

UNIVERSIDAD DE MURCIA



***Facultad de Comunicación y
Documentación
Departamento de Información y
Documentación***

TESIS DOCTORAL

**Diseño de un sistema colaborativo para
la creación y gestión de tesauros en
Internet basado en SKOS.**

JUAN ANTONIO PASTOR SÁNCHEZ

Directores:

Dr. D. José Vicente Rodríguez Muñoz
Dr. D. Francisco Javier Martínez Méndez

2009

*Ana, esto ha sido posible gracias a ti,
por el amor que me das todos los días,
por el apoyo que he recibido sin condiciones,
por creer en mí más que yo mismo*

Agradecimientos

Si soy sincero nunca me he imaginado escribiendo estas palabras de agradecimiento para aquellas personas que han estado a mi lado durante los largos años de reflexión, maduración y elaboración de esta tesis. En realidad este trabajo también ha sido posible gracias a ellos y su recuerdo me acompañará a lo largo de toda mi vida.

Doy las gracias a mis directores de Tesis, Javi y José Vicente, compañeros y amigos, a los cuáles conozco, quiero y admiro desde hace casi veinte años. Soy consciente que os he decepcionado en más de una ocasión, y sin embargo habéis permanecido a mi lado cuando os he necesitado de forma incondicional. Espero que algún día pueda devolveros una pequeña parte de lo que habéis sembrado en mí.

También agradezco el amor y dedicación de mis Padres, que hicieron posible que iniciara el recorrido en este largo camino. Lo habéis dado todo por mí y soy consciente de la ilusión que os hace verme conseguir esta meta.

Por supuesto a Ana, mi esposa, a la que dedico este trabajo, porque ha sido un sueño que ha compartido conmigo. Tu amor, cariño y paciencia han sido alimento y cura para mi alma durante todos los días en los que has estado a mi lado dándome ánimos.

A Tomás, por sus trabajos, ideas, conversaciones e imaginación que han orientado en algún momento mis pasos y mi inspiración, a Pascual por su apoyo y consejos en los momentos más duros durante la redacción de la tesis y a Javier cuyo trabajo y ejemplo han sido fundamentales para mí.

A mis amigos y compañeros Rosana, Goyo y Pedro que siempre han tenido momentos para escucharme, y a todos aquellos compañeros de departamento y facultad que me han dado ánimos y consejos a lo largo de estos años.

Y en definitiva a todos mis amigos que han visto como he desaparecido, más que de costumbre, durante los últimos meses.

A todos ellos mi cariño y mi agradecimiento más sincero.

*¿Qué es la palabra? Un signo arbitrario.
Pero vivimos en las palabras.
Nuestra realidad, entre palabras, no entre cosas.
No existe cosa tal como una cosa, de cualquier modo; una forma en la mente.
Entidad... sensación de sustancia. Una ilusión.
La palabra es más real que el objeto que representa.
La palabra no representa la realidad.
La palabra es la realidad.
Para nosotros, de cualquier modo.*

Philip K. Dick

Resumen

La presente tesis se enmarca dentro de los procesos de recuperación de información en el ámbito de la Web, más concretamente mediante la aplicación de tesauros y orientado hacia las propuestas de representación de información realizadas desde la Web Semántica, planteando como hipótesis de trabajo la creación de un modelo dirigido al desarrollo de aplicaciones web para la gestión de tesauros.

Las tecnologías actualmente utilizadas para la elaboración de la mayor parte de los contenidos Web carecen de estructuras que permitan una representación adecuada de los aspectos semánticos. Los motores de búsqueda utilizan técnicas basadas en la frecuencia de aparición de palabras en los procesos de recuperación, por lo que las nuevas herramientas asociadas a la Web 2.0, y la heterogeneidad y dinamismo de los contenidos de información, han hecho patentes ciertos problemas inherentes a este modelo de recuperación. La Web Semántica constituye una solución viable, fundamentada en el uso de tecnologías como XML, RDF y OWL, en donde los metadatos y las ontologías constituyen el núcleo conceptual de desarrollo. La descripción de la semántica de los recursos de información resulta bastante compleja debido al estado de desarrollo actual de la Web, pero es posible dar un paso previo al realizar una descripción y caracterización de dichos recursos empleando metadatos y esquemas conceptuales, tales como los tesauros. El Tesauro ha evolucionado como una herramienta de recuperación de información y actualmente se considera de gran utilidad en los procesos de organización del conocimiento. Las tecnologías de la información han sido partícipes de dicho proceso evolutivo, aunque su aplicación en la Web ha resultado poco fructífera. La Web Semántica supone una oportunidad para utilizar los tesauros en los procesos de descripción y recuperación de información. Muestra de ello son las múltiples iniciativas de representación de tesauros mediante RDF, que han dado lugar al desarrollo de SKOS, un modelo para esquemas conceptuales aplicables en la Web Semántica y que actualmente se encuentra en desarrollo por el W3C.

En este trabajo se realiza un el análisis de los requisitos previos de una aplicación web para el mantenimiento y uso de tesauros. Para ello, se han analizado diversas tecnologías de desarrollo de aplicaciones web, así como las pautas y criterios aportados por ciertas disciplinas técnicas como la Accesibilidad, la Usabilidad y la Arquitectura de la Información. También se ha llevado a cabo una descripción funcional de las principales características de una aplicación de este tipo, que integre las funciones de consulta y mantenimiento. Para la formalización de la propuesta se han confeccionado los diagramas UML de casos de uso y de clases necesarios para modelar la estructura estática de la aplicación, así como los aspectos dinámicos relacionados con la organización funcional desde el punto de vista del usuario. Consiguientemente, se ha obtenido un modelo conceptual para el desarrollo de una aplicación orientada a la gestión de tesauros de forma colaborativa y su aplicación en la indexación de recursos de información, agrupados en repositorios y descritos mediante metadatos.

Palabras Clave

Accesibilidad / Aplicaciones web / Arquitectura de la Información / Dublin Core / Esquemas conceptuales / Internet / Metadatos / Modelos de recuperación de información / Ontologías / RDF / Recuperación de información / Servicios web / Sistemas colaborativos / SKOS / Tesauros / Usabilidad / Web / Web Semántica / XML

Sumario

1. Introducción.....	3
1.1. Antecedentes.....	3
1.2. Motivaciones.....	5
1.3. Principales aportaciones.....	9
1.4. Hipótesis de trabajo y metodología utilizada.....	10
1.5. Estructura.....	15
2. Recuperación de Información y Web Semántica.....	19
2.1. Definición de Recuperación de Información.....	19
2.2. Modelos de recuperación de información.....	23
2.3. Los sistemas de recuperación de información.....	27
2.4. La Recuperación de Información en la Web.....	31
2.4.1. Motores de búsqueda y directorios web.....	32
2.4.2. Búsqueda mediante el análisis de la estructura hipertextual.....	34
2.4.3. Metabuscadores.....	35
2.4.4. Etiquetado colaborativo.....	36
2.5. La Web Semántica.....	37
2.5.1. Definición y objetivos de la Web Semántica.....	37
2.5.2. La Arquitectura de la Web Semántica.....	41
2.5.3. Metadatos y ontologías.....	46
2.6. Tecnologías de la Web Semántica para la Representación de la Información.....	52
2.6.1. XML.....	52
2.6.2. RDF.....	58
2.6.3. OWL.....	64
2.7. Web 2.0 y Web Semántica: el camino hacia la Web Ubícua.....	67
2.8. Esquemas conceptuales en la Recuperación de Información.....	69
3. Representación de Tesoros con SKOS.....	79
3.1. El concepto de Tesoro.....	79
3.2. Características principales de los tesoros.....	85
3.3. La evolución del concepto de Tesoro.....	86
3.4. Antecedentes de representación de tesoros en la Web con RDF.....	95
3.4.1. LIMBER.....	96
3.4.2. ILRT.....	96
3.4.3. CERES.....	96
3.4.4. GEM.....	97
3.4.5. CALL Thesaurus Ontology	97
3.4.6. AGROVOC/KAON.....	97
3.4.7. ETB: European Treasury Browser.....	98
3.5. Modelo de representación de tesoros para la Web Semántica.....	98
3.5.1. Tesoros basados en conceptos y tesoros basados en términos.....	99
3.5.2. Estructuras de agrupación.....	101
3.5.3. Relaciones léxicas entre etiquetas.....	101
3.5.4. Relaciones como arcos o como nodos.....	102
3.5.5. Estructuras multilingües.....	102
3.5.6. Restricciones semánticas.....	103
3.5.7. Ampliación de la tipología de relaciones.....	105

3.5.8. Equivalencia entre tesauros.....	105
3.5.9. Definiciones, explicaciones, notas de alcance y metadatos.....	106
3.6. SKOS.....	107
3.6.1. Introducción.....	107
3.6.2. Modelo SKOS.....	108
3.6.3. Conceptos y esquemas de conceptos.....	111
3.6.4. Etiquetas léxicas.....	112
3.6.5. Relaciones semánticas.....	116
3.6.6. Notaciones.....	122
3.6.7. Documentación.....	123
3.6.8. Colecciones de conceptos.....	125
3.6.9. Propiedades de mapeado.....	126
3.7. Ventajas de la aplicación de SKOS.....	129
4. Propuesta de modelo conceptual de aplicación.....	135
4.1. Entorno tecnológico de la aplicación.....	135
4.1.1. La Web como cliente/servidor universal.....	136
4.1.2. Lenguajes de programación para la Web.....	138
4.1.2.1. Programación web del lado del cliente.....	139
4.1.2.2. Programación web del lado del servidor.....	143
4.1.3. Servicios Web.....	150
4.1.4. Los sistemas de gestión de contenidos.....	152
4.1.5. Accesibilidad, Usabilidad y Arquitectura de la información.....	154
4.2. Análisis funcional de una aplicación web para la gestión y uso de tesauros.....	165
4.2.1. Indicaciones sobre la aplicación de SKOS en el diseño funcional.....	166
4.2.2. Funciones de trabajo en grupo y flujo de trabajo.....	168
4.2.3. Funciones de construcción y mantenimiento de tesauros.....	172
4.2.4. Funciones de gestión e indización de repositorios de recursos.....	177
4.2.5. Servicios web de consulta.....	178
4.3. La interfaz de usuario como núcleo funcional integrador.....	181
4.3.1. Consulta de tesauros.....	183
4.3.2. Mantenimiento de tesauros.....	187
4.3.3. Consulta de repositorios.....	189
4.3.4. Mantenimiento del repositorio.....	191
4.3.5. Indización.....	192
4.3.6. Flujo de trabajo.....	193
4.4. Formalización de la propuesta con UML.....	195
4.4.1. Diagramas de casos de uso.....	200
4.4.1.1. ESC-00 Consulta de tesauros y repositorios.....	202
4.4.1.2. ESC-01 Gestión de tesauros y repositorios.....	204
4.4.1.3. ESC-02 Funciones de flujo de trabajo.....	206
4.4.1.4. ESC-03 Configuración de tesauros y repositorios.....	207
4.4.1.5. ESC-04 Configuración del sistema.....	208
4.4.1.6. ESC-05 Servicios web.....	209
4.4.2. Diagramas de clases.....	210
4.4.2.1. CLA-00 Estructura de nodos y etiquetas.....	213
4.4.2.2. CLA-01 Estructura de etiquetas.....	215
4.4.2.3. CLA-02 Estructura de relaciones.....	216
4.4.2.4. CLA-03 Usuarios, grupos y permisos.....	218
4.4.2.5. CLA-04 Elementos de documentación.....	220
4.4.2.6. CLA-05 Elementos y asignación de metadatos.....	221

4.4.2.7. CLA-06 Indización.....	223
4.4.2.8. CLA-07 Control de flujo de trabajo.....	224
4.4.2.9. CLA-08 Marcadores.....	226
4.4.3. Diagramas de secuencia.....	227
4.5. Indicaciones para el desarrollo de la aplicación.....	229
5. Conclusiones.....	235
6. Referencias bibliográficas.....	251

Índice de ilustraciones

Ilustración 1: Descripción general de la metodología utilizada.....	12
Ilustración 2: Taxonomía de los modelos de recuperación de información según Baeza-Yates (1999)	23
Ilustración 3: Diagrama de operaciones realizadas por el usuario durante la búsqueda de información. Adaptado de Machionini (1992).....	29
Ilustración 4: Funcionamiento de un motor de búsqueda para la Web. Adaptación de Baeza-Yates y Ribeiro-Nieto (1999).....	32
Ilustración 5: Capas de la Web Semántica.....	42
Ilustración 6: Arquitectura detallada de la Web Semántica. Basado en W3C. Elaboración propia....	45
Ilustración 7: Vertientes Semántica y Pragmática de una ontología.....	51
Ilustración 8: Grafo RDF básico.....	59
Ilustración 9: Grafo RDF complejo.....	61
Ilustración 10: Constructores de RDF y RDF Schema.....	63
Ilustración 11: Constructores de OWL Lite, DL y Full.....	67
Ilustración 12: La Evolución de la Web. Traducido y adaptado de Davis (2008).....	69
Ilustración 13: Eficacia de la búsqueda en texto libre en relación a la evolución de la Web. Traducido y adaptado de Spivack (2005).....	71
Ilustración 14: Relación entre el incremento de la capacidad de búsqueda y el uso de metadatos. Traducido y adaptado de Davis (2006).....	73
Ilustración 15: La geometría del Hipertexto Documental. Adaptado de Pastor y Saorín (1995b).....	88
Ilustración 16: Niveles de abstracción del tesoro aplicados a la formulación de consultas. Adaptado y traducido de Kalervo (1996).....	89
Ilustración 17: La Web estructurada en tres niveles: conceptual, navegacional y estructura documental.....	94
Ilustración 18: Tesoro basado en términos frente a un tesoro basado en conceptos.....	99
Ilustración 19: Niveles de organización de un tesoro basado en conceptos.....	100
Ilustración 20: Relación expresada en forma de nodo.....	102
Ilustración 21: Independencia del nivel conceptual y adaptación del nivel léxico en estructuras multilingües.....	103
Ilustración 22: Ejemplos de relaciones semánticas incoherentes.....	105
Ilustración 23: Grafo correspondiente a la codificación del ejemplo anterior mediante SKOS XL y su correspondiente grafo en el que no se establecen relaciones entre entidades léxicas.....	109
Ilustración 24: Grafo correspondiente a la codificación del ejemplo 9 mediante SKOS XL y su correspondiente grafo en el que no se establecen relaciones entre entidades léxicas.....	116
Ilustración 25: Diagrama organizativo de las relaciones semánticas en SKOS.....	117
Ilustración 26: Grafo correspondiente a la codificación con SKOS del descriptor “Tesoro” según la entrada correspondiente en el Tesoro de la UNESCO.....	121
Ilustración 27: Diagrama organizativo de las relaciones semánticas y de correspondencia.....	127
Ilustración 28: El papel de SKOS y los esquemas conceptuales en los Sistemas de Información, complementado con las ontologías.....	131
Ilustración 29: Esquema básico de funcionamiento de la Web como Cliente/Servidor Universal...137	
Ilustración 30: Esquema básico de funcionamiento de AJAX.....	141

