeni, P.; Ceri, S.; Paraboschi, S.; Ramamohanarao, K.; Snodgrass, R.T. (2001). Generic Schema Matching with Cupid. *Proceedings of the 27th International Conference on Very Large Data Bases, 11-14 Sep. Roma, Italy*. Los Altos, CA, USA: Morgan Kaufmann Publishers Inc., pp. 49-58. ISBN: 1-55860-804-4.

Marinescu, F. (2002). *EJB Design Patterns: Advanced Patterns, Processes and Idioms*. New York, USA: John Wiley & Sons Inc, pp. 288. ISBN: 0-471-20831-0.

MarkLogic (2010). MarkLogic - XML Server. Disponible en: <http://www.marklogic.com/information/xml-server.html>. [Consultado en junio de 2010].

Marrero, M.; Morato, J.; Lloréns, J.; Moreiro, J. (2006). Definición de un marco para el desarrollo de metodologías para la fusión de ontologías. *Conferencia IADIS Ibero-Americana WWW/Internet, 6-7 Oct. Murcia, Spain*. IADIS Press, pp. 115-122. ISBN: 972-8924-20-8. Disponible en: http://www.iadis.net/dl/final_uploads/200607L015.pdf. [Consultado en junio de 2010].

Martin, P.; Eklund, P.W. (2000). Knowledge Retrieval and the World Wide Web. *IEEE Intelligent Systems*, **15**(3), pp. 18-25. ISSN: 1541-1672.

Martin, P.; Eklund, P.W. (2002). Manageable Approaches to the Semantic Web. *The Eleventh International World Wide Web Conference, 7-11 May. Honolulu, Hawaii, USA*. New York, USA: ACM Press, p. 33. ISBN: 1-880672-20-0. Disponible en: <http://www.webkb.org/doc/papers/www11/www11.html>. [Consultado en junio de 2010].

Massachusetts Institute of Technology Libraries (2010). Metadata Mappings (Crosswalks): Metadata Reference Guide: MIT Libraries. Disponible en: <http://libraries.mit.edu/guides/subjects/metadata/mappings.html>. [Consultado en junio de 2010].

Massachusetts Institute of Technology Libraries (2008). SIMILE Project. Disponible en: <http://simile.mit.edu/>. [Consultado en junio de 2010].

Massachusetts Institute of Technology Libraries (2008). Semantic Interoperability of Metadata and Information in unLike Environments. *Massachusetts Institute of Technology Libraries*. Disponible en: <http://simile.mit.edu/>. [Consultado en junio de 2010].

Massachusetts Institute of Technology (2008). Piggy Bank - SIMILE. Disponible en: http://simile.mit.edu/wiki/Piggy_Bank. [Consultado en junio de 2010].

Melnik, S.; Garcia-Molina, H.; Rahm, E. (2002). Similarity flooding: a versatile graph matching algorithm and its application to schema matching. *Proceedings 18th International Conference on Data Engineering, 26 Feb. - 1 Mar., San Jose, CA, USA*. IEEE Computer Society, pp. 117-128. ISBN: 0-7695-1531-2.

Melnik, S.; Rahm, E.; Bernstein, P.A. (2003). Rondo: a programming platform for generic model management. *Proceedings of the 22nd International Conference on Management of Data (SIGMOD), 9-12 Jun. San Diego, CA, USA*. New York, USA: ACM Press, pp. 193-204. ISBN: 1-58113-634-X. Disponible en: http://portal.acm.org/ft_gateway.cfm?id=872782&type=pdf&coll=GUIDE&dl=GUIDE&CFID=92742058&CFTOKEN=73663069. [Consultado en junio de 2010].

Meyer, H.; Bruder, I.; Heuer, A.; Weber, G. (2002). The Xircus Search Engine. *Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX), 9-11 Dic. Schloss Dagstuhl, Germany*. Computer Science Bibliography. Universität Trier, pp. 119-124. Disponible en: <http://www.xircus.de/publications/publications/inex-2002.pdf>. [Consultado en junio de 2010].

Microsoft (2010). Bing. Disponible en: <http://www.bing.com/>. [Consultado en junio de 2010].

Microsoft (2010). Powerset. Disponible en: <http://powerset.com/>. [Consultado en junio de 2010].

Mihalcea, R.; Mihalcea, S.I. (2001). Word Semantics for Information Retrieval: Moving One Step Closer to the Semantic Web. *Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence, 7-9 Nov. Dallas, TX, USA*. Washington, DC, USA: IEEE Computer Society, pp. 280-287. ISBN: 0-7695-1417-0.

Mitra, P.; Wiederhold, G.; Kersten, M.L. (2000). A Graph-Oriented Model for Articulation of Ontology Interdependencies. *Advances in Database Technology. EDBT 2000*. Berlin, Germany: Springer-Verlag, pp. 86-100. ISBN: 978-3-540-67227-2. Disponible en: <http://www.springerlink.com/content/mdebqfhphhtk5111/fulltext.pdf>. [Consultado en junio de 2010].

Mizoguchi, R.; Vanwelkenhuysen, J.; Ikeda, M. (1995). Task ontology for reuse of problem solving knowledge. *Towards very large knowledge bases: knowledge building and knowledge sharing (KBKS'95)*. Amsterdam, The Netherlands: IOS Press, pp. 46-57. ISBN: 9-051-99217-3.

Morato, J.; Fraga, A.; Adreadakis, Y.; Sanchez, S. (2008). Socialization of the Semantic Web: the integration with Web 2.0. *The Open Knowledge Society. A Computer Science and Information Systems Manifesto*. Berlin, Germany: Springer-Verlag, pp. 406-415. ISBN: 978-3-540-87782-0. Disponible en:

<http://www.springerlink.com/content/p7486786r75g6w85/fulltext.pdf>. [Consultado en junio de 2010].

Morato, J.; Moreiro, J.A.; Llorens, J.; Marzal, M.A. (2004). WordNet Applications. *Proceedings of the 2nd International Global Wordnet Conference, 20-23 Ene. Brno, Czech Republic*. Brno, Czech Republic: Masaryk University, pp. 270-278. ISBN: 80-210-3302-9. Disponible en: <http://www.fi.muni.cz/gwc2004/proc/105.pdf>. [Consultado en junio de 2010].

Morato, J.; Sánchez, S.; Valiente, M.C. (2005). Análisis de estrategias de posicionamiento en relación con la relevancia documental. *El profesional de la Información*, **14**(1), pp. 21-29. ISSN: 1386-6710. Disponible en: [http://elprofesionaldelainformacion.metapress.com/\(2kfnesjyquvnd3ir2npgecvzs\)/app/home/contribution.asp?referrer=parent&backto=issue,3,16;journal,6,44;linkingpublicationresults,1:105302,1](http://elprofesionaldelainformacion.metapress.com/(2kfnesjyquvnd3ir2npgecvzs)/app/home/contribution.asp?referrer=parent&backto=issue,3,16;journal,6,44;linkingpublicationresults,1:105302,1). [Consultado en junio de 2010].

Mueller, E.T. (2003). ThoughtTreasure commonsense knowledge base. Disponible en: <http://xenia.media.mit.edu/~mueller/papers/tt.html>. [Consultado en junio de 2010].

Musen, M.A.; Noy, N.F. (2001). Anchor-PROMPT: Using Non-Local Context for Semantic Matching. *Proceedings of the Workshop on Ontologies and Information sharing at the International Joint Conference on Artificial Intelligence (IJCAI-01), 4-5 Ago. Seattle, USA*. Artificial Intelligence Journal Division of IJCAI, pp. 63-70. ISBN: 1-55860-777-3. Disponible en: <http://dit.unitn.it/~accord/RelatedWork/Matching/noy.pdf>. [Consultado en junio de 2010].

National Biological Information Infrastructure (2010). NBII Clearinghouse Search. Disponible en: <http://mercury.ornl.gov/nbii/>. [Consultado en junio de 2010].

Noy, N. (2004). Ontology Mapping. *International Handbook on Information Systems, Handbook on Ontologies*. Berlin, Germany: Springer-Verlag, pp. 551-573. ISBN: 978-3-540-70999-2. Disponible en: <http://www.uni-koblenz.de/~staab/Research/Publications/handbook2ndEdition/HandbookOnOntologies2nd.html>. [Consultado en junio de 2010].

Noy, N. (2009). Protegewiki: Prompt. Disponible en: <http://protege.cim3.net/cgi-bin/wiki.pl?Prompt>. [Consultado en junio de 2010].

Noy, N.; McGuinness, D.L. (2001). Ontology Development 101: A Guide to Creating Your First Ontology. , pp. 25. Disponible en: <http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html>. [Consultado en junio de 2010].

O'Reilly, T. (2005). What is Web 2.0 – design patterns and business models for the next generation of software. Disponible en: <http://oreilly.com/web2/archive/what-is-web-20.html>. [Consultado en junio de 2010].