Ilustración 31: Esquema básico de funcionamiento de la tecnología JSP.....	149
Ilustración 32: Integración de la Accesibilidad, Usabilidad y Arquitectura de la Información en el desarrollo de un sitio web.....	164
Ilustración 33: Dinámica integral de las funciones de consulta y mantenimiento.....	176
Ilustración 34: Ejes funcionales del modelo con el proceso de Indización como elemento integrador	182
Ilustración 35: Niveles de proximidad en la vista de contexto de un concepto.....	184
Ilustración 36: Tipos de búsqueda conceptuales en repositorios de recursos y su posible ampliación a partir de relaciones de correspondencia.....	186
Ilustración 37: Similitud de recursos a partir de la indización conceptual.....	190
Ilustración 38: Tipología de diagramas UML.....	197
Ilustración 39: Diagrama de casos de uso de consulta de tesauros y repositorios.....	203
Ilustración 40: Diagrama de casos de uso de gestión de tesauros y repositorios.....	205
Ilustración 41: Diagrama de casos de uso de funciones de flujo de trabajo.....	206
Ilustración 42: Diagrama de casos de uso de configuración de tesauros y repositorios.....	207
Ilustración 43: Diagrama de casos de uso de configuración del sistema.....	208
Ilustración 44: Diagrama de casos de uso de servicios web.....	209
Ilustración 45: Diagrama de clases con la estructura de nodos.....	214
Ilustración 46: Diagrama de clases con la estructura de etiquetas.....	215
Ilustración 47: Diagrama de clases con la estructura de relaciones.....	217
Ilustración 48: Diagrama de clases para usuarios, grupos y permisos.....	219
Ilustración 49: Diagrama de clases para los elementos de documentación.....	220
Ilustración 50: Diagrama de clases para los elementos y asignación de metadatos.....	222
Ilustración 51: Diagrama de clases para la indización de recursos con conceptos.....	223
Ilustración 52: Diagrama de clases para los procesos de flujo de trabajo.....	225
Ilustración 53: Diagrama de clases para los marcadores del sistema.....	226
Ilustración 54: Diagrama de secuencia para el caso de uso CU-C03: Búsqueda de conceptos por etiquetas.....	228
Ilustración 55: Diagrama UML de componentes del sistema.....	230

Índice de tablas

Tabla 1: Diferencias entre recuperación de datos y recuperación de información. Fuente: Rijsbergen (1999).....	20
Tabla 2: Diferencias entre directorios y motores de búsqueda web. Fuente: Delgado Domínguez (1998)	34
Tabla 3: Ámbitos de aplicación de los metadatos.....	47
Tabla 4: Elementos de Dublin Core.....	48
Tabla 5: Equivalencia en tripletas de la ilustración 8.....	60
Tabla 6: Equivalencia en tripletas de la ilustración 9.....	61
Tabla 7: Clases y propiedades para conceptos, esquemas, pertenencia a esquemas y conceptos cabecera.....	111
Tabla 8: Vocabulario y definiciones de clases y propiedades para etiquetas léxicas.....	113
Tabla 9: Clases y propiedades para conceptos, esquemas.....	114
Tabla 10: Vocabulario y definiciones de clases y propiedades para elementos de relaciones semánticas.....	118
Tabla 11: Vocabulario y definiciones de clases y propiedades para notaciones.....	122
Tabla 12: Vocabulario y definiciones de clases y propiedades para elementos de documentacion.....	124
Tabla 13: Vocabulario y definiciones de clases y propiedades para elementos de colecciones	125
Tabla 14: Vocabulario y definiciones de clases y propiedades para elementos de colecciones	128
Tabla 15: Roles con sus respectivos permisos.....	169

Índice de ejemplos

Ejemplo 1: Código XML.....	55
Ejemplo 2: DTD correspondiente al ejemplo 1.....	55
Ejemplo 3: Definición XML Schema equivalente a la definición de DTD del ejemplo 1.....	56
Ejemplo 4: Definición de prefijos.....	60
Ejemplo 5: Código RDF/XML correspondiente a la ilustración 9 y a la tabla 6.....	62
Ejemplo 6: Definición de conceptos y asociaciones a un esquema de conceptos.....	112
Ejemplo 7: Definición de un esquema de conceptos y de conceptos cabecera.....	112
Ejemplo 8: Definición de concepto y etiquetas léxicas asociadas.....	113
Ejemplo 9: Definición de una relación entre etiquetas léxicas.....	115
Ejemplo 10: Definición de conceptos y relaciones semánticas asociadas.....	119
Ejemplo 11: Entrada del descriptor “Tesoro” en el Tesoro de la UNESCO.....	119
Ejemplo 12: Codificación con SKOS del descriptor “Tesoro” según la entrada correspondiente en el Tesoro de la UNESCO.....	120
Ejemplo 12: Codificación con SKOS del descriptor “Tesoro” según la entrada correspondiente en el Tesoro de la UNESCO.....	120
Ejemplo 13: Definición de una notación para un concepto.....	122
Ejemplo 14: Definición de concepto y elementos de documentación asociados.....	124
Ejemplo 15: Conceptos y agrupación de los mismos utilizando dos colecciones (una de ellas ordenada).....	126
Ejemplo 16: Definición de relaciones de correspondencia.....	129
Ejemplo 17: Indización conceptual de un recursos mediante el elemento dc:subject.....	178
Ejemplo 18: Ejemplos de invocación del método “etiquetaNodo”.....	212

1

Introducción

1. Introducción

En Egipto, a las bibliotecas se las denominaba "tesoro de los remedios del alma". En efecto, curábase en ellas la ignorancia, la más peligrosa de las enfermedades y origen de todas las demás. (Jacques Bénigne Bossuet.)

Resumen: En este capítulo se exponen los antecedentes y motivaciones de elaboración del presente trabajo. Asimismo se realiza una exposición de los objetivos y principales aportaciones realizadas. Finalmente se describe la hipótesis de trabajo y la metodología utilizada, junto con una descripción de la estructura discursiva de este documento.

1.1. Antecedentes

Es bien conocido que la voz "Tesoro" proviene del vocablo latino 'Thesaurus' y éste a su vez del griego 'θησαυρός' (thesauros) que significaba en su origen almacén o tesorería, habiéndosele, de alguna manera, anexado con el transcurrir de los tiempos un aura casi sagrada de almacén o repositorio del conocimiento y de la sabiduría, hasta tal punto que tesoro sigue siendo según la definición de la RAE el "nombre dado por sus autores a ciertos diccionarios, catálogos o antologías ordenadas de datos". Con el desarrollo y normalización de las Ciencias de la Información y de la Documentación, este término se elige para designar a los lenguajes documentales postcoordinados por antonomasia, constituyéndose en un "vocabulario de un lenguaje controlado de indización, organizado formalmente, de manera que las relaciones entre los conceptos se establezcan a priori, para ser usadas en sistemas de recuperación de información" (Aitchinson y Gilchrist, 1987) en la forma de "vocabulario controlado y dinámico, compuesto por términos que tienen entre ellos relaciones semánticas y genéricas y que se aplica a un dominio particular del conocimiento" (Norma ISO 2788). Así pues, la consideración del tesoro como instrumento lingüístico, concebido al estilo de un diccionario y aplicable generalmente en el campo de la creación literaria, da paso a una herramienta de control terminológico, donde los elementos se estructuran mediante relaciones conceptuales de gran utilidad en la búsqueda documental a través

de la indización de los documentos del sistema con los términos del tesoro (los descriptores).

La elaboración de Tesoros ha sido siempre una tarea compleja, debido fundamentalmente a la necesidad de coordinar distintos grupos de trabajo y al uso indistinto de múltiples fuentes de información en la elaboración de la terminología y su posterior estructuración. Una vez superados (al menos en teoría) los problemas de normalización en la construcción de estos lenguajes documentales gracias a la redacción de normas internacionales, surge la idea de facilitar esta tarea por medio de las TIC para proporcionar una mayor rapidez al proceso de creación, fomentando así su uso en los centros e instituciones documentales, tanto en las tareas de la descripción documental como a la hora de asesorar y asistir a los usuarios en sus búsquedas, que indudablemente verían mejorada su efectividad. Si bien es una razón más técnica aunque no menos trascendental, la gestión informática permite un control más eficaz sobre los términos recogidos en el tesoro, aumentando su consistencia. Finalmente, también se favorece la generación automática de distintos tipos de presentaciones e índices.

Esta amplia serie de beneficios ha propiciado el desarrollo de un conjunto de aplicaciones informáticas con este propósito, destinadas principalmente a la creación y representación de estos lenguajes. En la completa revisión del software utilizado para la gestión de tesoros realizada por Moya Martínez y Gil Leiva (2001) se evaluaban varios paquetes informáticos y se destacaba como ventaja fundamental de estos sistemas la adecuada gestión de las relaciones y de los formatos de visualización, no pudiéndose decir lo mismo de los formatos de impresión y de la gestión de las notas explicativas. Igualmente, se valoraba de forma positiva las capacidades de exportación de ficheros a otros sistemas.

En Internet es cada vez más frecuente encontrar un número interesante de Tesoros en línea, sistemas que generalmente permiten la consulta de sus términos y sus relaciones. Una amplia selección de estos tesoros aparece recogida en el trabajo de Alejandro de la Cueva (1999). La mayoría de estos tesoros que podemos encontrar en la web, a pesar de su variedad temática se agrupan básicamente en dos únicas categorías: presentaciones (sistemáticas o alfabéticas) y diccionarios terminológicos. En realidad, este tipo de tesoros sólo emplea la web como medio para difundir más sus contenidos, potenciar su visibilidad y aumentar así su utilidad. El segundo grupo

más abundante de tesauros en línea son diccionarios de términos, más o menos elaborados, que aprovechan la interactividad de la web para la realización de consultas a su contenido (por ejemplo la web Hypertext Webster Gateway¹ que proporciona una interface de consulta a algunos de principales diccionarios en línea de Internet.

Un tercer campo de actuación, algo menos profuso pero no por ello menos interesante, es la construcción automática de tesauros aunque, tal como recuerda Pérez Agüera (2007) *“cuando se repasa la literatura sobre generación automática de tesauros observamos que la mayor parte de los esfuerzos se centran en la generación de tesauros lingüísticos, siendo muy pocos los casos en los que encontramos trabajos sobre generación de tesauros documentales. Por este motivo, aunque los trabajos sobre generación de tesauros lingüísticos son muy aprovechables, es muy complicado encontrar investigaciones que tengan en cuenta todos los procesos necesarios para la generación automática de tesauros documentales”*. Así, la mayor parte de estos esfuerzos no son del todo aprovechables desde un punto de vista documental.

1.2. Motivaciones

Nosotros pensamos que si al uso de la web como medio de presentación del tesoro o como vínculo interactivo para su consulta, se le añadiera la posibilidad de realizar la construcción del propio tesoro en línea (utilizándola como soporte o medio), bien de forma individual o bien por parte de un grupo colaborativo, las prestaciones de estos sistemas aumentarían enormemente, especialmente en los procesos de mantenimiento, por no hablar de las inherentes ventajas que se obtendrían en las posteriores operaciones de recuperación de información.

Es por ello que consideramos que Internet en general, y la web en particular se constituyen en unos espacios de intercambio y colaboración científica muy adecuados para la implantación de sistemas gestores de tesauros en línea, ya que:

1. Internet permite establecer grupos de trabajo, incluso muy alejados en espacio y banda horaria, que pueden colaborar en la elaboración de un tesoro. El uso de bases de datos relacionales a través de una interfaz web, junto con el uso del correo electrónico permitiría establecer una dinámica de trabajo alrededor de una serie de flujos de trabajo para la creación y mantenimiento de términos

1 La web mencionada puede consultarse en: http://www.bennetyee.org/http_webster.cgi?

y relaciones del tesoro.

2. La filosofía cliente-servidor subyacente en todas las aplicaciones de la red Internet (y muy particularmente en la Web), permitirá construir en común un corpus léxico documental, íntegramente adaptable a las características particulares de cada usuario, por medio del desarrollo de aplicaciones cliente específicas para diversos propósitos.
3. De forma casi consuetudinaria con la propia naturaleza de la Web como sistema de información, se pueden extender las capacidades del tesoro gestionado, convirtiéndolo en un instrumento muy válido para la recuperación de información.

La irrupción de la Web ha democratizado plenamente la publicación de contenidos, permitiendo la superación del marco editorial con su serie de filtros ineludibles en la transmisión del conocimiento. Si bien esto no siempre viene acompañado de una mayor calidad en los contenidos difundidos, consideramos que la participación de los usuarios de la Web en la generación de contenidos es altamente positiva. Es un hecho que este fenómeno comunicativo/participativo está cambiando radicalmente la manera de comunicarnos y está convirtiendo en obsoletos a muchos sistemas y fuentes tradicionales de información, generando nuevos mecanismos de participación y difusión (donde destacan principalmente los wikis y los blogs, verdaderos paradigmas de lo que se ha dado en llamar Web 2.0), sin olvidar el concepto de “folksonomía” (o etiquetado libre de los documentos) mucho más cercano a nuestro campo (con todas las limitaciones que se les quieran poner, no dejan de ser una lista de descriptores libres que además permiten una inmediata recuperación de información).

Parece claro que es el contexto de la Web donde se deben aportar nuevas soluciones e ideas. Este proceso de apertura se refleja también en el protagonismo que han adquirido los grupos de trabajo cooperativos, de carácter descentralizado pero coordinados o tutelados por organismos asociados al desarrollo de la Web Semántica como el World Wide Web Consortium. Estas estructuras creadas en los últimos años se han demostrado mucho más eficaces y flexibles que los tradicionales comités científicos y técnicos que plantean resultados a más largo plazo. Se trata de una dinámica de trabajo más cercana a la realidad actual, donde interactúan comunidades de usuarios de forma espontánea a través de la Web, siendo el intercambio de ideas más importante que la formalización de propuestas, al menos en las primeras fases de

desarrollo de nuevos enfoques y tecnologías.

Asimismo, en la actualidad ha crecido exponencialmente el número de organizaciones que precisan de la gestión de contenidos para su subsistencia, dependencia que hasta ahora se presentaba más en las tradicionales instituciones documentales principalmente y en menor medida en otras instituciones académicas o de investigación (que de alguna manera representaban una “élite” dentro del campo de la gestión del conocimiento).

Las tecnologías informática y documental han resuelto en gran medida esta problemática, aunque cuando las organizaciones deben manejar grandes cantidades de documentos y necesitan más extraer información de forma pertinente y eficaz que recuperar cientos y cientos de documentos, se encuentran con la práctica ausencia de sistemas de gestión de documentos integrados con sistemas de indización por medio del manejo de lenguajes controlados.

Esta preocupación no responde únicamente a planteamientos teóricos sino más bien aplicados. A lo largo de los últimos años nuestro grupo de investigación ha participado en el desarrollo de diversos sistemas de información de gestión organizacional y hemos detectado la carencia de una solución global a este problema aplicable en la gestión de repositorios de documentos (independientemente de su presencia en entorno intranet o extranet), en entornos hospitalarios (historiales clínicos), en centros tecnológicos (documentación normativa y científica) e incluso laboratorios o centros de investigación (bibliografía y documentación especializada) y también, por qué no, en los tradicionales centros e instituciones documentales que tampoco tienen tan cubiertas sus necesidades en este campo.

Nuestra tesis de que la Web -por ser un nuevo espacio- representa por tanto una oportunidad para el desarrollo de aplicaciones para la creación de tesauros, es mucho más que un simple planteamiento, responde a una realidad en la que nos encontramos inmersos y también es fruto de una profunda reflexión y dedicación a esta línea de trabajo y de investigación. Ya en 1992, cuando finalizamos nuestros estudios de Diplomado en Biblioteconomía y Documentación, acabábamos de desarrollar el software SGAT² como implementación física de parte de los planteamientos realizados

2 Desarrollado en un entorno de programación no visual por medio del software dBASE IV y posteriorme rediseñado varias veces con entornos de programación Windows, concretamente dBFast y Visual Basic.

por Rodríguez Muñoz et al (1990) de gestión de tesauros en entornos de bases de datos relacionales. Las limitaciones a nivel de presentación y de interfaz de esta herramienta nos condujeron a una sucesiva -e inevitable- serie de rediseños de la misma, centrados tanto en los aspectos formales y de interfaz de usuario como en la necesidad de insertar la gestión del tesoro dentro del entorno de un hipertexto³ (Pastor Sánchez y Saorín Pérez, 1996).

Es verdaderamente con el advenimiento e implantación masiva de la Web (un hipertexto al fin y al cabo) cuando surge la oportunidad del desarrollo cooperativo, lo que nos llevó al desarrollo de iSGAT, sistema de creación y gestión de tesauros en línea a través de la web, aprovechando al máximo las prestaciones cliente/servidor de esta tecnología y convertirlo en un medio para una óptima recuperación de información en una colección de documentos previamente indizados (Pastor Sánchez y Martínez Méndez, 2003). Si bien esta herramienta se encuentra plenamente operativa y la vienen empleando algunos gestores de información en sus organizaciones, tanto a nivel profesional como en la realización de alguna actividad docente o taller, creemos sinceramente que la misma precisa de una mayor normalización y adaptación que la aportada a lo largo de los distintos procesos de rediseño desarrollados hasta la actualidad. En los servicios y unidades de información y documentación suelen utilizarse aplicaciones que realizan una indización a texto completo o una clasificación asistida (generalmente bastante sencilla), por medio de un software de gestión documental. Para solventar esta disfuncionalidad se han realizado varias propuestas centradas en la idea de la Web Semántica, entorno hacia el que, a pesar de lo que dicen algunos autores de forma bastante equívoca, aún estamos aproximándonos. El objetivo fundamental de estos trabajos es dotar de una mayor interacción entre usuario y máquina, mediante el uso de técnicas que permitan una descripción adecuada del contenido de recursos de información.

Con este motivo se están desarrollando iniciativas que explotan las posibilidades que ofrecen XML/RDF, las cuales -en el caso de los tesauros- están concretándose en el desarrollo y uso -por parte del World Wide Web Consortium- de la propuesta que supone SKOS⁴ para proporcionar un modelo para la migración de sistemas de organización de conocimiento al entorno de la Web Semántica, sirviendo además para

3 No debemos olvidar que la Web es un sistema de hipertexto.

4 SKOS: *Simple Knowledge Organization System*. Modelo de representación de esquemas conceptuales en la Web Semántica basado en RDF/RDFS.

la construcción de esquemas de conceptos simples y su utilización en la Web.

SKOS está ideado como un complemento a OWL, ya que proporciona un marco básico para la construcción de esquemas de conceptos pero sin la definición semántica tan estricta que exige la utilización de OWL. Supone en cierta medida una simplificación mayor de la que encontramos con el uso directo de ontologías, lo cual permite acceder a un mayor número de personas a este tipos de tecnologías para la representación del conocimiento (W3C, 2004a1).

1.3. Principales aportaciones

El actual ritmo de crecimiento de la información en la Web precisa de la aplicación de técnicas de indización automática que se apoyen en tesauros desarrollados de forma cooperativa, aprovechando sinergias positivas emanadas de esa amplia voluntad de colaboración que caracteriza a la actual Web 2.0.

En este contexto, pensamos firmemente que nuestra propuesta representa una oportunidad para poner un poco de orden en el caos actual de la indización que se lleva a cabo en la Web, realizada principalmente por los autores de los contenidos por medio de sistemas de etiquetado libre, intentando aportar un sistema que proporcione los metadatos más adecuados para la descripción de esos documentos. A partir de ahora, elementos tales como ‘description’ de la especificación Dublin Core (o de otro estándar que pudiera adoptarse) se convierten así en la piedra angular de una adecuada indización de los documentos porque resulta obvio que las técnicas de indización automática no son suficientes para manejar la información de Internet, debido a su gran volumen de información, y su naturaleza dinámica y heterogénea.

Estas características reclaman una *“simple, compatible y conveniente descripción estándar de la información para asistir y facilitar la indización automática de los recursos de Internet de una manera efectiva y eficiente”* (Zhang y Dimitroff, 2005). Parece innecesario subrayar el hecho de que cuanto más facilitemos la interacción con tesauros, más fácil resultará a los indizadores la elección del adecuado término descriptor.

Por tanto, nuestra principal aportación es la creación de un modelo conceptual para el diseño de una aplicación web, orientada a la creación y mantenimiento de tesauros y su uso a través de la indización de repositorios de recursos de información. Esta

aportación se lleva a cabo por medio de un análisis funcional completo que contempla además la aplicación de los tesauros gestionados en la indización de repositorios de información, integrando en un único procedimiento los procesos de consulta, mantenimiento y explotación. Todo lo anterior se ha formalizado con un lenguaje de modelado de aplicaciones de gran difusión como es UML.

Además del modelo anteriormente indicado, se ofrece una visión actualizada del tesoro como resultado de un proceso evolutivo en el que la intervención de las Tecnologías de la Información ha resultado decisiva en el surgimiento de un nuevo concepto de tesoro generalizado como esquema conceptual, aplicable en tareas de recuperación y organización de información en entornos muy dinámicos (como la Web) y perfectamente integrable con otras herramientas como los Topic Maps o los mapas conceptuales.

Resulta también de gran interés el análisis de las primeras iniciativas de representación de tesauros mediante RDF. De este modo se fue conformando un modelo conceptual que cristalizaría posteriormente en SKOS, del que además se aporta un análisis detallado de todos sus componentes de forma ejemplificada, ofreciendo una exposición de las ventajas que conllevaría su aplicación.

También destacamos la presentación unificada de tres disciplinas, utilizadas en el diseño web y que hasta ahora la mayoría de autores han estudiado y aplicado de forma separada y en ocasiones subordinando unas a otras: Accesibilidad, Usabilidad y Arquitectura de la Información.

Finalmente, consideramos relevante la visión basada en tres niveles (conceptual, navegacional y estructural) que tenemos de una Web en donde se apliquen los esquemas conceptuales en los procesos de recuperación de información

1.4. Hipótesis de trabajo y metodología utilizada

Nuestra intención es demostrar que la Web Semántica ofrece una alternativa a la recuperación de información con técnicas estadísticas y probabilísticas a través del uso de tesauros, siendo posible plantear un modelo conceptual de aplicación web que contemple la gestión, consulta, uso y explotación integral de dichos esquemas de forma colaborativa, todo ello utilizando SKOS como modelo para la representación de tesauros.

Para ello hemos seguido de forma combinada varias metodologías de investigación, principalmente desde el punto de vista analítico. Esta visión ha permitido la identificación y estudio de diversos conceptos, con la finalidad de realizar una abstracción que permita comprender las principales propiedades y estructura de los mismos. Así pues, ha sido posible la concreción mediante técnicas de modelado orientadas de forma parcialmente sistémica.

En definitiva, el modelo conceptual propuesto es fruto del análisis de diversas tecnologías, como las de la Web Semántica, y conceptos, como el de Tesoro, que ha permitido abstraer determinadas características, en forma de propiedades y relaciones, para posteriormente construir nuestro propio modelo, contemplando principalmente aspectos estáticos, ya que los aspectos dinámicos están más cercanos a la aplicación del mismo en soluciones concretas.

El análisis del concepto de “recuperación de información” y la caracterización de las diferencias con el de “recuperación de datos” marca el inicio de nuestro trabajo. Dicha comprensión se realiza a través de estudios de diversos autores que han caracterizado dichos conceptos, haciendo especial énfasis en la relación existente entre la representación de la información y la eficacia de los procesos de recuperación. Igualmente se destaca la incidencia de la ambigüedad del lenguaje natural, por lo que el uso de herramientas lingüísticas y esquemas conceptuales permitiría aportar soluciones a dicha problemática.

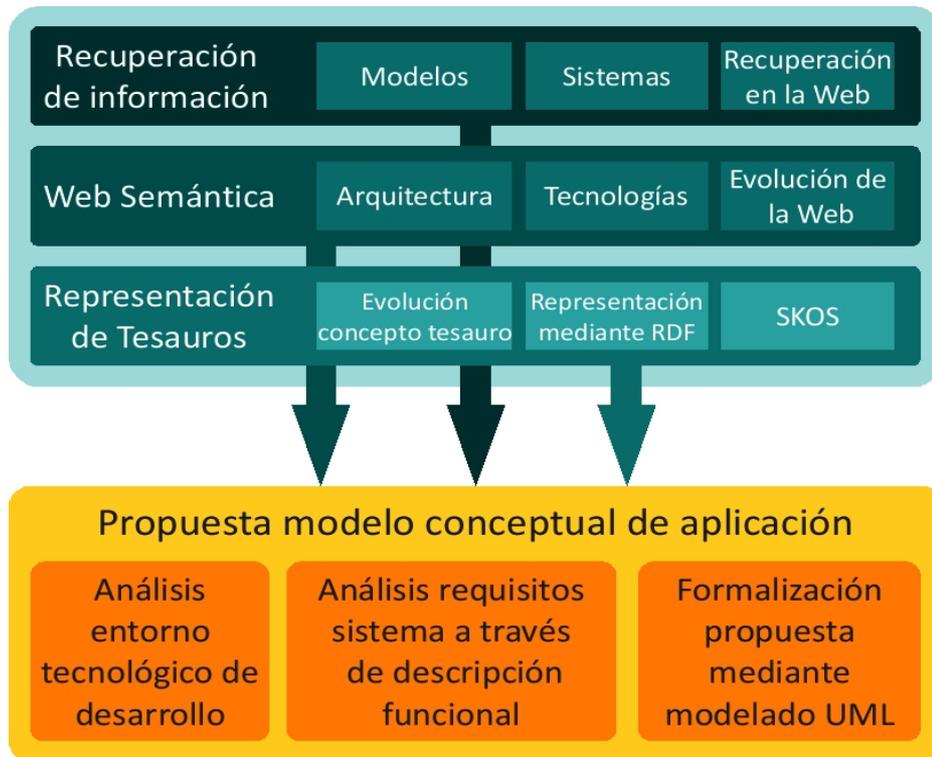


Ilustración 1: Descripción general de la metodología utilizada

Seguidamente se ha completado el análisis anterior con un estudio de los diferentes modelos de recuperación de información, recurriendo a un estudio detallado de los modelos representados en la taxonomía propuesta por Baeza-Yates (1999). A continuación se ha realizado una descripción de los principales elementos y procesos de los sistemas de recuperación de información, principalmente desde la perspectiva de la interfaz de usuario. La recuperación de información en la Web también ha sido estudiada de forma detallada, en especial las diferentes herramientas: motores de búsqueda, directorios, metabuscadores, etiquetado colaborativo y análisis de la estructura hipertextual.