- Object Management Group (2009). ODM 1.0. Disponible en: <http://www.omg.org/spec/ODM/1.0/>. [Consultado en junio de 2010].
- Object Management Group (2010). Object Management Group. Disponible en: <http://www.omg.org/>. [Consultado en junio de 2010].
- Object Management Group (2006). Unified Modeling Language (UML), version 2.1.1. Disponible en: <http://www.omg.org/technology/documents/formal/uml.htm>. [Consultado en junio de 2010].
- OBO Foundry (2010). The Open Biological and Biomedical Ontologies. Disponible en: <http://www.obofoundry.org/>. [Consultado en junio de 2010].
- Ogbuji, U. (2001). Thinking XML: Basic XML and RDF techniques for knowledge management. *IBM Developer Works*. Disponible en: <http://www-128.ibm.com/developerworks/xml/library/x-think6.html>. [Consultado en junio de 2010].
- Online Computer Library Center (2010). OCLC - Dewey Decimal Classification. Disponible en: <http://www.oclc.org/dewey/>. [Consultado en junio de 2010].
- Online Computer Library Center (2010). Persistent URL Home Page. Disponible en: <http://purl.oclc.org/>. [Consultado en junio de 2010].
- Ontopia (2003). Ontopia PSIs. Disponible en: <http://psi.ontopia.net/>. [Consultado en junio de 2010].
- Ontotext (2010). KIM Platform. Disponible en: <http://www.ontotext.com/kim/>. [Consultado en junio de 2010].
- Ontotext (2010). OWLIM - OWL Semantic Repository. Disponible en: <http://www.ontotext.com/owlim/>. [Consultado en junio de 2010].
- Open Archives (2008). Open Archives Initiatives. Disponible en: <http://www.openarchives.org/>. [Consultado en junio de 2010].
- OpenRDF.org (2010). OpenRDF.org: Home of Sesame. Disponible en: <http://www.openrdf.org/>. [Consultado en junio de 2010].
- Oracle (2010). Welcome to NetBeans. *Oracle Corporation*. Disponible en: <http://www.netbeans.org/>. [Consultado en junio de 2010].
- Oren, E.; Delbru, R.; Catasta, M.; Cyganiak, R.; Stenzhorn, H.; Tummarello, G. (2008). Sindice.com: a document-oriented lookup index for open linked data. *International Journal of Metadata, Semantics and Ontologies*, **3**(1), pp. 37-52.
- OwlSeek (2009). OwlSeek.com: Owl Ontology. Disponible en: <http://www.owlseek.com/>. [Consultado en junio de 2010].
- Özel, S.A.; Altingövde, I.S.; Ulusoy, Ö.; Özsoyoglu, G.; Meral Özsoyoglu, Z. (2004). Metadata-based modeling of information resources on the Web. *Journal of the American Society for Information Science and Technology*, **55**(2), pp. 97-110. ISSN: 1532-2882.

Palacios, V.; Morato, J.; Fraga, A.; Llorens, J. (2006). A methodology for semantic qualification of schemas. *Proceedings of the ISCW-2006, Fifth International Semantic Web Conference, 5-9 Nov. Athens, GA, USA*. Berlin, Germany: Springer Verlag, pp. 4. ISBN: 978-3-540-49029-6.

Palacios, V.; Morato, J.; Llorens, J.; Moreira, J.A. (2006). DCMI Abstract Model analysis. Resource Model. *International Conference on Dublin Core and Metadata Applications, 3-6 Oct. Manzanillo, Méjico*. DCMI Conference Papers, pp. 10. Disponible en: <http://dcpapers.dublincore.org/ojs/pubs/article/view/854/850>. [Consultado en junio de 2010].

Palacios, V.; Morato, J.; Llorens, J.; Moreira, J.A. (2006). Indicadores Web sobre utilización de ontologías. *Conferência Ibérica de Sistemas e Tecnologias de Informação (CISTI-06), 21-23 Jun. Ofir, Portugal*. Actas da 1ª Conferência Ibérica de Sistemas e Tecnologias de Informação. Vol II, pp. 199-214. ISBN: 978-989-20-0271-2.

Palacios, V.; Morato, J.; Llorens, J.; Sanchez, S.; Moreira, J.A. (2006). Metodología para cualificación semántica de esquemas: aplicación a dublin core. *Conferencia IADIS Iberoamericana WWW/Internet 2006, 6-7 Oct. Murcia, Spain*. IADIS Press, pp. 8. ISBN: 972-8924-20-8.

Palacios, V.; Morato, J.; Sanchez, S.; Llorens, J.; Moreira, J.A. (2006). An improved methodology for semantic scheme qualification. *The 1st International Workshop: Semantic Information Integration on Knowledge Discovery (SIK 2006), 4-6 Dic. Yogyakarta, Indonesia*. Disponible en: http://74.125.155.132/scholar?q=cache:QNumoxZaOUJ:scholar.google.com/+An+improved+methodology+for+semantic+scheme+qualification&hl=es&as_sdt=2000&as_vis=1. [Consultado en junio de 2010].

Parada, R. (2008). DOAC Vocabulary Specification. Disponible en: <http://ramonantonio.net/doac/0.1/>. [Consultado en junio de 2010].

Peig Olivé, E. (2004). Interoperabilidad de metadatos en sistemas distribuidos. , p. 92. Disponible en: http://www.tdr.cesca.es/TESIS_UPF/AVAILABLE/TDX-0316104-132946/tepo1de1.pdf. [Consultado en junio de 2010].

Plant Ontology Consortium (2010). Plant Ontology Consortium web site. Disponible en: <http://www.plantontology.org/>. [Consultado en junio de 2010].

PostgreSQL (2010). PostgreSQL: The world's most advanced open source database. Disponible en: <http://www.postgresql.org/>. [Consultado en junio de 2010].

Powel, A.; Nilsson, M.; Naeve, A.; Johnston, P.; Baker, T. (2007). DCMI Abstract Model. Disponible en: <http://dublincore.org/documents/abstract-model/>. [Consultado en junio de 2010].

Princeton University (2010). WordNet - a lexical database for the English language. Disponible en: <http://wordnet.princeton.edu/>. [Consultado en junio de 2010].

Protein Information Resource (2009). The Protein Ontology. Disponible en: <http://pir.georgetown.edu/pro/pro.shtml>. [Consultado en junio de 2010].

PSI Metadata (2006). Published Subject Indicators for Modelling Thesaurii. Disponible en: <http://www.techquila.com/psi/thesaurus/>. [Consultado en junio de 2010].

Radhakrishnan, A. (2009). 9 Semantic Search Engines That Will Change the World of Search. *Search Engine Journal*. Disponible en: <http://www.searchenginejournal.com/semantic-search-engines/9832/#ixzz0nugn9jls>. [Consultado en junio de 2010].

Rahm, E.; Bernstein, P. (2001). A survey of approaches to automatic schema matching. *VLDB Journal: Very Large Data Bases*, **10**(4), pp. 334-350. ISSN: 1066-8888. Disponible en: http://portal.acm.org/ft_gateway.cfm?id=767154&type=pdf&coll=GUIDE&dl=GUIDE&CFID=92758544&CFTOKEN=57654903. [Consultado en junio de 2010].

Ruixin, Y.; Xinhua, D.; Menas, K.; Changzhou, W.; Xiaoyang Sean, W. (2001). An XML-Based Distributed Metadata Server (DIMES) Supporting Earth Science Metadata. *Proceedings of the 13th International Conference on Scientific and Statistical Database Management, 18-20 Jul. Fairfax, VA, USA*. Washington, DC, USA: IEEE Computer Society, pp. 251-256. ISSN: 1099-3371.

Rumbaugh, J.; Firesmith, D. (1996). *OMT Insights: Perspective on Modeling from the Journal of Object-Oriented Programming*. New York, USA: SIGS Books, pp. 412. ISBN: 1-884842-58-5.

Sánchez, L.; Fernández, N. (2005). La Web Semántica: fundamentos y breve estado del arte. *Novatica*, **6**(178), pp. 6-12. Disponible en: <http://www.ati.es/novatica/2005/178/178-6.pdf>. [Consultado en junio de 2010].

SCHEMAS (2003). Forum for Metadata Schema Designer and Implementers. *UKOLN: UK Office for Library and Information Networking*. Disponible en: <http://www.schemas-forum.org/>. [Consultado en junio de 2010].

SchemaWeb (2008). SchemaWeb: RDF Schemas Directory. Disponible en: <http://www.schemaweb.info/>. [Consultado en junio de 2010].

Schlieder, T.; Meuss, H. (2002). Querying and ranking XML documents. *Journal of the American Society for Information Science and Technology*, **53**(6), pp. 489-503. ISSN: 1532-2882.