Puesto que los principales problemas de recuperación en la Web derivan de las deficiencias en la representación de la información se han estudiado en profundidad la estructura y componentes tecnológicos de la propuesta orientada a solucionar este problema: la Web Semántica. Por tanto, ha resultado crucial el análisis de su arquitectura general y detallada, así como de sus tecnologías más representativas: XML, RDF y OWL. Tras haber comprobado el papel esencial que juegan los metadatos y

ontologías en el desarrollo de la Web Semántica, se ha procedido a un análisis del proceso evolutivo de la Web y en qué medida el incremento en el grado de estructuración y abstracción de los metadatos podría condicionar el desarrollo de procesos de recuperación de información en la Web más eficaces. También se ha podido comprobar que el desarrollo de la Web continúa más allá de la Web Semántica, hacia un estado denominado Web Ubicua. En esta nueva concepción las nuevas tecnologías de creación y representación de la información con metadatos y ontologías se funden con los procesos orientados a la mejorar la conectividad entre usuarios y procesos produciendo un nuevo paradigma en donde la Web constituye el núcleo de los sistemas informáticos.

Tras examinar las tecnologías que ofrece la Web Semántica hemos constatado que sus objetivos últimos no son alcanzables en su totalidad a corto plazo, primordialmente por lo complejo y exigente de los procesos de representación de información mediante ontologías. Buscando alguna solución a esta problemática hemos encontrado que existen iniciativas para desarrollar estados intermedios mediante el uso conjunto de metadatos y esquemas conceptuales. Por tanto el concepto de Tesauro resulta de especial importancia en el desarrollo de la Web Semántica y por tanto en la recuperación de información en la Web, si bien a través de una idea más general: el esquema conceptual.

Al estudiar el concepto de tesauro se ha observado un claro proceso evolutivo, desde una herramienta lingüística hacia otra en donde resultan esenciales los aspectos de organización del conocimiento. Hemos confirmado que las tecnologías de la información han sido (y continúan siendo) un elemento catalizador en el proceso evolutivo mencionado anteriormente, destacando principalmente el Hipertexto y las posibilidades de integración con otras herramientas conceptuales como los Topic Maps.

Así pues, en la Web Semántica podrían resultar de gran utilidad los tesauros y los mismos han evolucionado para adaptarse a las necesidades y posibilidades de los procesos automatizados. Hemos procedido a examinar las técnicas de representación de tesauros en ese entorno. Por ello se han revisado aquellas iniciativas basadas en RDF, puesto que este modelo supone el núcleo principal de los procesos descriptivos en la Web Semántica. Fruto de estas iniciativas es SKOS, un modelo para la representación de esquemas conceptuales que aún se encuentra en desarrollo pero

cuya estructura está consolidada y descrita claramente.

Por tanto, se ha realizado un análisis pormenorizado de los elementos de SKOS, su estructura y su relación con OWL. Esta tarea se ha llevado a cabo a través de una síntesis de los diferentes documentos que conforman la especificación de SKOS, tanto del modelo como del vocabulario y de sus extensiones, siendo constatar que la propuesta que ofrece SKOS está perfectamente adaptada a la descripción de recursos de información en la Web y su posterior recuperación.

Acto seguido hemos analizado las posibilidades de desarrollo de una aplicación que permita utilizar SKOS como modelo de representación de tesauros y cuyo ámbito de aplicación fuera la Web.

Se ha inferido que lo ideal sería el desarrollo de una aplicación que también se ejecutara en dicho entorno, porque la realidad tecnológica en este campo, desde el punto de vista de la arquitectura de sistemas (considerando la Web como cliente/servidor universal), así como los lenguajes de programación y los sistemas de almacenamiento de datos hacen viable crear una aplicación de este tipo. Asimismo se ha realizado una exposición conjunta de las disciplinas de Accesibilidad, Usabilidad y Arquitectura de la Información, puesto que sus pautas y principios deben estar contemplado en el desarrollo de sistemas de información basados en Web.

De forma previa a la creación de un sistema de este tipo, resulta necesario plantear un modelo conceptual aplicable a los procesos de diseño, desarrollo e implementación posteriores. Así, se ha desarrollado un estudio de los requisitos de un sistema como el descrito anteriormente, considerando la integración de los procesos de consulta, mantenimiento, uso y explotación a través de estructuras de trabajo en grupo y técnicas flujo de trabajo (algo contemplado en nuestra hipótesis).

Dicho estudio se ha realizado en forma de análisis de las funcionalidades del sistema. Finalmente se ha procedido a la formalización de los aspectos estáticos de la aplicación mediante el uso de diagramas de casos de uso y diagramas de clases UML. Adicionalmente, hemos ofrecido un esbozo de la dinámica del modelo mediante un ejemplo de diagrama de secuencia junto con algunas consideraciones a tener en cuenta para la implementación del sistema.

1.5. Estructura

A continuación se comentará brevemente la distribución de los contenidos de esta memoria de Tesis Doctoral. La estructura del documento se corresponde en cierta medida con el desarrollo de la metodología aplicada a lo largo de la investigación.

A lo largo del primer capítulo se expone una introducción con los antecedentes, motivaciones y principales objetivos, así como la hipótesis planteada y la metodología de trabajo seguida.

Es necesario estudiar la realidad de la recuperación de información en la Web. Dicho análisis se realiza en el capítulo segundo, junto con una revisión de los objetivos, estructura y tecnologías de la Web Semántica, así como la conexión entre procesos de recuperación más eficientes mediante el uso de metadatos, ontologías y esquemas conceptuales.

En el capítulo tercero abordamos el concepto de tesoro y su evolución en el campo de la Documentación para tareas de organización del conocimiento y recuperación de información. Dentro del mismo capítulo se realiza un análisis de propuestas para la representación de tesauros en la Web Semántica que han cristalizado en el modelo propuesto por SKOS que asimismo ha sido estudiado en detalle.

Nuestra propuesta de un modelo de aplicación para la gestión y el uso de tesauros en la Web de forma colaborativa y utilizando el modelo propuesto por SKOS, se formaliza el cuarto capítulo. Para ello se estudia el entorno tecnológico existente para el desarrollo de aplicaciones web. Esta visión se completa con los aspectos metodológicos planteados por disciplinas utilizadas en el diseño de sitios web como son la Accesibilidad, la Usabilidad y la Arquitectura de la Información. Se ha utilizado una metodología de modelado adaptada inicialmente a la representación de aspectos conceptuales, pero abierta al diseño en su vertiente más cercana a la implementación de aplicaciones.

En el quinto capítulo se recogen aquellas conclusiones obtenidas tras el estudio de los conceptos abordados durante la estructura discursiva de este trabajo, así como del análisis del modelo conceptual obtenido. En el mismo capítulo se ha incluido una reflexión final acerca de las futuras líneas de trabajo y desarrollo de esta tesis.

El último capítulo incluye la bibliografía utilizada. Para elaborar las referencias se ha

1. Introducción

utilizado el formato recomendado por la Asociación de Psicología Americana, adaptando algunos aspectos tipográficos con la finalidad de facilitar la lectura de las direcciones de aquellos recursos accesibles a través de la Web.

2

*Recuperación de
Información y Web
Semántica*

2. Recuperación de Información y Web Semántica

"El lenguaje ha creado al hombre más que el hombre al lenguaje" (Jacques Monod)

Resumen: La Web constituye una fuente de nuevas ideas para el desarrollo de técnicas de Recuperación de Información. En los últimos años, según ha aumentado la facilidad para publicar contenidos, se han tornado evidentes los problemas asociados a la Recuperación de Información en la Web. Esto ha motivado el desarrollo de nuevas técnicas que han ido más allá de la simple extracción de términos. Uno de los objetivos de la Web Semántica, es la representación y descripción del contenido semántico de recursos de información para tratar de solventar los problemas de aplicación de técnicas de Recuperación de Información en la Web.

2.1. Definición de Recuperación de Información

La expresión "recuperación de información" fue acuñada por Calvin Mooers en 1948 junto con el término "descriptor". Pese a ser empleada desde el principio de los años 50 su gran difusión a partir de la década siguiente implica una gran variedad de definiciones. En una primera aproximación subyace la idea de "recuperación de datos" en la que no existe ningún criterio complejo de selección para recuperar elementos de un almacén de datos. En la recuperación de datos no se aplican los conceptos de utilidad, relevancia o pertinencia, sino que simplemente se realiza un proceso de localización de patrones alfanuméricos en los registros o documentos con los que se trabaja. Grossman y Frieder (1998) indican que *"la recuperación de información supone encontrar documentos relevantes, no encontrar simples correspondencias a unos patrones de bits"*.

A menudo, tal como advierte Rijsbergen (1999)⁵, el sentido de esta expresión se utiliza en un sentido muy amplio y por lo tanto se utiliza para referirse a diferentes

⁵ Esta obra es la versión en línea de la segunda edición del clásico de Rijsbergen "Information Retrieval" publicada en 1979 (Rijsbergen, C.J. "Information Retrieval" London: Butterworths, 1975).

conceptos. Este mismo autor diferencia entre recuperación de datos y recuperación de la información; resumidas en la siguiente tabla.

Aspectos de análisis	Recuperación de datos	Recuperación de información
Acierto (correspondencia)	Exacta	Parcial, la mejor
Inferencia	Algebraica	Inductiva
Modelo	Determinista	Posibilista
Lenguaje de consulta	Fuertemente estructurado	Estructurado o natural
Especificación de la consulta	Precisa	Imprecisa
Error en la respuesta	Sensible	Insensible

Tabla 1: Diferencias entre recuperación de datos y recuperación de información. Fuente: Rijsbergen (1999)

Otros autores, como Pérez-Carballo y Strzalkowski (2000), inciden también en esta realidad cuando afirman que *“una tarea típica de recuperación de información es la de encontrar documentos relevantes desde un gran archivo en respuesta a una pregunta formulada por un usuario y ordenar estos documentos de acuerdo a su relevancia”*.

Es posible que esta confusión terminológica, e incluso conceptual, pueda deberse a que en realidad los sistemas y procesos de recuperación de información operan, en último término, con datos. En efecto, tal como afirma Meadow (1992) refiriéndose a la recuperación de información *“se trata de una disciplina que involucra la localización de una determinada información dentro de un almacén o base de datos”*.

Puede afirmarse que la recuperación de información implica necesariamente una recuperación de datos pero no al revés. En estos procesos no se tiene en cuenta el punto de vista del usuario. Son operaciones puramente informáticas producto de la simple comparación lógica de cadenas de bits. Desde nuestro punto de vista habría que tener en cuenta los criterios que según Blair (1990) distinguen la recuperación de datos de la recuperación de la información:

1. En la recuperación de datos las consultas se especifican mediante un lenguaje con un nivel de formalización muy alto que se traducen en una serie de resultados exactos en relación a dicha consulta. Sin embargo en recuperación de información el usuario suele especificar sus consultas en un lenguaje poco formalizado muy cercano al lenguaje natural.

2. En la recuperación de datos el éxito de la búsqueda depende directamente de la exactitud entre la consulta realizada por el usuario y el conjunto de los elementos recuperados, mientras que en la recuperación de la información este factor se asocia al grado de satisfacción del usuario en relación a la percepción personal de utilidad de la información obtenida durante el proceso de búsqueda.
3. Por lo tanto podemos afirmar que la recuperación de datos es un proceso determinista ya que el usuario espera obtener unos resultados concretos y exactos. Sin embargo la recuperación de la información se trata de un proceso probabilista ya que el usuario no puede predecir de antemano si los elementos recuperados serán de utilidad para cubrir sus necesidades de información.

Baeza-Yates (1999) afirma que las diferencias entre ambos conceptos radican en que los datos pueden representarse mediante una estructuración clara y definida mientras que la representación de la información documental no es una tarea fácil. Tal y como afirma este autor *“la representación y organización debería proveer al usuario un fácil acceso a la información en la que se encuentre interesado. Desafortunadamente, la caracterización de la necesidad informativa de un usuario no es un problema sencillo de resolver”*. Esta afirmación entronca con la definición de Salton (1983) *“la recuperación de la información tiene que ver con la representación, almacenamiento, organización y acceso a los ítem de información”*.