Schwartz, C. (2009). Metadata crosswalks. *Cataloging Futures*. Disponible en: <http://www.catalogingfutures.com/catalogingfutures/metadata-crosswalks/>. [Consultado en junio de 2010].

SearchTools.com (2009). XML Query and Searching Resources. Disponible en: <http://www.searchtools.com/info/xml-resources.html>. [Consultado en junio de 2010].

SEKT Project (2006). Welcome to SEKT — SEKT Portal. Disponible en: <http://www.sekt-project.com/>. [Consultado en junio de 2010].

SemWebCentral (2007). Snoggle -- A graphical, swrl-based ontology mapper. Disponible en: <http://snoggle.projects.semwebcentral.org/>. [Consultado en junio de 2010].

SemWebCentral (2010). SemWebCentral: Open Source Tools for the Semantic. Disponible en: <http://www.semwebcentral.org>. [Consultado en junio de 2010].

Semantic Metadata Search (2010). Ontología SEMSE. Disponible en: <http://purl.org/semse/>. [Consultado en junio de 2010].

Semantic Metadata Search (2010). SEMSE: SEMantic Metadata SEArch. Disponible en: <http://www.kr.inf.uc3m.es/SEMSE.htm>. [Consultado en junio de 2010].

Semantic Metadata Search (2010). SVN para desarrollo colaborativo. Disponible en: <http://163.117.152.157/repos>. [Consultado en junio de 2010].

Semantic Metadata Search (2010). Servidor de Aplicaciones SEMSE. Disponible en: <http://163.117.152.157:8080/Semse/Inicio.jsp>. [Consultado en junio de 2010].

Semantic Web Abreemnt Group (2001). Metavocab (MVCB). Disponible en: <http://webns.net/mvcb/>. [Consultado en junio de 2010].

SemanticWeb.org (2002). Library: WordNet. Disponible en: <http://www.semanticweb.org/library/>. [Consultado en junio de 2010].

SHOE (2000). The SHOE Knowledge Annotator. Disponible en: <http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html>. [Consultado en junio de 2010].

Shvaiko, P.; Euzenat, J. (2005). Survey of schema-based matching approaches. *Journal on Data Semantics IV*. Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, pp. 146-171. ISBN: 978-3-540-31001-3. Disponible en: <http://www.springerlink.com/content/yj2072765w71548h/fulltext.pdf>. [Consultado en junio de 2010].

Smart, P.R.; Engelbrecht, P.C. (2008). An Analysis of the Origin of Ontology Mismatches on the Semantic Web. *Proceedings of the 16th international conference on Knowledge Engineering: Practice and Patterns, 29 Sep. - 2 Oct. Acitrezza, Italy*. Berlin: Springer Verlag, pp. 120-135. ISBN: 978-3-540-87695-3.

Soto, J.; García, E.; Sanchez, S. (2006). Problemas de almacenamiento e inferencia sobre grandes conjuntos de metadatos en un repositorio semántico de objetos de aprendizaje. *Actas del III Simposio Pluridisciplinar sobre objetos y diseños de aprendizaje, 25-27 Sep. Oviedo, España*. Disponible en: <http://www.imai-software.com/openlab/data/SOTOGARCIASANCHEZ.pdf>. [Consultado en junio de 2010].

SpringSource (2010). SpringSource.org. Disponible en: <http://www.springsource.org/>. [Consultado en junio de 2010].

St. Pierre, M.; LaPlant Jr., W.P. (1998). *Issues in Crosswalking Content Metadata Standards*. National Information Standards Organization. Disponible en: <http://www.niso.org/press/whitepapers/crswalk.html>. [Consultado en junio de 2010].

Stanford KSL Network Services (2001). Chimaera. Disponible en: <http://www-ksl-svc.stanford.edu:5915/&service=CHIMAERA>. [Consultado en junio de 2010].

Stanford KSL Network Services (2001). Ontolingua Ontology Editor. Disponible en: <http://www-ksl-svc.stanford.edu:5915/&service=FRAME-EDITOR>. [Consultado en junio de 2010].

Stanford University (2010). Stanford University main page. Disponible en: <http://www.stanford.edu/>. [Consultado en junio de 2010].

Stanford University (2010). The Protégé Ontology Editor and Knowledge Acquisition System. Disponible en: <http://protege.stanford.edu/>. [Consultado en junio de 2010].

Steve, G.; Gangemi, A.; Pisanelli, D.M. (1997). Integrating Medical Terminologies with ONIONS Methodology. *Information Modeling and Knowledge Bases VIII*. Amsterdam: IOS Press, Amsterdam, pp. 380. ISBN: 978-90-5199-334-9. Disponible en: <http://citeseer.ist.psu.edu/cache/papers/cs/20800/http:zSzzSzzaussure.irmkant.rm.cnr.itzSzontozSzpublzSzonions97zSzonions97.pdf/steve97integrating.pdf>. [Consultado en junio de 2010].

Suggested Upper Merged Ontology (2010). Ontology Portal. Disponible en: <http://www.ontologyportal.org/>. [Consultado en junio de 2010].

Sullivan, J. (2006). Search Engines and Algorithms: Semantic Search. *Search Optimization*, pp. 5. Disponible en: <http://www.seoachat.com/c/a/Search-Engine-Optimization-Help/Search-Engines-and-Algorithms-Semantic-Search/>. [Consultado en junio de 2010].

Sun Microsystems (2010). Developer Resources for Java Technology. Disponible en: <http://java.sun.com/>. [Consultado en junio de 2010].

Sure, Y.; Studer, R. (2005). Semantic Web Technologies for Digital Libraries. *Library Management*, 4-5(26), pp. 190-195. ISSN: 0143-5124. Disponible en: http://www.uni-koblenz.de/~sure/publications/2005_sw_for_dl.pdf. [Consultado en junio de 2010].

Swoogle (2007). Swoogle Semantic Web Search Engine. Disponible en: <http://swoogle.umbc.edu/>. [Consultado en junio de 2010].

Swoogle (2007). Swoogle Statistic's of the Semantic Web. Disponible en: http://swoogle.umbc.edu/index.php?option=com_swoogle_stats. [Consultado en junio de 2010].

Tang, J.; Li, J.; Liang, B.; Huang, X.; Li, Y.; Wang, K. (2006). Using Bayesian Decision for Ontology Mapping. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(4), pp. 243-262. ISSN: 1570-8268.

- Taulé, M.; Martí, M.A. (2003). SENSEVAL, una aproximació computacional al significat. *Digithum: Revista digital d'humanitats*, (5), pp. 13. Disponible en: <http://www.uoc.edu/humfil/articles/cat/taule0303/taule0303.html>. [Consultado en junio de 2010].
- Taylor, M. (2006). Zthes: a Z39.50 profile for thesaurus navigation, version 1.0. Disponible en: <http://zthes.z3950.org/z3950/zthes-z3950-1.0.html>. [Consultado en junio de 2010].
- Techquila PSI (2003). Techquila's Published Subject Identifiers. Disponible en: <http://www.techquila.com/psi/>. [Consultado en junio de 2010].
- Terziev, I.; Kiryakov, A.; Manov, D. (2005). *Base Upper-level Ontology (BULO) Guidance Deliverable 1.8.1, SEKT project*. Disponible en: http://proton.semanticweb.org/D1_8_1.pdf. [Consultado en junio de 2010].
- The Board of Trustees of Leland Stanford Junior University (2010). NCBO BioPortal: Welcome to the NCBO BioPortal. Disponible en: <http://bioportal.bioontology.org/>. [Consultado en junio de 2010].
- The Eclipse Foundation (2010). Eclipse Modeling - EMF - Home. Disponible en: <http://www.eclipse.org/modeling/emf/>. [Consultado en junio de 2010].
- The Gene Ontology (1999). The Gene Ontology. Disponible en: <http://www.geneontology.org/>. [Consultado en junio de 2010].
- The Text Encoding Initiative (2010). The Text Encoding Initiative. Disponible en: <http://www.tei-c.org/>. [Consultado en junio de 2010].
- U. S. Environmental Protection Agency (2010). Environmental Data Registry. Disponible en: <http://www.epa.gov/edr/>. [Consultado en junio de 2010].
- UDCC (2009). Universal Decimal Classification Consortium. Disponible en: <http://www.udcc.org/>. [Consultado en junio de 2010].
- UKOLN (2002). Mapping between metadata formats. Disponible en: <http://www.ukoln.ac.uk/metadata/interoperability/>. [Consultado en junio de 2010].
- UKOLN (2003). UKOLN Metadata Resources. Disponible en: <http://www.ukoln.ac.uk/metadata/resources/>. [Consultado en junio de 2010].
- UMBEL.org (2009). UMBEL Index. Disponible en: <http://umbel.org/>. [Consultado en junio de 2010].
- Unicode (2010). Página principal. Disponible en: <http://www.unicode.org>. [Consultado en junio de 2010].
- Universidad Politécnica de Cataluña (2008). Web MCR Interface MEANING. *Departament De Sistemes I Llenguatges Informàtics*. Disponible en: <http://garraf.epsevg.upc.es/cgi-bin/wei4/public/wei.consult.perl>. [Consultado en junio de 2010].