Otros autores no se centran en las diferencias entre recuperación de datos y recuperación de información. Un claro ejemplo es la definición de Feather y Storges (1997) que consideran la recuperación de información como *“el conjunto de actividades necesarias para hacer disponible la información a una comunidad de usuarios”*. Para Croft (1987) la recuperación de información es *“el conjunto de tareas mediante las cuales el usuario localiza y accede a los recursos de información que son pertinentes para la resolución del problema planteado. En estas tareas desempeñan un papel fundamental los lenguajes documentales, las técnicas de resumen, la descripción del objeto documental, etc.”*.

En esta misma línea se encuentra la caracterización de la recuperación de información de Tramullas Saz (1997) como aquellos *“mecanismos más adecuados para extraer, de un conjunto de documentos, aquellos que fuesen pertinentes a una necesidad informativa dada”*.

Desde nuestro punto de vista la recuperación de información no debe limitarse a un proceso en la que el usuario realiza una consulta y el sistema de recuperación de información devuelve un conjunto ordenado de elementos relevantes. Como ya indica Salton el problema de la recuperación de información tiene que ver en gran medida con la forma en la que se representa la información. Si nos ceñimos únicamente a trabajar con información documental sin tener en cuenta la estructura semántica de la misma estaremos desarrollando una tarea más cercana a la recuperación de datos que a la recuperación de información. El procesamiento de documentos textuales o hipertextuales sin ningún tipo de estructura discursiva o semántica obliga a los sistemas de recuperación de información a realizar una representación de la información documental a posteriori.

La propia ambigüedad del lenguaje dificulta la recuperación de información ya que la información solamente es legible y no entendible por el sistema cuando se enfrenta a documentos planos sin ningún tipo de estructuración. Esta dificultad es resumida de forma bastante clara por Gil Leiva (1999) en su revisión de argumentaciones en contra de la indexación automatizada, y es aplicable también al problema de la recuperación de la información: la obtención de una representación del documento para la posterior recuperación de información. Esta estrategia implica el uso de métodos estadísticos asistidos en cierta medida por herramientas lingüísticas de desambiguación, lematización y etiquetado morfológico a partir de una serie de criterios específicos derivados del propio modelo de recuperación de información aplicado.

Es factible pensar que un usuario que esté buscando información sobre “estudios de primer y segundo ciclo” encuentre más relevantes aquellos documentos en cuyos títulos o epígrafes aparezca dicha expresión. Para ello es necesario partir de una representación previa del documento que haga explícita la estructura semántica o discursiva del mismo. Dicha representación debe ser formal, estándar, flexible y abierta de modo que permita operar con diferentes tipos de documentos en cuanto al grado de estructuración y heterogeneidad de contenido.

Además de los criterios de relevancia y utilidad que pueda tener el usuario hay que contemplar en la búsqueda de información cada usuario utiliza un esquema mental individual de organización del conocimiento. El mismo usuario del ejemplo anterior puede interrogar al sistema utilizando la expresión “titulaciones de primer y segundo ciclo”, “estudios de ciclo largo” o “licenciaturas universitarias”.

Así pues el uso conjunto de esquemas conceptuales y herramientas lingüísticas ofrecerían una guía en el proceso de búsqueda al tiempo y una visión de la organización subyacente del conocimiento en el contexto del conjunto de documentos sobre el que realiza la búsqueda. El usuario no solamente podría interrogar al sistema acerca de la aparición de expresiones en el texto completo de los documentos sino que además dispondría de mecanismos interactivos de retroalimentación para expandir, refinar y orientar su búsqueda de información.

2.2. Modelos de recuperación de información

El proceso de recuperación de información se realiza en el entorno de un sistema, que utiliza un modelo que define la forma en la que se realizan las representaciones de los documentos y de las consultas. También indica como calcular el orden de relevancia de los resultados recuperados en el proceso de búsqueda, al tiempo que establece los mecanismos para guiar al usuario en el proceso de retroalimentación, para refinar o ampliar las consultas.



Ilustración 2: Taxonomía de los modelos de recuperación de información según Baeza-Yates (1999)

Baeza-Yates (1999) realiza una clasificación de los modelos de recuperación de información basándose en el modo en el que el usuario realiza este proceso. El usuario puede especificar una consulta, o ecuación de búsqueda (recuperación), o realizar una navegación u hojear (browse) más o menos detallado dentro de un documento en particular o recorriendo una red de documentos. Desde este punto de vista, el autor divide a los modelos basados en la recuperación en dos grupos: clásicos y estructurados. Los modelos clásicos de recuperación incluyen los basados en el álgebra de Boole (booleanos), los que hacen uso de la teoría del espacio vectorial y los que utilizan funciones probabilísticas.

El modelo booleano está basado en la teoría de conjuntos y el Álgebra de boole. Se puede considerar el modelo más simple y fue utilizado por los primeros sistemas de recuperación de información. Los documentos se representan por conjuntos de términos contenidos en ellos. A pesar de ser el modelo más simple es bastante complejo traducir una necesidad de información de un usuario a una expresión booleana a lo que se suma que las consultas deben expresarse a través de una función normal disjunta óptima. Puede considerarse que este modelo está más cercano a la recuperación de datos que a la recuperación de información.

El modelo probabilístico como modelo de recuperación de independencia binaria fue desarrollado por Robertson y Spark Jones (1976 y 1977). Este modelo afirma que pueden caracterizarse los documentos de una colección mediante el uso de términos de indización. Obviamente existe un subconjunto ideal de documentos que contiene únicamente los documentos relevantes a una necesidad de información para lo cual se realiza una ponderación de los términos que componen la consulta realizada por el usuario. A continuación el sistema calcula la semejanza entre cada documento de la colección y la consulta y presentando los resultados ordenados por grado de probabilidad de relevancia en relación a la consulta. Este modelo evita la comparación exacta (existencia o no de un término de la consulta en el documento) y posibilita al usuario realizar un proceso de retroalimentación valorando la relevancia de los documentos recuperados para que el sistema pueda calcular la probabilidad de que los términos utilizados en la consulta sean o no relevantes.

Tal vez, el modelo más utilizado actualmente sea el vectorial descrito por Salton y McGill (1983) para su aplicación tanto en la representación de las consultas como de los documentos mediante vectores cuyas componentes se corresponden a cada uno

de los términos presentes en el sistema. Este modelo asigna un peso determinado a cada uno de los términos que conforman los documentos. Esta misma operación se realiza con la consulta especificada con la finalidad de calcular la similitud entre documentos y consultas. Finalmente se obtiene un conjunto de documentos ordenados por relevancia.

Este modelo es muy utilizado actualmente aunque en consultas muy complejas y en corpus documentales muy amplios puede conllevar un alto número de operaciones. Según Salton y McGill un sistema de recuperación de información que utilice este modelo y que utilice n términos de indización utilizará un vector de n dimensiones donde cada componente tomará el valor dependiendo del peso con respecto a la presencia del término en el documento o en la consulta. A partir de la representación se calcula una función de comparación donde se ubicarán los documentos en un espacio de n dimensiones.

Los documentos semejantes tienden a formar agrupaciones (*clusters*) en dicho espacio vectorial. De idéntica forma se sitúa la consulta en el mismo espacio. Aquellos documentos o agrupaciones que estén más próximos a la pregunta se consideran relevantes. En realidad este modelo recupera todos los documentos del sistema pero mediante la aplicación de un umbral de relevancia solo se mostrarán al usuario aquellos que no superen cierta distancia con respecto al vector de la consulta.

Estos modelos, a su vez, presentan una serie de variaciones alternativas que mejoran el rendimiento y los resultados obtenidos. Baeza-Yates (1999) realiza la siguiente clasificación:

- Teoría de conjuntos: conjuntos difusos y booleano extendido.
- Algebraicos: Vector generalizado, indización por semántica latente y redes neuronales.
- Probabilísticos: redes de inferencia y redes de conocimiento).

Los modelos estructurados utilizan la estructura del documento junto con el contenido del mismo. Los principales modelos de este tipo son:

- Modelo de listas no solapadas. Esta propuesta de Burkowski (1992) no realiza la recuperación de información mediante búsquedas en todo el texto sino en

secciones disjuntas (no solapadas) que se corresponden con la estructura del documento.

- Modelo de nodos proximales. Desarrollados por Gonzalo Navarro y Baeza-Yates (1995) es una variación del anterior pero que además define una estructura de árbol para representar la estructura del documento.

Estos modelos están escasamente difundidos debidos fundamentalmente a que requieren una estructuración explícita de los documentos con los que trabaje el sistema. Además el lenguaje de consulta se aleja del lenguaje natural ya que el usuario debe especificar en ocasiones el nivel estructural del documento en el que se realiza la búsqueda.

Otro de los inconvenientes es la perdida del concepto de relevancia aunque algunos soluciones permiten la aplicación de algoritmos para calcular una relevancia parcial a partir de los documentos recuperados en una consulta. Sin embargo existen aproximaciones a estos modelos estructurados como la de Grabs y Shek (2002) que combinan los modelos estructurales con el modelo vectorial. Estos autores afirman que los modelos clásicos no son adecuados para la recuperación de información en conjuntos de documentos XML⁶ por su estructura jerárquica arborescente.

Respecto a los modelos basados en la navegación Baeza-Yates (1999) nos ofrece la siguiente clasificación:

- Estructura plana: Este modelo es en realidad la lectura de un único documento aislado del contexto del resto del conjunto de documentos del sistema.
- Estructura guiada: Los documentos se organizan en conjuntos de directorios jerárquicos aplicando los conceptos de clase y subclase.
- Hipertexto: La colección de documentos se organiza según una red de enlaces, que es utilizada el usuario para adquirir información de forma no secuencial.

Posiblemente el futuro de los sistemas de recuperación de información no pase por el uso exclusivo de uno u otro modelo. La alternativa más viable, tal y como proponen

⁶ XML: *eXtensible Markup Language*. Lenguaje extensible de etiquetas desarrollado por el W3C producto de una simplificación y adaptación del SGML para definir la gramática de lenguajes específicos.

Grabs y Shek (2002) es el uso conjunto de varios de los modelos anteriormente descritos para facilitar el acceso no secuencial a la información previa búsqueda en el texto completo de documentos formalmente estructurado.

2.3. Los sistemas de recuperación de información

El origen de los sistemas de recuperación de información (en adelante SRI) se remonta a los años 40 con la finalidad de asistir a los técnicos encargados de la gestión de grandes volúmenes de información documental. Salton y McGill (1983) definen los SRI como *“un conjunto de ítems de información (DOCS), un conjunto de peticiones (REQS) y algún mecanismo (SIMILAR) que determine qué ítems satisfacen las necesidades de información expresadas por el usuario en la petición”*. Sin embargo el propio Salton añade además un elemento que permite la representación a través de un proceso de indexación o categorización (*LANG*). Así pues, según Salton, el objetivo de los SRI es el tratamiento de documentos y el procesamiento de las consultas de los usuarios para que puedan acceder a la información relevante en un intervalo de tiempo adecuado.

Tal y como apunta Marchionini (2004) este esquema de funcionamiento se corresponde con la visión más clásica de los SRI en donde se realiza una comparación entre las representaciones de los documentos y las consultas. Tanto la representación como el algoritmo de comparación se basan en algún modelo de recuperación de información.

En este esquema de funcionamiento, en ocasiones, el usuario no interviene en el mantenimiento del conjunto de documentos con el que opera el SRI sino que únicamente se limita a buscar información. Sin embargo otros autores también lo consideran para aquellos usuarios que precisan de un sistema personal para el almacenamiento y la recuperación de información.

Tal es la orientación que ofrece Korfhage (1997) centrada en estos dos aspectos (almacenamiento y recuperación) como las dos vertientes de estos sistemas, *“un usuario de un sistema de información lo utiliza de dos armas posibles: para almacenar información en anticipación de una futura necesidad, y para encontrar información en respuesta una necesidad”*.

También es importante considerar las funciones que debe realizar un SRI. A este respecto Chowdhury (1999) enumera las siguientes:

1. Localización de aquellos conjuntos de documentos que son relevantes a las necesidades de información de los usuarios.
2. Análisis del contenido de los documentos con los que opera el SRI.
3. Representación de los contenidos de los documentos analizados de forma que pueda ser comparada con la consulta que pueda realizar el usuario.
4. Análisis de las consultas realizadas por el usuario y realizar la representación de las mismas de manera que puedan ser comparadas con la representación de los documentos.
5. Comparación entre la representación de la consulta del usuario para la búsqueda en la representación de los documentos de los documentos almacenados en el SRI.
6. Recuperación de aquellos documentos que resulten similares a la consulta del usuario y mostrarlos ordenados según criterios de relevancia.
7. Ejecución de mecanismos que permitan la ejecución de procesos de retroalimentación a partir de las acciones y decisiones del usuario.

Es posible adoptar un modelo diferente para el análisis de las funciones de los SRI a partir de las operaciones que realiza el usuario, la organización de los interfaces y la formas de interacción durante el proceso de búsqueda más concretamente cómo intervienen los interfaces en la organización . En este punto los interfaces de usuario desarrollan un papel determinante. Marchionini (1992) ofrece una aproximación a las funciones de un SRI a partir de la identificación de y formulación de las necesidades de información del usuario.

El proceso de búsqueda de información dependerá, en gran medida, de las fuentes disponibles, de las posibilidades del lenguaje de búsqueda, de la forma en la que el sistema represente los resultados obtenidos y de las posibilidades de interacción de las que disponga el usuario. La definición del problema, para el que se efectúa la búsqueda de información, constituye el núcleo de este proceso, siendo posible su redefinición en cada operación realizada por el usuario. Esta redefinición es vital ya que se extrae información de la misma con el fin de seleccionar las fuentes sobre las que se efectúa la búsqueda, para comparar los resultados obtenidos y durante la

elaboración de la consulta según las posibilidades del lenguaje de interrogación.