Université de Neuchatel (2003). Information Management Institute: knOWLer. Disponible en: http://www2.unine.ch/imi/page11291_en.html. [Consultado en junio de 2010].

University of Southern California (2007). Ontosaurus: loom web browser. Disponible en: <http://www.isi.edu/isd/ontosaurus.html>. [Consultado en junio de 2010].

Uren, V.; Cimiano, P.; Iria, J.; Handschuh, S.; Vargasvera, M.; Motta, E.; Ciravegna, F. (2006). Semantic annotation for knowledge management: Requirements and a survey of the state of the art. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(1), pp. 14-28. ISSN: 1570-8268.

Van Hage, W.R.; Katrenko, S.; Schreiber, G. (2005). A Method to Combine Linguistic Ontology-Mapping Techniques. *Proceedings of the 4th International Semantic Web Conference (ISWC-2005), 6-10 Nov, Galway, Ireland*. Berlin, Germany: Springer-Verlag, pp. 732-744. ISBN: 3-540-29754-5. Disponible en: <http://www.springerlink.com/content/u6203272m174k20n/fulltext.pdf>. [Consultado en junio de 2010].

Visual Resources Association (2010). Visual Resources Association. Disponible en: <http://www.vraweb.org/>. [Consultado en junio de 2010].

Wache, H. Vögele, T. Visser, U. Stuckenschmidt, H. Schuster, G. Neumann, H. Hübner, S. (2001). *Proceedings of the Workshop on Ontologies and Information sharing at the International Joint Conference on Artificial Intelligence (IJCAI-01), 4-5 Ago. Seattle, USA*. Artificial Intelligence Journal Division of IJCAI, pp. 108-117. ISBN: 1-55860-777-3. Disponible en: <http://www.citeulike.org/user/pchampin/article/593500>. [Consultado en junio de 2010].

Washington University (2010). Foundational Model of Anatomy ontology. Disponible en: <http://sig.biostr.washington.edu/projects/fm/index.html>. [Consultado en junio de 2010].

WebKB (2003). Integration of WordNet 1.7 in WebKB-2. Disponible en: <http://www.webkb.org/doc/wn/>. [Consultado en junio de 2010].

WebKB (2003). WebKB home page. Disponible en: <http://www.webkb.org/>. [Consultado en junio de 2010].

WebODE (2002). Welcome to WebODE 2.0. Disponible en: <http://webode.dia.fi.upm.es/>. [Consultado en junio de 2010].

WebOnto (1998). WebOnto Page. Disponible en: <http://projects.kmi.open.ac.uk/webonto/>. [Consultado en junio de 2010].

Weibel, S.L.; Baker, T. (2004). DCMI Policy on Naming Terms. Disponible en: <http://dublincore.org/documents/naming-policy/>. [Consultado en junio de 2010].

Wolfram Alpha LLC (2010). Computational Knowledge Engine. Disponible en: <http://www.wolframalpha.com/>. [Consultado en junio de 2010].

World Wide Web Consortium (1998). Document Object Model (DOM) Level 1 Specification. Disponible en: <http://www.w3.org/TR/REC-DOM-Level-1/>. [Consultado en junio de 2010].

World Wide Web Consortium (1999). Resource Description Framework (RDF) Model and Syntax specification. Disponible en: <http://www.w3.org/RDF/>. [Consultado en junio de 2010].

World Wide Web Consortium (1999). XML Path Language (XPath). Disponible en: <http://www.w3.org/TR/xpath>. [Consultado en junio de 2010].

World Wide Web Consortium (1999). XSL Transformations (XSLT) Version 1.0. Disponible en: <http://www.w3.org/TR/xslt>. [Consultado en junio de 2010].

World Wide Web Consortium (2001). XML Linking Language (XLink) Version 1.0. Disponible en: <http://www.w3.org/TR/xlink/>. [Consultado en junio de 2010].

World Wide Web Consortium (2002). XHTML 1.0: The Extensible HyperText Markup Language (Second Edition). Disponible en: <http://www.w3.org/TR/xhtml1>. [Consultado en junio de 2010].

World Wide Web Consortium (2003). XPointer Framework. Disponible en: <http://www.w3.org/TR/xptr-framework/>. [Consultado en junio de 2010].

World Wide Web Consortium (2004). RDF Primer. Disponible en: <http://www.w3.org/TR/rdf-primer/>. [Consultado en junio de 2010].

World Wide Web Consortium (2004). RDF Vocabulary Description Language 1.0: RDF Schema. World Wide Web Consortium. Disponible en: <http://www.w3.org/TR/PR-rdf-schema/>. [Consultado en junio de 2010].

World Wide Web Consortium (2004). XML Information Set (Second Edition). Disponible en: <http://www.w3.org/TR/xml-infoset/>. [Consultado en junio de 2010].

World Wide Web Consortium (2004). XML Schema (second edition). Disponible en: <http://www.w3.org/XML/Schema>. [Consultado en junio de 2010].

World Wide Web Consortium (2005). SKOS Core Vocabulary Specification. Disponible en: <http://www.w3.org/2004/02/skos/>. [Consultado en junio de 2010].

World Wide Web Consortium (2006). Extensible Stylesheet Language (XSL) Version 1.1. Disponible en: <http://www.w3.org/TR/xsl11/>. [Consultado en junio de 2010].

World Wide Web Consortium (2006). Ontaria - Easy Access to the Semantic Web. Disponible en: <http://www.w3.org/2004/ontaria/>. [Consultado en junio de 2010].

World Wide Web Consortium (2006). RDF/OWL Representation of WordNet. Disponible en: <http://www.w3.org/TR/wordnet-rdf/>. [Consultado en junio de 2010].

World Wide Web Consortium (2007). Tabulator: Generic data browser. Disponible en: <http://www.w3.org/2005/ajar/tab>. [Consultado en junio de 2010].

World Wide Web Consortium (2008). Extensible Markup Language (XML) 1.0 (fifth edition). Disponible en: <http://www.w3.org/TR/REC-xml/>. [Consultado en junio de 2010].

World Wide Web Consortium (2009). Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification. Disponible en: <http://www.w3.org/TR/REC-CSS2/>. [Consultado en junio de 2010].

World Wide Web Consortium (2009). Namespaces in XML 1.0 (third edition). Disponible en: <http://www.w3.org/TR/REC-xml-names/>. [Consultado en junio de 2010].

World Wide Web Consortium (2009). OWL 2 Web Ontology Language Document Overview. Disponible en: <http://www.w3.org/TR/2009/REC-owl2-overview-20091027/>. [Consultado en junio de 2010].

World Wide Web Consortium (2009). SKOS Simple Knowledge Organization System Reference. Disponible en: <http://www.w3.org/TR/skos-reference/>. [Consultado en junio de 2010]. [Consultado en junio de 2010].

World Wide Web Consortium (2010). Página principal. Disponible en: <http://w3c.org>. [Consultado en junio de 2010].

World Wide Web Consortium (2010). Representing vCard Objects in RDF. Disponible en: <http://www.w3.org/Submission/vcard-rdf/>. [Consultado en junio de 2010].

XML Pitstop (2010). Largest Source of XML Examples on the Web. Disponible en: <http://www.xmlpitstop.com/>. [Consultado en junio de 2010].

XML.ORG (2008). XML.ORG: Applying XML and Web Services Standards in Industry. Disponible en: <http://www.xml.org/>. [Consultado en junio de 2010].

XMLmind (2010). Fast XML searching | Qizx, native XQuery database engine. Disponible en: <http://www.xmlmind.com/qizx/>. [Consultado en junio de 2010].

Xiaomeng, S.; Hakkarainen, S.; Brasethvik, T. (2004). Semantic enrichment for improving systems interoperability. *Proceedings of the 2004 ACM symposium on Applied computing, 14-7 Mar., Nicosia, Cyprus*. New York, USA: ACM Press, pp. 1634-1641. ISBN: 1-58113-812-1. Disponible en: http://portal.acm.org/ft_gateway.cfm?id=968227&type=pdf&coll=GUIDE&dl=GUIDE&CFID=92766536&CFTOKEN=40490680. [Consultado en junio de 2010].

Yau, H.S.; Hawker, J.S. (2004). SA_MetaMatch: relevant document discovery through document metadata and indexing. *Proceedings of the 42nd annual Southeast regional conference, 2-3 Abr. Huntsville, AL, USA*. New York, USA: ACM Press, pp. 385-390. ISBN: 1-58113-870-9. Disponible en: http://portal.acm.org/ft_gateway.cfm?id=986630&type=pdf&coll=GUIDE&dl=GUIDE&CFID=92766921&CFTOKEN=65702489. [Consultado en junio de 2010].

Zeng, M.L.; Chan, L.M. (2004). Trends and issues in establishing interoperability among knowledge organization systems. *Journal of the American Society for Information Science and Technology*, **55**(5), pp. 377-395. ISSN: 1532-2882.

ZVON.org (2010). ZVON. Disponible en: <http://www.zvon.org/>. [Consultado en junio de 2010].

Capítulo VIII: Glosario

- **Alineamiento (*Alignment*):** Resultado de un proceso de correspondencia entre ontologías. Está formado por un conjunto de correspondencias (*correspondences*) que establecen relaciones entre conceptos de dos ontologías. Dichas relaciones pueden ser de equivalencia, generalización, reglas de transformación, etc.
- **Ciclo de Vida del Software:** Periodo comprendido entre la concepción de un producto software y el momento en el que deja de estar disponible. Sus principales fases son la especificación de requisitos, de usuario y software, el diseño de la arquitectura del sistema, el diseño detallado de componentes, la codificación o programación del sistema y, finalmente, su paso a producción y mantenimiento.
- **Concepto Temático (*subject concept*):** Clase definida en la ontología SEMSE. Representa una clase de un esquema como concepto en la ontología específica. Posee una definición y un semset asociado por cada idioma soportado.
- **Correspondencia (*correspondence*):** Relación establecida ente dos entidades de diferentes ontologías, de acuerdo con un algoritmo de correspondencia o un individuo. Las entidades pueden ser clases, propiedades, instancias o fórmulas.
- **Correspondencia de esquemas (*schema matching* o *schema mapping*):** Comparación de dos esquemas con el objetivo de encontrar las relaciones existentes entre la semántica de sus elementos.
- **Correspondencia de ontologías (*ontology matching* o *ontology mapping*):** Comparación de dos ontologías con el objetivo de encontrar las relaciones existentes entre la semántica de sus entidades. El resultado del proceso es un *alineamiento* entre ambas ontologías.
- **Cualificador de Codificación de Sintaxis:** Elementos del conjunto de metadatos definido por el DCMI que permiten representar formatos y vocabularios controlados.
- **Cualificador de Refinamiento:** Elementos del conjunto de metadatos definido por el DCMI que permiten el refinamiento de un término original, haciendo uso de la relación de generalización. El proceso de refinamiento permite la redefinición de la semántica heredada del término padre, siempre que se encuentre dentro de la definición del término padre, esto es, sólo puede especializarla.

- **Cualificador Semántico:** Cualificador de refinamiento definido en el ámbito de la propuesta, especializado en la representación de la semántica de términos. Los cualificadores semánticos permiten la desambiguación semántica de los términos, al tiempo que mantienen la compatibilidad con los términos originales que especializan.
- **Dublin Core Metadata Initiative (DCMI):** Organización destinada al desarrollo de estándares de metadatos interoperables y multipropósito.
- **Espacio de nombres (*namespace*):** Método para la cualificación de elementos y atributos de un documento XML. Mediante la definición de un prefijo asociado al URI del recurso donde se define el elemento, se facilita su identificación unívoca y su reutilización en múltiples documentos.
- **Esquema de Lenguaje de Marcado Extensible (*XML Schema*):** Recomendación del W3C que define el metalenguaje de XML, es decir, que define la estructura y los tipos que deben tener los elementos incluidos en un documento XML.
- **Estilo Arquitectónico:** Referido a la Ingeniería del Software, es una solución práctica a un problema recurrente de diseño arquitectónico. Un estilo arquitectónico plantea una plantilla de solución que debe adecuarse al contexto del problema para su aplicación.
- **Esquema de Marco de Descripción de Recursos (*RDF Schema*):** Recomendación del W3C que define el metalenguaje de RDF, proporcionando un conjunto extensible de tipos básicos, la estructura y restricciones que debe cumplir un documento RDF.
- **Esquema de Metadatos:** Conjunto de metadatos que incluye la definición de uno o más elementos. También denominados Vocabularios de Metadatos.
- **Esquema Cualificado Semánticamente:** Representación de un esquema de metadatos que permite la inclusión de semántica de sus elementos representada mediante una ontología específica. La representación proporciona, además, una representación homogénea del esquema, al tiempo que facilita la compatibilidad con la versión original del mismo.
- **Esquema de Cualificadores Semánticos:** Esquema que alberga el conjunto de cualificadores semánticos, permitiendo su acceso a través de la definición de un espacio de nombres y el uso de URIs.
- **Exhaustividad (*recall*):** Proporción de documentos relevantes recuperados frente al total de documentos relevantes.
- **Identificador Uniforme de Recurso (*Universal Resource Identifier: URI*):** Identificador que permite la localización unívoca de recursos dentro de la Web Semántica.

- **Interoperatividad (*interoperability*):** Capacidad de algunos sistemas para intercambiar y utilizar información procedente de otro sistema diferente. Referido a metadatos, es la capacidad de intercambio de datos entre diferentes plataformas, estructuras de datos e interfaces con una mínima pérdida de contenido y funcionalidad.
- **Lenguaje de Marcado Extensible (*Extensible Markup Language: XML*):** Lenguaje de marcado habitualmente utilizado para la serialización de información, con el fin de facilitar su intercambio entre aplicaciones.
- **Lenguaje de Modelado Unificado (*Unified Modelling Language: UML*):** Lenguaje estándar para el modelado de los componentes del proceso de desarrollo de aplicaciones.
- **Localizador Uniforme de Recurso (*Uniform Resource Locator: URL*):** Identificador unívoco, o dirección, de un recurso en Web.
- **Localizador Uniforme de Recurso Persistente (*Persistent Uniform Resource Locator: PURL*):** Servicio que garantiza la accesibilidad a un recurso, mediante la definición de URLs permanentes, independientemente de los cambios en la dirección real del recurso.
- **Marco de Descripción de Recursos (*Resource Description Framework: RDF*):** Recomendación del W3C que define un lenguaje simple para expresar relaciones en forma de ternas Recurso-Propiedad-Valor.
- **Mediador (*mediator*):** Módulo software que proporciona interoperabilidad entre fuentes de conocimiento heterogéneas. En aplicaciones de consulta, incorpora dos pares de traductores que transforman la consulta de una ontología a otra y viceversa.
- **Metadato:** Descripciones asociadas a la información que permiten la recuperación, utilización o gestión de recursos de información.
- **Mezcla de ontologías (*merging*):** Proceso de creación de una nueva ontología a partir de dos o más ontologías. Implica la definición de un alineamiento entre las ontologías a mezclar y el resultado es la unión mínima de las ontologías mezcladas, esto es, eliminando entidades replicadas.
- **Ontología:** Teoría lógica que da cuenta del significado intencional de un vocabulario formal, es decir, de su compromiso ontológico hacia una conceptualización particular del mundo. Una ontología necesariamente incluirá un vocabulario de términos y alguna especificación de sus significados.
- **Ontología de Representación:** Ontología que proporciona conceptos subyacentes a los paradigmas o formalismos de representación de conocimiento. De este modo, proporciona el vocabulario necesario para modelar otras ontologías, utilizando un paradigma de representación determinado.

- **Ontología Genérica o Meta-ontología:** Ontología que proporciona términos genéricos reutilizables en diferentes dominios.
- **Ontología de Dominio:** Ontología que expresa conceptos específicos de un dominio determinado. Típicamente incluye conceptos que provienen de la especialización de conceptos existentes en ontologías genéricas.
- **Ontología de Aplicación:** Ontología que contiene todas las definiciones necesarias para modelar el conocimiento requerido por una aplicación particular. Típicamente, incluye conceptos tomados de ontologías de dominio y genéricas.
- **Ontología de Referencia:** Ontología genérica que, en el ámbito de la propuesta, permite la desambiguación semántica de términos de los esquemas a través de la correspondencia de sus términos y su extensión.
- **Ontología Específica:** Ontología de dominio dirigida a capturar la semántica concreta de un esquema de metadatos. Junto con el esquema cualificado semánticamente, al cual complementa, proporciona una representación de la semántica que facilita su gestión, al tiempo que cubre aspectos como la sinonimia y el plurilingüismo.
- **Ontología SEMSE:** Ontología de representación diseñada para la generación de ontologías específicas en el ámbito de la propuesta. También alberga la definición de las reglas de correspondencia y permite su extensión.
- **OWL Web Ontology Language (OWL):** Recomendación del W3C que define un lenguaje para facilitar la representación y el procesamiento automático de información en la Web Semántica. Posee tres especies que proporcionan distinto nivel de expresividad, de mayor a menor: OWL Full, OWL-DL y OWL Lite.
- **Patrón de Diseño:** Referido a la Ingeniería del Software, es una solución práctica a un problema recurrente en el diseño detallado de componentes. Un patrón de diseño proporciona una plantilla con la solución al problema, que debe adecuarse al contexto del problema para su aplicación.
- **Perfil de Aplicación:** Esquema definido a partir de elementos obtenidos a partir de otros esquemas. Pueden agregar y/o refinar metadatos provenientes de distintos vocabularios con la restricción de no definir elementos nuevos.
- **Precisión (*precision*):** Proporción de los documentos relevantes recuperados frente al total de documentos recuperados. Esto es, proporción de documentos recuperados que son relevantes.
- **Primitiva:** Palabra que representa una función interpretable por computadoras, definida en un lenguaje de programación o representación de conocimiento.
- **Proceso de Desarrollo Software:** Proceso que establece el marco de trabajo, tanto técnico como de gestión, para poder aplicar los métodos, herramientas y personas a la tarea de desarrollo de software.