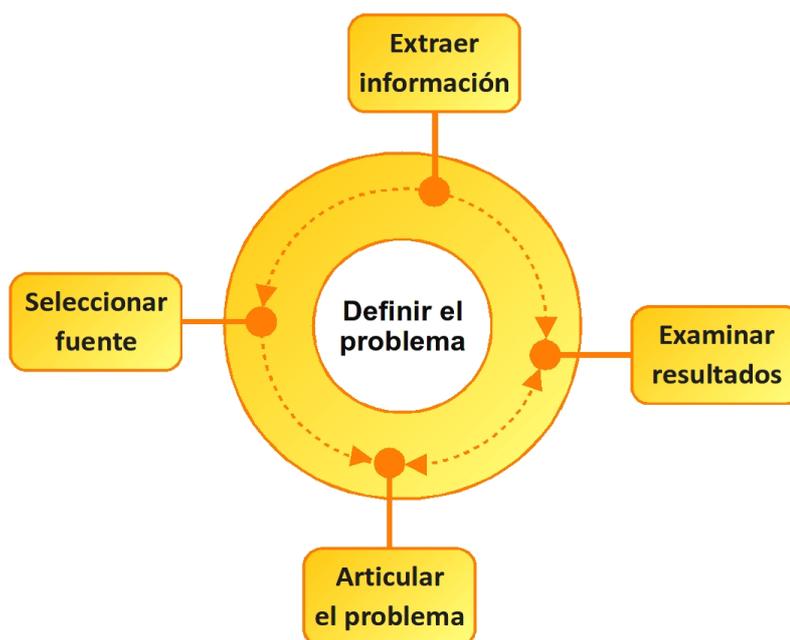


Ilustración 3: Diagrama de operaciones realizadas por el usuario durante la búsqueda de información.

Adaptado de Machionini (1992)

Aproximarse a las funciones de un SRI a partir de los interfaces de usuario constituye el núcleo fundamental de uno de los trabajos de Norlin (1992) en el que asocia estos interfaces con el modo en el que el usuario accede a la información en el ámbito de este tipo de sistemas. Marchionini (1992) insiste en que un SRI no debe ofrecer un único mecanismo de búsqueda y acceso a la información, sino que debe combinar diferentes herramientas, lenguajes y sistemas de navegación. Así pues, podemos afirmar, que la amigabilidad de los interfaces de usuario es fundamental para la estructuración y la jerarquización de las funciones de un SRI. Para García Marco (1995a) una interfaz amigable debe combinar adecuadamente distintos tipos de códigos comunicativos y estructurar las funciones en niveles jerárquicos de forma que propicie la inferencia metafórica por parte del usuario.

Hemos de tener en cuenta que actualmente los SRI operan en el ámbito de entornos de red. Múltiples usuarios publican y mantienen documentos al tiempo que realizan búsquedas de información. Por lo tanto para Engel (1997) los SRI en red deben ser adaptables a las necesidades de los usuarios y este hecho ha de reflejarse en la interfaz.

No hay que olvidar que uno de los grandes problemas a los que se enfrentan los usuarios es la gran cantidad de información con la que opera un SRI y que debe ser visualizada por el usuario. Por lo tanto, nos encontramos ante un nuevo concepto en la relación hombre máquina en donde, además de la funcionalidad de un sistema, también es imprescindible tener en cuenta que el usuario visualizará enormes cantidades de información muy heterogénea (Rao, 1996).

Precisamente la recuperación y manejo de información hipermedia y la integración de un sistema de gestión de información documental (que incluye un subsistema de recuperación de información) en el entorno ofimático y de trabajo en grupo del usuario centran los trabajos de Pastor y Saorín (1995 y 1998). El SRI no debe desarrollarse sin tener en cuenta que funciona a partir de una interfaz más general que ofrece el sistema operativo informático sobre el que se ejecuta, al tiempo que debe poder relacionarse con otras aplicaciones (procesadores de texto, sistemas personales de bases de datos, hojas de cálculo). Asensi y Pastor (1998) estructuran las funciones de un SRI a partir de cómo son ofrecidas al usuario en la interfaz de usuario:

1. Arquitectura de acceso e integración de fuentes de información.
2. Funciones generales de la interfaz de usuario.
3. Adaptación al usuario.
4. Gestión de contenidos para integración de diversas fuentes de información (bases de datos, páginas web, repositorio personal de documentos, intranet.).
5. Esquemas de búsqueda mediante el lenguaje natural, operadores de búsqueda, uso de tesauros y redes conceptuales, enlaces de hipertexto, etc.
6. Recuperación consulta de la información.
7. Funciones de ayuda y asistencia al usuario.
8. Integración con el entorno del usuario para la generación de fuentes de información propias.

La evolución de los SRI no solamente se refleja en los cambios que ha sufrido el modelo funcional de estos sistemas. También puede apreciarse en el ámbito de los

contenidos que forman parte del conjunto de documentos que mantiene o sobre los que se realiza la búsqueda de información. Desde los primeros SRI informáticos en donde se trabajaba sobre conjuntos cerrados de documentos especializados hasta los actuales SRI, que en realidad son Sistemas de Gestión de Información, se han incorporado nuevas funciones, esquemas de acceso a la información y tipos de contenidos gestionados. Los SRI han evolucionado hacia el concepto de sistemas de recuperación y almacenamiento de información motivado por la necesidad de integrar contenidos personales del propio usuario almacenados de forma local o remota, información existente en la Web, repositorios corporativos de documentos y bases de datos.

2.4. La Recuperación de Información en la Web

La Web se ha convertido en la mayor fuente de información, conformando una auténtica esfera de datos planetaria en donde se publica y consulta información en cualquier momento y prácticamente en cualquier lugar. Hace ya tiempo que autores como Lawrence y Giles (1998) o Kobayashi y Takeda (2000) se han percatado de que la Web ha pasado a ser una red en donde más que encontrar información el usuario puede verse atrapada en ella. Este hecho se debe fundamentalmente a que la Web en un principio no se ideó como un medio para almacenar información, sino como un entorno de comunicación distribuida de información (Pastor Sánchez, 1997).

Los esquemas de acceso a la información Web basados en la hipertexto son relativamente eficaces para la consulta de un conjunto limitado de documentos. El problema aparece cuando se desea buscar información en la totalidad de un sitio Web concreto o en toda la Web, debido al gran número de páginas web existentes y la heterogeneidad de los documentos (en cuanto a su formato, su estructura o la calidad de sus contenidos). Se trata, en definitiva, de un problema asociado al uso (o no uso) de herramientas adecuadas que permitan describir de un modo más exacto el contenido semántico de los recursos de información.

Según Baeza-Yates (1999) existen tres formas de buscar información en la Web:

- Motores de búsqueda (*search engine*).
- Búsqueda de información en un directorio (*web directory*).
- Aprovechar la estructura hipertextual de la Web con sistemas que recorran

redes de hiperenlaces (*hiperlinks*) para localizar la información.

En la mayoría de las ocasiones el usuario maneja servicios que combinan varias de estas estrategias de búsqueda de información en la Web. Esto puede motivar que el usuario no sepa distinguir claramente el método de búsqueda información, lo cual puede incrementar la desorientación del usuario al intentar decidir cual es la herramienta que debe utilizar.

2.4.1. Motores de búsqueda y directorios web

Los motores de búsqueda son herramientas que no operan directamente sobre la Web, sino que operan con bases de datos de páginas web recopiladas de forma automática. Estos sistemas ejecutan agentes⁷ cuya misión es la de recorrer la estructura de enlaces a partir de una lista inicial de páginas web (por regla general las páginas principales de los servidores).

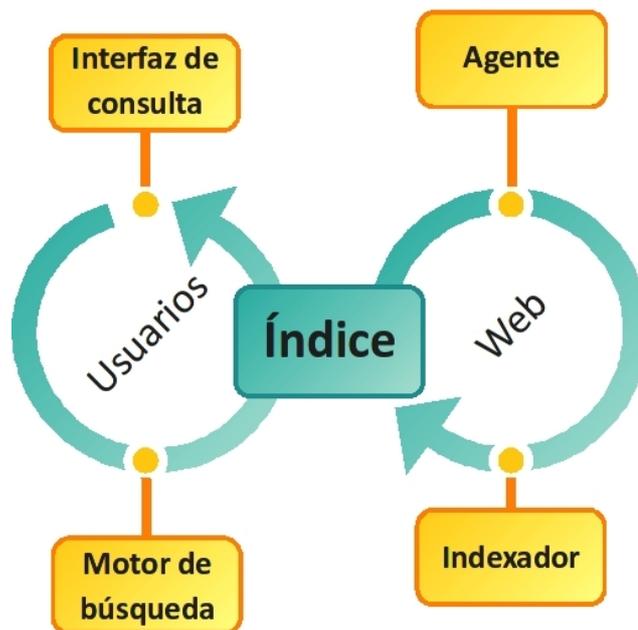


Ilustración 4: Funcionamiento de un motor de búsqueda para la Web. Adaptación de Baeza-Yates y Ribeiro-Nieto (1999)

Además de recopilar direcciones Web, estos agentes analizan y extraen cierta información y construyen un índice a partir del contenido de las páginas que visitan.

⁷ Estos agentes o aplicaciones se denominan "Crawlers".

De forma periódica, los agentes de búsqueda vuelven a visitar las páginas para actualizar la base de datos. El usuario accede a una interfaz de consulta para especificar los términos de su búsqueda y el motor de consulta devuelve los documentos relevantes.

Algunos ejemplos de motores de búsqueda son: Google, Altavista, Yahoo!, Alltheweb o Live.com, entre otros. En un principio, los motores de búsqueda realizan una indización del contenido textual de los documentos. Para ello se analizan los hipervínculos, el contenido del cuerpo de los documentos así como las metaetiquetas que de las páginas web. Actualmente también se efectúa la indización de elementos multimedia tales como imágenes, videos o sonidos.

Los directorios web son sistemas que están gestionados directamente por operadores humanos que gestionan bases de datos con descripciones de páginas web. Estas bases de datos incluyen el título, autor, entidad, palabras clave, descripciones, fecha de la introducción en la base de datos. Posteriormente se realiza una clasificación mediante un sistema de categorías temáticas donde predomina la estructura jerárquica. El usuario de un directorio web realiza una navegación entre las categorías temáticas, hasta que alcanza un nivel de concreción en el directorio con respecto al ámbito de la información que está buscando. En cada categoría o subcategoría el usuario puede seguir profundizando en la estructura jerárquica de categorías o abrir alguna de las páginas referenciadas en la categoría que esté consultando en ese momento. Los directorios también permiten realizar búsquedas terminológicas en base de datos. Ejemplos de estos directorios son Yahoo! Directory, Hispavista o Starting Point Directory, JoeAnt, DMOZ o BUBL.

Los directorios web fueron los primeros en surgir para ayudar al usuario a encontrar información en la Web. La clasificación por categorías temáticas resultaba sencilla, intuitiva y bastante clara. Pero comenzaron a tener limitaciones debido al enorme crecimiento del número de páginas web existentes en Internet. En ese momento, la dificultad no sólo se centraba en el registro y descripción de direcciones web, sino en la complejidad de las operaciones para crear y jerarquizar las categorías temáticas. Cuando una categoría contiene un gran número de recursos se divide en varias subcategorías para facilitar la búsqueda. Conforme aumenta el número de niveles jerárquicos también se incrementa el volumen de decisiones que ha de tomar el usuario para orientar su búsqueda hacia una u otra categoría, al tiempo que se

incrementan las posibilidades de errores de selección y desorientación por parte del usuario.

Los motores de búsqueda surgen para ofrecer una alternativa a esta complejidad creciente en la estructura de los directorios web. No es necesario realizar sucesivas elecciones de categoría temática en la búsqueda de información, simplemente es necesario plantear la consulta a través de un formulario en forma de términos o palabras clave. El usuario refina la búsqueda evaluando la relevancia de los resultados a partir de la utilidad de los documentos recuperados. Es interesante la observación que hace Aguilar González (2002) según la cual las claves de los motores de búsqueda son los procesos de recopilación e indización de páginas y el algoritmo implementado para ordenar los documentos según un criterio de relevancia. Para Delgado Domínguez (1998) las principales diferencias entre buscadores y directorios web pueden resumirse en la siguiente tabla.

Tipo de servicio	Descubrimiento de los recursos	Representación del contenido	Representación de la consulta	Presentación de los resultados
Directorio	Lo realizan personas	Clasificación manual	Implícita (navegación por categorías)	Páginas creadas antes de la consulta. Poco exhaustivos, muy precisos
Motores de búsqueda	Principalmente de forma automática por medio de robots	Indización automatizada	Explícita (palabras clave, operadores, etc.)	Páginas creadas dinámicamente en cada consulta. Muy exhaustivos, poco precisos.

Tabla 2: Diferencias entre directorios y motores de búsqueda web. Fuente: Delgado Domínguez (1998)

2.4.2. Búsqueda mediante el análisis de la estructura hipertextual

La tercera forma de recuperación de información en la Web enumerada por Baeza-Yates (1999) es la búsqueda analizando los enlaces que incluyen las páginas web, así como los lenguajes de consulta en la web y la búsqueda dinámica. Los lenguajes de consulta a la web (*web query languages*) permiten localizar páginas web que no contengan ninguna imagen y que sean enlazadas como mínimo por diez páginas web. Generalmente estos lenguajes de consulta utilizan una representación basada en nodos (páginas web), enlaces y el contenido estructural de los nodos. Hay que tener en cuenta que la Web alberga documentos muy heterogéneos desde el punto de vista de su estructuración.