- **Propiedad Temática (*subject property*):** Clase definida en la ontología SEMSE. Representa una propiedad de un esquema como concepto en la ontología específica. Posee una definición y un semset asociado por cada idioma soportado.
- **PROto ONtology (PROTON):** Ontología fundacional con una estructura jerárquica, modular y extensible, diseñada para su publicación en el entorno Web. Está definida mediante OWL Full y ha sido seleccionada en la propuesta como ontología de referencia.
- **Registro de Metadatos:** Sistema basado en repositorios que proporciona funciones de gestión, recuperación, referencia y reutilización de vocabularios de metadatos.
- **Relevancia:** Concepto que expresa si un documento resuelve una necesidad informativa, esto es, si debería formar parte a o no del conjunto de resultados de una consulta.
- **Repositorio Semántico:** Sistema software de soporte para el almacenamiento y gestión de ontologías. La posibilidad de gestión de la persistencia de ontologías es lo que ha dado nombre a este tipo de sistemas.
- **Semset:** Conjunto de etiquetas asociado a un concepto o propiedad temáticos. El conjunto de etiquetas incluye una etiqueta preferente y cero o más etiquetas alternativas. Se define un semset por cada idioma soportado.
- **Traductor (*translation*):** Programa que transforma fórmulas del contexto de una ontología en fórmulas del contexto de otra. Típicamente se definen como un conjunto de reglas de transformación.
- **Transformador (*transformation*):** Programa que transforma una ontología de un lenguaje de definición a otro.
- **Web Semántica (*semantic web*):** Extensión de la Web original que incluye la capacidad de procesamiento automático de la información por parte de computadoras. Para esto se requiere que la información posea un significado bien definido y que esté expresada formalmente, facilitándose así la cooperación por parte de personas y computadoras.
- **World Wide Web Consortium (W3C):** Comunidad internacional que desarrolla estándares encaminados a lograr el desarrollo a largo plazo de la Web.

Anexo A: Plantilla de descripción de Esquemas de Metadatos

Título del Esquema

Título:	<i>Título del esquema</i>
Autor:	<i>Autor del esquema</i>
Identificador:	<i>Identificador del esquema, p.e. nombre del fichero</i>
Fecha de creación:	<i>Fecha de creación del documento</i>
Última versión:	<i>URI a la última versión del documento</i>
Sustituye a:	<i>URI a la versión del documento sustituida por esta</i>
Traducciones:	<i>URI a la página con traducciones de este documento</i>
Estado del documento:	<i>Estado del documento relativo a su aprobación</i>
Descripción:	<i>Breve descripción del objetivo del documento</i>
Fecha de aprobación:	<i>Fecha de aprobación del documento (en su caso)</i>
Lenguaje	<i>Lenguaje en el que se describe el documento</i>

<Salto de página>

1. Introducción

1.1. Objetivo del Esquema

Incluir un breve resumen del objetivo del esquema.

1.2. Ámbito del Esquema

Incluir una breve descripción del ámbito de publicación y uso del esquema.

1.3. Definiciones, Acrónimos y Abreviaturas

Incluir definiciones, acrónimos y abreviaturas que faciliten la interpretación del esquema.

1.4. Referencias

Incluir referencias útiles para la interpretación del esquema y a otros esquemas utilizados en su definición.

1.5. Metadatos de los elementos

Describir los atributos necesarios para la definición de cada elemento del esquema. Se recomienda la definición de un conjunto mínimo y uno recomendado adicional.

1.5.1. Metadatos Mínimos

Conjunto mínimo de atributos que todo elemento debe incluir en su definición. De este modo, para considerarse definido, todo término deberá proporcionar información sobre todos y cada uno de los siguientes atributos.

Nombre:	<i>Identificador unívoco asignado al término dentro del esquema. Se recomienda, por compatibilidad, mantener el nombre del elemento original.</i>
URI:	<i>URI utilizado para identificar unívocamente un término.</i>
Etiqueta:	<i>Etiqueta asignada al término.</i>
Comentario:	<i>Información adicional sobre el término o su aplicación.</i>
Refina:	<i>Referencia al término original del esquema a cualificar.</i>
Tipo de término:	<i>Tipo de término, tal y como se define en el documento que contiene los principios gramaticales.</i>
Semántica:	<i>URI al concepto de la ontología específica</i>
Esquema:	<i>URI al esquema en el que se define.</i>
Fecha de Publicación:	<i>Fecha en la que se ha publicado el término</i>

1.5.2. Metadatos Opcionales

Conjunto de atributos recomendado. Se recomienda la adición al conjunto mínimo,

siempre que sean de aplicación.

Ver:	Enlace a la documentación autorizada.
Referencia:	Cita o URL a un recurso referenciado en "Definición" o "Comentario".
Fecha de creación:	Fecha en la que se definió por primera vez.
Más general:	Referencia de un término más general a uno más específico, de un vocabulario.
Más específico:	Referencia de un término más específico a uno más general, de un vocabulario.

2. Esquema de Metadatos

Para cada uno de los elementos del esquema, definir los valores de los atributos mínimos e incluir, opcionalmente, aquellos de los recomendados que sean aplicables.

3. Documento de Gramática

Extracto de lo que sería el documento de gramática, necesariamente, accesible vía web. Se debe incluir el extracto del documento de gramática para aquellos tipos de términos definidos para el esquema y no incluidos en el esquema original. En el caso de no definirse nuevos tipos, el apartado no será de aplicación y deberá etiquetarse como "No Aplicable". Las referencias a los documentos de gramática donde se definen los tipos utilizados en el esquema original, deberán estar incluidas en el apartado de referencias. En caso de no encontrarse en el documento original, se deberá incluir la definición de elemento semántico:

"Elemento Semántico: Un Elemento Semántico es una propiedad de un recurso que comparte el significado de un elemento particular DCMI estableciendo, de manera no ambigua, la semántica del recurso. Los Elementos Semánticos pueden ser considerados como Elementos de Refinamiento, también llamados Cualificadores. Los Elementos Semánticos pueden también ser utilizados en registros de metadatos, independientemente de las propiedades que desambiguan. Un Elemento Semántico refina sólo una propiedad DCMI padre."

4. Codificación RDF (opcional)

En caso de contar con la versión RDF del esquema cualificado semánticamente, o una vez generada, se recomienda la inclusión de la misma en este apartado.

Anexo B: Resultados de la evaluación de repositorios semánticos

A continuación se describen los criterios utilizados en la evaluación:

- **API para Gestión de Ontologías:** Indica si el servidor utiliza APIs para la gestión de ontologías.
- **Lenguajes de Consultas:** Lenguajes en los que se pueden hacer consultas contra las ontologías desde el servidor.
- **Lenguaje de Representación:** Lenguaje utilizado para la representación de conocimiento en cada uno de los servidores.
- **Lenguaje de Implementación:** Lenguaje en el que se ha implementado el servidor.
- **Soporte de Inferencias (motor):** Determina si el servidor soporta reglas de inferencia y por tanto, lleva a cabo algún tipo de razonamiento con motores de inferencias.
- **Características del Modelo:** Conjunto de características que definen el modelo de datos de las ontologías. Se suele indicar si el modelo soporta jerarquías, qué elementos forman parte del modelo, en qué está basado el modelo, etc.
- **Escalabilidad:** Determina si un editor ofrece plugins o mecanismos que permiten extender su funcionalidad.
- **Soporte Multi-usuario:** Determina si el servidor permite a varios usuarios (dos o más) el acceso y edición de ontologías simultáneamente.
- **Almacenamiento en Memoria:** Determina si el servidor puede almacenar ontologías en memoria.
- **Almacenamiento en BBDD:** Determina si el servidor puede almacenar ontologías en una base de datos. Además, si se conoce, se puede indicar la tipología de las BBDDs en las que permite almacenar ontologías.
- **Interfaz Gráfica:** Determina si el servidor en cuestión ofrece una interfaz gráfica de usuario o no.
- **Plataforma Ejecución:** Indica si es multiplataforma, es decir, si puede ejecutarse sobre entornos Windows, Linux, etc.
- **Compatible con otros Sistemas:** Define sistemas con los que se puede utilizar el servidor de manera conjunta (repositorios, editores de ontologías, etc.).
- **Última Versión Estable:** Indica el número y año en el que se ha publicado la última versión estable del servidor en cuestión.
- **Licencia:** Define el tipo de licencia necesaria para poder adquirir el servidor. También se indica si el software es gratuito o no.
- **Lenguajes de Ontologías que Soporta:** Lenguajes soportados en la importación y exportación de ontologías.
- **Soporte Comparación Ontologías:** Determina si el servidor permite establecer

mecanismos para comparar dos o más ontologías entre sí.