Una de las ideas derivadas de este tipo de sistemas es la posibilidad de realizar búsquedas dinámicas siguiendo los enlaces de hipertexto de aquellas páginas recuperadas en una búsqueda de información u obtenidas por el motor de búsqueda durante el proceso de exploración e indización de la Web. Brin y Page (1998) y posteriormente Hu (2001) indican, acertadamente, que los enlaces entre páginas web también pueden constituir fuentes de información.

Esta información es explotada por buscadores como Google que no solo se limitan a extraer términos de las páginas que indiza sino que examinan la información de los enlaces hacia los que apunta una página como criterio para establecer la relevancia de una página obtenida como resultado de una búsqueda.

2.4.3. Metabuscadores

Dentro de los SRI para la Web, junto con los motores de búsqueda y los directorios Chang (2001) también habla de los metabuscadores y de las técnicas de filtrado de información. Los metabuscadores pueden utilizarse con una aplicación instalada en el puesto de trabajo del usuario o accediendo a un formulario de búsqueda en la Web y simplifican la búsqueda de información en varios motores de búsqueda.

Estos sistemas son en realidad registros de esquemas de consulta en diferentes motores de forma que el usuario realiza una única consulta y el sistema se encarga de adaptarla a los lenguajes de cada motor de búsqueda registrado, enviarles dicha consulta y obtener los resultados.

Según Aguilar González (2002) los metabuscadores pueden clasificarse a su vez en multibuscadores (cuando los resultados se organizan únicamente según el buscador que devuelve los resultados) o metabuscadores propiamente dichos (que eliminan las referencias duplicadas y vuelve a realizar cálculos de relevancia para obtener un único conjunto de resultados).

Es interesante resaltar la reflexión de Baeza-Yates (1999) según la cual *“la principal ventaja de los metabuscadores es su capacidad de combinar los resultados de muchas fuentes y el hecho de que el usuario pueda acceder a varias fuentes de forma simultánea a través de una simple interfaz de usuario”*.

Por su parte, las técnicas de filtrado de información constituyen un apoyo a los

motores de búsqueda. De esta manera se evalúa si un documento es o no relevante a priori evitando su incorporación al índice en caso de no serlo. Esto permite una mayor velocidad en la recuperación de información ya que la base de datos con la que opera el SRI disminuye su volumen.

2.4.4. Etiquetado colaborativo

Otra técnica de recuperación de información se basa en el etiquetado colaborativo con metadatos compartidos. Estos sistemas permiten que los usuarios puedan realizar una indización asignando etiquetas o palabras clave (*tags*) a direcciones de páginas web. La particularidad de este método estriba en que las etiquetas son compartidas por la comunidad de usuarios del servicio. La técnica se denomina etiquetado social (*social tagging*) o folksonomias (*folksonomies*)⁸. Algunos ejemplos de este método de trabajo lo constituyen los portales *Del.icio.us* o *Flickr*⁹. De esta forma es posible obtener metadatos e información de interés para el usuario a través de la evaluación del contenido del documento estableciendo la cercanía de dicho contenido al concepto expresado por la etiqueta asignada. Constituyen además una herramienta para la organización personal de la información permitiendo al usuario gestionar su propio repertorio de páginas web al tiempo que comparte con los usuarios tanto direcciones como etiquetas.

Algunos autores (Mathes, 2004 y Moreira González, 2006) indican que existen una serie de limitaciones inherentes a estos sistemas, tales como la ambigüedad del lenguaje empleado en la elaboración de etiquetas, la subjetividad en la descripción de los recursos y la necesidad de aplicar herramientas lingüísticas como lematizadores y gestores de sinónimos. Pese a lo anterior, constituyen un importante punto de partida porque en cierto modo cambian el paradigma actual de la mayoría de los SRI, cuya indización de páginas Web se basa en el empleo de robots o la asignación de categorías jerárquicas por un equipo de trabajo cerrado. En cierta medida estos servicios intentan suplir las carencias de representación de los aspectos semánticos de los documentos, mediante el uso de la “inteligencia colaborativa” de los usuarios que añaden recursos y etiquetas al sistema.

8 La acuñación del término *folksomony* se atribuye a Thomas Van der Val, director de InfoCloud Solutions, especialista y consultor en organización de información en la Web.

9 Estos servicios permiten a los usuarios referenciar páginas web y asignarles etiquetas (*tags*) que describen su contenido y que con compartidas por la totalidad de usuarios del sistema.

2.5. La Web Semántica

El concepto de Web Semántica constituye en estos momentos un proyecto de futuro para la organización de la información en la Web. Su propósito fundamental es la reutilización y comunicación de información no solamente entre los usuarios (hombre) y las aplicaciones informáticas (máquina), sino también para el intercambio de dicha información entre distintas aplicaciones informáticas, e incluso para la elaboración o re-elaboración de información.

Como ya veremos a lo largo de este trabajo, la Web Semántica es, en realidad, la génesis de un nuevo lenguaje o lenguajes para que las aplicaciones informáticas puedan representar, compartir, manipular, elaborar o inferir información. Así, con la Web Semántica, la información no solamente será legible para las máquinas sino que también podrá ser inteligible para éstas. Se vislumbra así una nueva generación de aplicaciones para Internet como buscadores Web más eficaces, asistentes y agentes que asistan al usuario a buscar información o usar servicios, sistemas de edición y generación de contenidos basados en procesos de identificación, estructuración e interrelación de la información a nivel semántico o la automatización de procesos de información, algo inimaginable con la estructura y contenidos de la Web actual.

2.5.1. Definición y objetivos de la Web Semántica

Examinemos a continuación el significado del concepto de Web Semántica partiendo del análisis de algunas definiciones realizadas por diferentes autores. Berners-Lee, Hendler y Lassila (2001) ofrecen la siguiente definición: *“La Web Semántica es una extensión de la actual Web en la que a la información disponible se le otorga un significado bien definido que permita a los ordenadores y las personas trabajar en cooperación. Está basada en la idea de proporcionar en la Web datos definidos y enlazados, permitiendo que aplicaciones heterogéneas localicen, integren, razonen y reutilicen la información presente en la web”* A nuestro entender las bases de esta definición se asientan en tres ideas. En primer lugar se habla de la Web Semántica como una *“extensión de la actual Web”*; esto implica un desarrollo evolutivo de herramientas, tanto conceptuales como informáticas, que permitan la coexistencia de la Web actual mientras que se alcanzan los objetivos de la Web Semántica. La segunda idea se refiere a la necesidad de representar la información con el fin de desarrollar mecanismos para que la comunicación entre hombre y máquina sea efectiva y eficaz.

Por último la anterior definición introduce la posibilidad de que las aplicaciones informáticas sean capaces de manipular e incluso elaborar información con objetivos concretos a los problemas que se les planteen (*“descubran, integren, razonen, utilicen...”*).

Similar definición es la que ofrece Berners-Lee junto con Miller (2002): *“La Web Semántica es una extensión de la actual Web en la que a la información disponible se le otorga un significado bien definido que permita a los ordenadores y las personas trabajar en cooperación. La W3C Semantic Web Activity, en colaboración con un gran número de investigadores y socios industriales, se encarga de la definición de estándares y tecnologías que permitan a los datos de la Web ser definidos y enlazados de forma que puedan ser usados para una localización más eficaz, automatización, integración y reutilización entre aplicaciones”*. Los autores ya no abordan únicamente los objetivos de la Web Semántica sino que también exponen cómo el W3C¹⁰ está coordinando estos objetivos.

Una definición algo más concreta y siguiendo la línea de la última parte de la definición anterior, nos la ofrece el W3C en la página web oficial de la Web Semántica¹¹: *“La Web Semántica es la representación de datos en la Web. Es un esfuerzo colaborativo liderado por W3C con la participación de un gran número de investigadores y socios industriales. Se basa en el uso de RDF¹²), que integra una gran variedad de aplicaciones mediante el uso de XML para la sintaxis y el uso de URIs para su identificación”*. Teniendo en cuenta los trabajos de coordinación del W3C es normal que en esta definición se incluyan dos tecnologías fundamentales asociadas al desarrollo de la Web Semántica: la especificación RDF y el lenguaje XML. Además también utiliza la expresión URI¹³ para identificar de forma universal y expansible un espacio de nombres de recursos de información. De esta definición también se desprende la existencia de una amplia colaboración a nivel científico y de fabricantes de software para la

10 W3C: *World Wide Web Consortium*. Organización internacional creada en 1994 cuyo objetivo es el desarrollo de estándares para la Web. Está dirigida por Tim Berners-Lee. Tiene su sede en el Instituto Tecnológico de Massachussets. La oficina española se creó en 2003.

11 W3C. W3C Semantic Web Activity. [En línea] <<http://www.w3.org/2001/sw/>> [Consulta: 26 de julio de 2008]

12 RDF: *Resource Description Framework*. Modelo que ofrece una especificación para la descripción metadatos en la Web, desarrollado por el W3C. Organiza la información en forma de tripletas sujeto-predicado-objeto y permite su serialización con XML.

13 URI: *Uniform Resource Identifier*. Cadena de texto que identifica unívocamente cualquier recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia ...) accesible en una red.

adopción y uso de especificaciones comunes referentes a la Web Semántica, algo que también está presente en la definición de Berners-Lee y Miller (2002).

Euzenat (2002) afirma que: *“La idea de la Web Semántica (un término acuñado por Tim Berners-Lee, el creador de la Web) implica anotar documentos con “marcas semánticas”, esto es, marcas que no son interpretadas para su visualización sino como la expresión del contenido del documento. Esto es descrito a menudo como “una Web para las máquinas” en oposición a una Web para ser leída por el hombre”*

Codina (2003) propone la siguiente definición: *“La Web Semántica es un conjunto de iniciativas, tecnológicas en su mayor parte, destinadas a crear una futura World Wide Web en la cual los ordenadores puedan procesar la información, esto es, representarla, encontrarla, gestionarla, como si los ordenadores poseyeran inteligencia”*. Es evidente que esta definición se centra en las posibilidades de automatización de procesos de gestión de información a partir de un desarrollo tecnológico concreto, cuyo objetivo es la creación de una Web donde ciertas aplicaciones o agentes informáticos puedan realizar tareas de manera más o menos autónoma. No obstante el autor ignora el papel fundamental de la Web Semántica como medio de comunicación entre hombres y máquinas o entre las propias máquinas limitándose a enumerar una serie de procesos en los que interviene la Web Semántica.

Mucho más interesante es el trabajo que nos ofrecen Halpin y Thompson (2006) donde se ofrece una breve indicación que, aunque no podría calificarse de definición, si aporta una visión diferente del resto de definiciones anteriormente citadas argumentando que *“la Web Semántica es la solución al problema de la integración de datos”*. Esta afirmación nos ofrece una visión de la Web Semántica que no termina con el desarrollo de especificaciones basadas en XML y RDF para el intercambio de datos de una forma normalizada. Los autores sugieren la integración de datos para su uso sin necesidad de realizar conversiones o traducciones sirviéndose directamente de XML de forma conjunta con procedimientos de mapeado¹⁴ con RDF. Así pues, esta orientación permite una consecución gradual de los objetivos de la Web Semántica ya que no exige la descripción o codificación de los documentos en RDF (donde la descripción semántica de los documentos se realiza de forma explícita) sino que permite un paso intermedio entre el uso de XML utilizado para el intercambio de datos y RDF.

14 Los términos “mapeado” o “mapear” provienen de la expresión anglosajona “mapping” referida a la relación de correspondencia que se establece entre dos elementos de conjuntos distintos.

Hablar de Web Semántica no supone hablar de un objetivo concreto, sino que implica la puesta en práctica de una nueva orientación en el desarrollo de la Web actual. La Web no es fruto de una planificación en donde se parte del análisis de una situación determinada para obtener una serie de resultados concretos y específicos; su desarrollo ha venido marcado por las necesidades de sus usuarios a cualquier nivel (usuarios que consultan información, gestores de información que editan páginas web y mantienen bases de datos, informáticos que desarrollan aplicaciones, etc.).

Al referirse a la Web Semántica hay que contemplar el aspecto evolutivo de la Web convencional. Si algo la ha caracterizado desde sus orígenes ha sido la capacidad de adaptación a partir de unas premisas muy sencillas que se han mantenido desde su creación. Este trabajo se enmarca dentro de lo que entendemos por Web Semántica: un conjunto de iniciativas para crear nuevas aplicaciones y servicios en Internet a partir del uso de especificaciones basadas en XML y RDF que permitan la automatización o asistencia en los procesos de gestión, intercambio y elaboración de información, tanto para el hombre como para las máquinas (Méndez, 1999).

En una Web con un carácter tan participativo como la actual los clásicos motores de búsqueda encuentran dificultades por la ausencia de mecanismos para controlar la calidad, originalidad y utilidad de los contenidos. La proliferación de técnicas para incrementar de forma artificial el posicionamiento en los buscadores web constituyen, en muchas ocasiones, el origen de este problema. La Web Semántica podría resolver este problema formalizando la representación del contenido de los documentos.

Por tanto, los objetivos de la Web Semántica pueden resumirse en los siguientes puntos:

- Uso de metadatos para describir el contenido semántico de los documentos.
- Reutilización, localización e integración de recursos usando datos estructurados mediante especificaciones con la finalidad de obtener información que posteriormente pueda ser tratada tras aplicar un conjunto de reglas y procesos para la obtención de conocimiento.
- Intercambio de información entre usuarios y máquinas en cualquiera de sus posibles combinaciones, es decir, usuario-usuario, usuario-máquina y máquina-máquina.
- Separación entre estructura, contenido, relaciones y formato de visualización de la información.