- **Soporte Fusión Ontologías:** Determina si el servidor permite fusionar ontologías.
- **Soporte Léxico:** Determina si el servidor trabaja con información léxica y/o si posee repositorios léxicos que utiliza.

Las siguientes tablas muestran, agrupadas por conjuntos de criterios, los valores de cada repositorio semántico.

Primer conjunto de criterios

Servidor	API Gestión Ontologías	Lenguajes de Consultas	Lenguaje de Representación	Lenguaje de Implementación
OWLIM	OWLIM se ha implementado por OntoText y usa la API de Sesame como su principal interfaz.	Se utiliza como lenguaje SPARQL.	Soporta RDF(S), OWL-DLP y OWL-Horst. Al cargarlo en memoria desde la BBDD, es capaz de gestionarlo tal cual (por ejemplo, razonar con TRREE)	Java.
Plataforma KIM	KIM Server tiene una API para acceso remoto e integración.	Consulta de ontologías, de metadatos (anotaciones), documentos, etc. (y de manera más general, conocimiento). Para ello se utiliza el lenguaje que usa Sesame: SPARQL.	RDF(S) y extensiones posibles a OWL Lite.	Java.
Sesame	Hay dos APIs programáticas (que las programa el propio usuario): API del repositorio y API gráfica.	SeRQL, RQL y RDQL.	RDF.	Java.
Jena	La versión 2 de Jena incluye una API, también programática, para el manejo de ontologías y soporte del lenguaje OWL.	SPARQL.	RDF, XML y N-Triples.	Java.

KAON	Sí.	RDBMS para consultas contra ontologías OWL y un subconjunto del lenguaje SPARQL.	KAON (extensión propietaria de RDFS).	Java.
IODT	Sí (EODM).	Conjunctive Query, que es un subconjunto de SPARQL.	RDF(S) y OWL. Está basado en el Metamodelo de Definición de Ontologías de OMG (ODM).	Java (implementado en el Marco de Trabajo de Eclipse para el Modelado o EMF).
AllegroGraph	Sí, con mejoras de la API de Java: Java, HTTP de Sesame, Lisp, Pitón y Ruby.	SPARQL y Prolog.	RDF.	Java.
Ontolingua Server	Sí (OKBC).	Knowledge Querying and Manipulation Language (KQML).	KIF.	No se especifica.
OntoSaurus	Tanto Loom como PowerLoom tienen una API para la gestión de ontologías. En función de los permisos que tenga el usuario, podrá sólo visualizar o visualizar y editar.	KIF.	Loom y PowerLoom.	Java.
DOME	Sí.	SPARQL.	WSML.	Java.

Segundo conjunto de criterios

Servidor	Soporte de Inferencias (Motor)	Características del Modelo	Escalabilidad
OWLIM	Sí, TRREE o BigTRREE realiza un razonamiento y evaluación de consultas directamente contra la imagen permanente del repositorio (que puede ser Sesame).	<p>OWLIM es muy usado para clasificar ontologías que contienen gran cantidad de elementos. Este tipo de modelos suelen ser casi imposibles de manejar con otros sistemas.</p> <p>OWLIM proporciona completo soporte para RDF(S) y OWL DLP: soporta OWL Horst, que es más expresivo que OWL DLP y se adapta 100% a RDFS.</p>	Sí, mediante plugins (o componentes).
Plataforma KIM	Sí, ya que ofrece soporte para múltiples motores de búsqueda u otras fuentes de datos.	KIM analiza textos y reconoce referencias a entidades (como personas, organizaciones, localizaciones, fechas). Por tanto, intenta hacer corresponder la referencia con una entidad conocida, teniendo una única URI y descripción. Alternativamente, se generan una nueva URI y descripción. Finalmente, la referencia en el documento es anotada con la URI de la entidad. A este proceso se le llama anotación semántica.	Sí, y ha quedado demostrado porque es una plataforma utilizada en proyectos para tratar grandes volúmenes de datos. Además, el equipo de desarrollo extendió KIM con el Clúster de Arquitectura para poder conseguir la escalabilidad del rendimiento de la plataforma.
Sesame	Sí, ya que proporciona soporte para razonamiento.	<p>Sesame proporciona las herramientas necesarias para analizar gramáticamente, interpretar, consultar y almacenar la información, embebida en una aplicación o que reside en una base de datos.</p> <p>Sesame se usa como motor para almacenar todos los datos relacionados con el dominio, usando tripletas RDF. Las tripletas almacenadas en Sesame representarían el <i>punto de unión</i>.</p>	Sí.

Jena	<p>Sí, incluye varios razonadores:</p> <ul style="list-style-type: none"> - Razonador OWL - Razonador DAML - Razonador de Reglas RDF - Razonador de Reglas Genéricas 	<p>El Modelo general de ontologías permite el acceso a las sentencias en una colección de datos RDF. OntModel (en Jena), además de permitir dicho acceso, añade soporte para los diferentes tipos de objetos que pueden aparecen en una ontología: clases (en una jerarquía de clases), propiedades (en una jerarquía de propiedades) e individuos.</p>	Sí.
KAON	Sí, DIG.	<p>Extiende RDFS con relaciones simétricas, transitivas e inversas, cardinalidad de relaciones, meta-modelos, etc. Similar a F-Logic usando patrones axiomáticos. El editor soporta jerarquías de conceptos, etc.</p>	<p>Usa una representación de base de datos escalable para el almacenamiento de ontologías KAON. Acceso también es escalable.</p>
IODT	<p>Sí, el módulo de inferencia se compone de los siguientes elementos:</p> <ul style="list-style-type: none"> - Interfaz DIG - Razonador DL para inferencia T-Box <ul style="list-style-type: none"> ➤ Pellet ➤ Racer ➤ Motor estructural T-Box - Conjunto de reglas lógicas para inferencia A-Box 	<p>Esta herramienta incluye EODM (EMF Ontology Definition Metamodel) para realizar modelos OWL y RDF. El metamodelo se obtiene de ODM y es implementado en el Marco de Trabajo de Eclipse para el Modelado (EMF). Además, dispone de un repositorio de ontologías llamado Minerva.</p> <p>Distingue entre propiedades y relaciones; define contextos; razonamiento por defecto; relaciones temporales en el modelo; relaciones de alta cardinalidad; metapropiedades y metarelaciones.</p>	<p>Sí, por ejemplo un repositorio de ontologías escalable que soporta razonamiento en SPARQL y DLP.</p>
AllegroGraph	<p>Sí. Soporta razonamiento RACER, RACER Pro, RDFS, OWL, Prolog.</p>	<p>Cada modelo es un conjunto de tripletas, aunque realmente se almacenan cinco slots: sujeto, predicado y objeto, slot gráfico e identificador único.</p>	Sí.
Ontolingua Server	Sí.	<p>Modelo OKBC con axiomas KIF completos.</p>	Sí.
OntoSaurus	Sí.	<p>Ofrece un rico navegador Web para bases de conocimiento con sencillas operaciones para edición, contextos, propiedades <i>same-class-as</i>, metaclasses, etc.</p>	<p>Sí, gracias al uso de Loom y sobretodo de PowerLoom (más escalable).</p>

DOME	Sí.	Conceptos, relaciones y restricciones son alineadas contra especificaciones ER.	Sí.
-------------	-----	---	-----

Tercer conjunto de criterios

Servidor	Soporte Multi-Usuario	Almacenamiento en Memoria	Almacenamiento en BBDD	Interfaz Gráfica
OWLIM	Sí.	Sí, con la versión estándar SwiftOWLIM.	Sí, con la versión BigOWLIM viene empaquetado como una capa SAIL.	Sí.
Plataforma KIM	Sí.	Sí (Sesame se puede configurar en memoria).	Sí. Además, KIM persigue objetivos similares a los de los sistemas RDBMS y Gestión de Conocimiento.	Sí.
Sesame	Sí.	Sí.	Puede usarse como base de datos relacional para RDF y RDF Schema, o como una librería Java para aplicaciones que necesitan trabajar con RDF internamente.	Sí.
Jena	Sí.	Sí, y además asegurando persistencia.	Sí, y además asegurando persistencia.	Sí.
KAON	Sí.	Sí.	Sistema de Gestión de BBDD Relacionales (RDBMS).	Sí.
IODT	Sí.	Sí. El repositorio de IODT se llama Minerva.	BBDD Relacionales.	Sí.
AllegroGraph	Sí.	Sí.	Si, con una base de datos gráfica basada en RDF. También soporta consultas contra bases de datos distribuidas.	Sí.
Ontolingua Server	Sí.	Sí.	BBDD Relacionales.	Sí.
OntoSaurus	Sí.	Sí, utilizando el sistema PowerLoom.	Sistema de Gestión de BBDD Relacionales (RDBMS).	Sí.
DOME	Sí.	Sí.	Sí.	Sí.

Cuarto conjunto de criterios

Servidor	Plataforma de Ejecución	Compatible con otros Sistemas	Última Versión Estable	Licencia
OWLIM	JDK 1.6.	Sí. Por ejemplo, con Sesame, ya que BigOWLIM está empaquetado como una capa SAIL para la base de datos Sesame RDF. Además, es utilizado como razonador por TopBraid Componder (uno de los entornos de edición más avanzados en RDF(S) y OWL).	2.8.2 (2006).	Se pueden conseguir copias de evaluación en caso de serio interés. BigOWLIM: disponible bajo licencia comercial. No es gratuito pero sí que es de código abierto. La versión gratuita es SwiftOWLIM, y se puede obtener bajo licencia LGPL.
Plataforma KIM	Java 1.4 (o superior) y sobre Linux, Windows y Solaris. También se ha probado sobre plataformas AMD64 (con JDK 1.5), ofreciendo buenos resultados.	Sí. Por ejemplo, con Sesame, con OWLIM, con Lucene, etc.	1.7.12.15 (2006).	No es de código abierto ni tampoco gratuito para uso libre o comercial. Sin embargo, Ontotext lo proporciona libre de cargo para usuarios registrados y para propósitos de investigación y evaluación.
Sesame	Sesame puede ser desplegado en cualquier sistema que proporcione contenedores Java (Apache Tomcat, JBoss, etc.).	Sí, se puede unir con una gran variedad de aplicaciones.	2.0 (2007).	Sesame 2.x está disponible bajo licencia BSC-style (es equivalente a la licencia MIT). Además, es de código abierto.
Jena	Java, por tanto es multiplataforma.	Sí, con una gran variedad, como por ejemplo: Sesame, MySQL, Oracle, PostgreSQL, etc.	2.5.4 (2007).	Jena es de código abierto y es gratuito.

<p>KAON</p>	<p>Multiplataforma (Java J2SE 1.4.0_01 o superior): Linux y Windows.</p>	<p>Sí. Se ha probado con MS SQL Server 2000. También se ha ejecutado con éxito sobre PostgreSQL, IBM DB2 7.2 y Oracle 9i. Además, funciona con los servidores de aplicaciones JBoss y Tomcat.</p>	<p>1.2.7 (2005).</p>	<p>Gratuita. GNU Lesser General Public Licence (LGPL).</p>
<p>IODT</p>	<p>Eclipse 3.1.0</p> <p>SSOO: Windows y Linux.</p> <p>Software probado con éxito sobre JRE 1.4.2: Repositorio de ontologías OWL (Minerva), y el Workbench de EODM.</p> <p>Para usar el Workbench de EODM: GEF 3.0.0.</p> <p>Para usar el repositorio de ontologías: Puede ser necesario un sistema RDBMS de los siguientes para almacenamiento:</p> <ul style="list-style-type: none"> • DB2 8.2 • Apache Derby 10.0.2.1 • HSQLDB 1.8.0 	<p>Con todos los que forman parte de la plataforma de ejecución.</p>	<p>1.1.2 (2006).</p>	<p>Es necesario registrarse para poder descargar una versión de la herramienta. No es de código abierto.</p>

AllegroGraph	<p>AllegroGraph se puede ejecutar sobre todas las arquitecturas populares de 64-bit.</p> <p>La versión de Java/HTTP de AllegroGraph proporciona una interfaz HTTP para la carga y recuperación de tripletas RDF, y una API Java para desarrollar aplicaciones semánticas en Java de gran escala.</p> <p>Multiplataforma. Se puede ejecutar en Windows, Linux, Solaris y Mac OS.</p>	<p>Sí, por ejemplo la versión Java se puede montar sobre Sesame.</p>	<p>2.2.4 (2007).</p>	<p>Hay una versión gratuita de AllegroGraph (sólo es necesario registrarse), aunque limitada a 50 millones de tripletas.</p> <p>También es de código abierto.</p>
Ontolingua Server	<p>Multiplataforma, basado en Java.</p>	<p>Sí.</p>	<p>1.0.650 (2002).</p>	<p>Gratuita.</p>
OntoSaurus	<p>Multiplataforma, basado en Java.</p>	<p>Sí, con sistemas como: Loom, PowerLoom, OKBC, etc.</p>	<p>Con PowerLoom, la última versión estable se publicó en 2006: PowerLoom 3.2.</p> <p>Pero la última versión estable de (sólo) OntoSaurus es la 1.9, publicada en 2002.</p>	<p>Gratuita.</p>
DOME	<p>Multiplataforma (Windows y Linux), basado en Java.</p>	<p>Sí, con sistemas como por ejemplo, ORDI (Ontology Representation and Data Integration).</p>	<p>0.2.0 (2005).</p>	<p>De código abierto. Bajo licencia MIT.</p>

Quinto conjunto de criterios

Servidor	Lenguajes de Ontologías que Soporta	Soporte Comparación Ontologías	Soporte Fusión Ontologías	Soporte Léxico
OWLIM	RDF, RDFS, OWL DLP, OWL Lite y OWL Horst. OWLIM puede ser configurado para ejecutarse de tres modos: OWL-max, OWL Horst y RDFS (dependiendo de qué tipo de expresividad se necesite en la aplicación).	Sí, con la versión BigOWLIM.	Sí, pero sólo BigOWLIM.	Sí, ya que OWLIM es un repositorio semántico en KIM (que dispone de información léxica en la <i>KIM Lexical Ontology</i>).
Plataforma KIM	OWL, OWL Lite, RDF y RDFS.	Sí, si utiliza como repositorio semántico BigOWLIM.	Sí, si utiliza como repositorio semántico BigOWLIM.	Sí, ya que dispone de gran cantidad de información léxica en la <i>KIM Lexical Ontology</i> .
Sesame	RDF y RDFS.	Sí.	Sí.	Sí.
Jena	RDF, RDFS, DAML, y OWL.	Sí.	Sí.	Sí.
KAON	RDFS (KAON1) y OWL-DL (KAON2).	Sí. Herramienta OntologyComparison, para comparar dos ontologías.	Sí.	Si. Hay una capa léxica.
IODT	OWL y RDFS.	No.	Sí.	Sí, por ejemplo permite sinónimos.
AllegroGraph	RDF, RDFS, XML, OWL, N-Triples, Java, HTTP de Sesame, Lisp, Pitón y Ruby.	No.	Sí.	Sí.

Ontolingua Server	<p>Importa y exporta: DAML+OIL, KIF, OKBC, Loom, Prolog, Ontolingua, CLIPS.</p> <p>Importa sólo: Classic, Ocelot, Protégé.</p>	<p>Sí, Chimaera es el soporte para evolución y comparación de ontologías.</p>	<p>Sí. A través de Chimaera.</p>	<p>Sí. Permite la búsqueda de términos en todas las ontologías cargadas.</p>
OntoSaurus	<p>Incluye herramientas para la traslación de ontologías Loom/PowerLoom en otros lenguajes KR y en esquemas para lenguajes de programación como: C++, CLOS, interfaz de especificaciones IDL, y, objetos Java.</p>	<p>Sí, cuando tiene como sistema de representación del conocimiento a PowerLoom, es un sistema que ofrece herramientas y técnicas para comparar, entre otros elementos, ontologías desarrolladas de manera independiente.</p>	<p>Sí, utilizando PowerLoom.</p>	<p>Sí, cuando se combina con PowerLoom, ofrece herramientas para importar esquemas de datos, diccionarios de datos, léxico y bases de conocimiento, para crear ontologías y esqueletos.</p>
DOME	<p>WSML, RDF y OWL.</p>	<p>Sí, soporta correspondencias contra ontologías.</p>	<p>Sí.</p>	<p>Sí.</p>