- Desarrollo de herramientas de inferencia de información para crear aplicaciones y agentes “inteligentes”.

En definitiva, la Web Semántica persigue la creación de un entorno compartido para el intercambio de información estructurada (existente en forma de recursos web) cuyos contenidos a nivel semántico se representan de forma explícita de manera relativamente “inteligible” para las máquinas. La práctica totalidad de autores y trabajos sobre la Web Semántica coinciden en las nuevas posibilidades de comunicación que ofrece entre hombres y máquinas.

Es evidente que el uso de especificaciones normalizadas para la representación e intercambio de información amplía los horizontes en los procesos de reutilización e integración de la información. Por lo tanto, se incrementa la capacidad de interoperabilidad entre sistemas informáticos con una intervención mínima de operadores humanos en los procesos inteligentes de flujo de información.

2.5.2. La Arquitectura de la Web Semántica

La Web Semántica no es una idea o un concepto utópico. Berners-Lee ya ofreció una arquitectura básica que se ha ido desarrollando a lo largo de los últimos años. Esta evolución se ha hecho realidad con la adopción y aplicación gradual de una serie de tecnologías desarrolladas partir de lenguajes de definición y estructuración de datos y reglas lógicas como XML, RDF, OWL¹⁵, etc.

A continuación, se ofrece una descripción de la arquitectura y componentes de la Web Semántica desde un punto de vista global para analizar en más detalle los diferentes lenguajes que se han apuntado anteriormente y que han surgido en torno a la Web Semántica y sus desarrollos prácticos.

La Arquitectura Global de la Web Semántica se fundamenta en tres capas:

1. Capa de metadatos: Se describen los conceptos en forma de recursos web utilizando especificaciones comunes para obtener representaciones explícitas del contenido semántico de dichos recursos.
2. Capa de Esquema: En esta capa se realiza la descripción de las relaciones

¹⁵ OWL: *Ontology Web Language*. Lenguaje para la especificación de reglas lógicas y restricciones semánticas de los esquemas conceptuales especificados con RDF/RDFS.

asociativas o jerárquicas entre los distintos conceptos descritos en la Capa de metadatos.

3. Capa Lógica: En este nivel se ejecutan las inferencias por medio de la aplicación de reglas formalmente definidas que dan forma a ontologías a partir de la estructuración de los metadatos que se ha realizado en la Capa de Esquema.

Para la evolución de la Web Semántica son fundamentales las constantes sinergias entre cada una de las capas, producto del desarrollo de la aplicación y evaluación de diferentes tecnologías. A pesar de esto, el planteamiento y funcionamiento extensible de la Web Semántica permite que los cambios en un elemento concreto de una capa no afecten de forma inmediata al resto de capas.

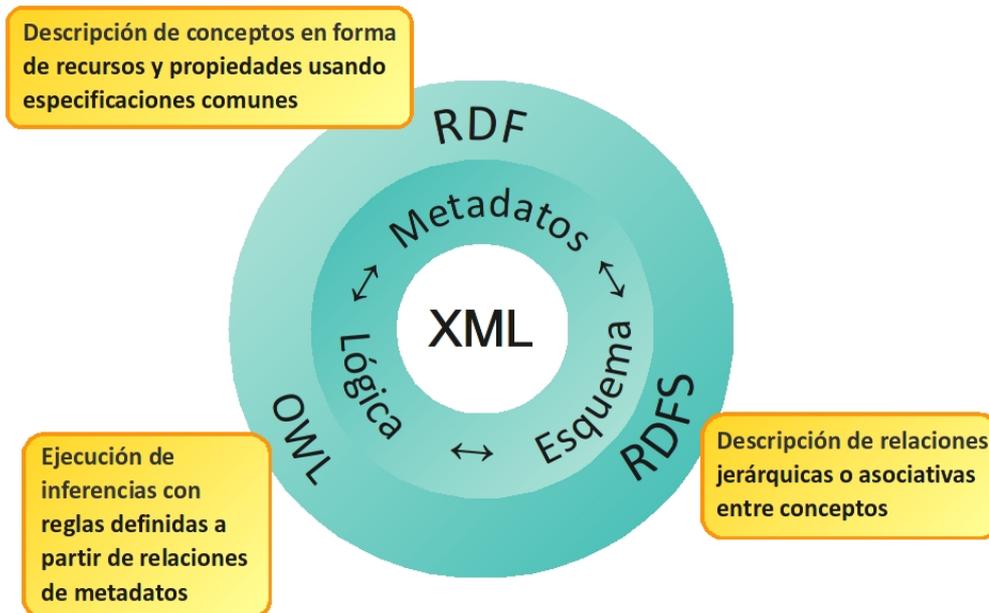


Ilustración 5: Capas de la Web Semántica.

Esta Arquitectura conlleva una serie de niveles con un grado creciente de estructuración y abstracción de la información. Uno de los aspectos más relevantes del modelo es la separación entre la estructuración de información, la creación de contenidos, la creación de relaciones entre los mismos desde los puntos de vista lógico y conceptual y el formato de visualización de la información.

Con el uso de metadatos es posible caracterizar el contenido de los elementos

informativos realizando una descripción analítica de aspectos descriptivos y de contenido. El uso de especificaciones de documentos permite separar la definición de la estructura informativa de estos elementos y su contenido. RDF juega un papel fundamental en esta capa ya que constituye un modelo potente y eficaz para la descripción de los metadatos asociados el recurso.

En la capa de esquema se definen las estructuras que permiten agrupar los diferentes elementos descritos gracias a RDF. La definición de un esquema permite estructurar los recursos aplicando relaciones de clases y subclases entre ellos. La tecnología desarrollada por el W3C para realizar esta tarea es RDFS¹⁶ (*RDF Schema*).

En la Capa Lógica se describen las reglas de inferencia y restricciones aplicables a las estructuras definidas en la Capa de Esquema. Es en esta capa donde toma forma el concepto de Ontología Web. Tras describir y estructurar correctamente los elementos de un determinado dominio mediante un esquema conceptual se definen toda una serie de reglas semánticas para complementar dichos esquemas. OWL es el lenguaje para la representación de ontologías que complementa a RDF/RDFS en el desarrollo de la Web Semántica.

Las diferentes capas de esta arquitectura se integran con la aplicación de XML que dota a la Web Semántica de un mecanismo para el desarrollo de vocabularios de lenguajes orientados al intercambio de datos para su procesamiento automático a partir de una sintaxis compartida. Analizando en mayor detalle la Arquitectura de la Web Semántica se identifican una serie de procesos:

- Localización y codificación de datos: Para referirse a cualquier recurso disponible en red es necesario un literal que permita identificarlo de forma unívoca. En la Web Semántica los recursos son referidos con una cadena denominada URI con independencia de si se encuentran en red o no. Se trata de un concepto más amplio que el de URL que solamente se refiere a la localización de dicho recurso en términos de un protocolo y una dirección de red. Esta diferencia entre URI y URL permite identificar los recursos al tiempo que se realiza una separación de manera que un recurso pueda ser accesible por diferentes aplicaciones de red.

¹⁶ RDFS: *Resource Description Framework Schema*, permite la estructuración de recursos definidos con RDF en clases y subclases al tiempo que permite la definición de dominios, propiedades y rangos de recursos RDF.

- Estructuración de datos con un vocabulario definido: En la Web Semántica los datos se estructuran con XML. Un esquema XML separa los datos y la definición de la estructura. Los *namespaces*¹⁷ definen los ámbitos de aplicación de los distintos identificadores para referirse a los recursos y a su definición.
- A continuación es necesario realizar una definición semántica de los datos en forma de recursos y las relaciones existentes entre los mismos. Se utiliza RDF para caracterizar los datos en forma de recursos con sus correspondientes atributos y RDFS para describir las relaciones que se establecen entre los mismos a nivel individual o entre clases de recursos. Posteriormente se definirán ontologías que permitan la descripción y representación formal del dominio de información sobre el que se pretende operar. Para definir estas ontologías se aplican lenguajes como DAML+OIL¹⁸ aunque la mayoría de las aplicaciones y sistemas que hacen uso de este lenguaje están migrándose a OWL. En realidad con OWL extiende RDFS para expresar relaciones y restricciones complejas entre diferentes clases de recursos.
- Una vez descritos los datos en forma de recursos, clases de recursos, relaciones y propiedades adecuadas, pueden definirse reglas lógicas para describir restricciones semánticas e inferencias de datos para obtener información que en un principio no se había descrito de forma explícita en el ámbito del dominio de información.
- Finalmente es necesario establecer mecanismos para evaluar el nivel de confianza y veracidad de los recursos de información. Para ello se aplica en todo el proceso tecnologías de Firma digital y Encriptación de datos para verificar la autenticidad y seguridad de los datos manejados y obtenidos.

Estos procesos pueden verse representados en la siguiente figura en donde se representa en detalle la arquitectura de la Web Semántica.

17 La existencia de elementos con idéntico nombre pero de esquemas XML diferentes no supone ningún problema ya que un namespace permite identificar un contexto específico de significado.

18 DAML+OIL, un lenguaje producto de la unificación de los lenguajes DAML (*DARPA's Agent Markup Language*) y OIL (*Ontology Inference Layer*) utilizado para la definición formal de ontologías para la Web. Se considera el antecedente de OWL.

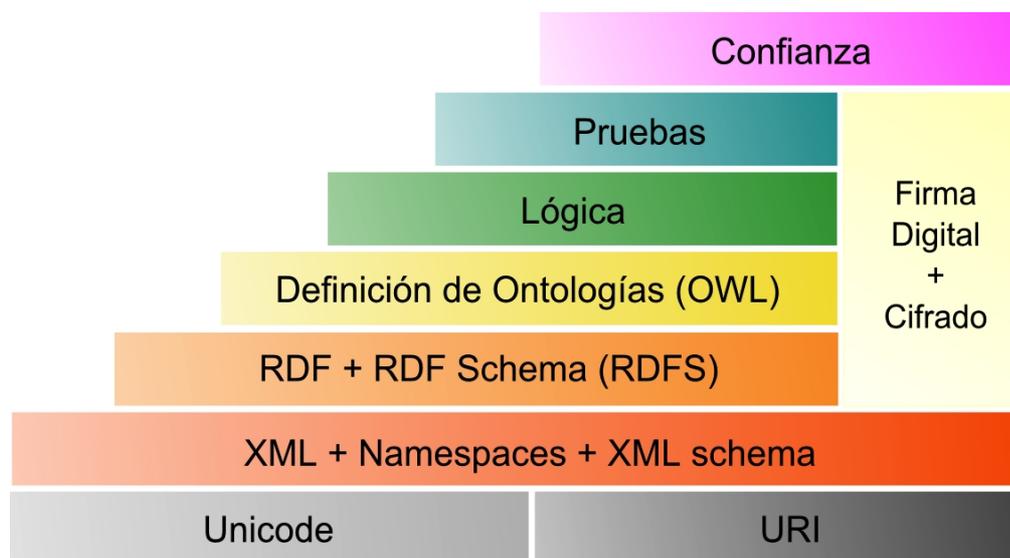


Ilustración 6: Arquitectura detallada de la Web Semántica. Basado en W3C.

Así pues, la Web Semántica tiene en XML un punto de partida para la definición de diferentes especificaciones, -como RDF, que constituyen la base semántica de los metadatos. Con XML también se construyen vocabularios de lenguajes que, como OWL, permiten un mayor grado de estructuración y abstracción para describir relaciones, propiedades y restricciones complejas entre recursos y objetos.

Una vez realizada esta descripción es posible obtener ontologías web que permitan operar con dichos recursos y organizarlos para el desarrollo de servicios de búsqueda de información en la Web para la localización de recursos descritos en forma de metadatos. Dichas ontologías web se tratan en realidad, tal y como nos indica García Jiménez (2004), de instrumentos para definir esquemas conceptuales tanto desde el punto de vista de estructura y composición, como de la especificación de restricciones semánticas, axiomas y reglas de inferencia.

Este tipo de ontologías permitirá la creación de agentes, para que las aplicaciones web puedan trabajar con metadatos. Los recursos de información se firmarán electrónicamente, para verificar la fiabilidad y autenticidad de los mismos. Mediante técnicas de cifrado se garantizará la confidencialidad en el intercambio de datos entre aplicaciones y usuarios.

2.5.3. Metadatos y ontologías

La representación de la información en la Web Semántica se fundamenta en dos conceptos: metadatos y ontologías. Los metadatos son descripciones estructuradas y codificadas que describen características y propiedades de objetos y recursos de forma que se facilite su localización, recuperación, valoración, administración, persistencia e interoperabilidad. No solo son capaces de realizar una descripción de dichos recursos, sino que también pueden hacer lo propio con los procesos en los que intervienen, sus componentes, las restricciones y las relaciones que se establecen entre ellos.

La dispersión de recursos, los grandes volúmenes de información, la actualización de contenidos y su heterogeneidad estructural son factores, entre otros, que han hecho que los procesos clásicos de catalogación se hayan demostrado ineficaces cuando se han aplicado a los recursos de información existentes en la Web. Gilligand-Swetland (2003) esboza una clasificación de los metadatos atendiendo a su posible ámbito de aplicación.

Para el presente trabajo el tipo de metadatos de interés es el Descriptivo lo que permitirá centrarnos en la representación de recursos dentro de procesos de recuperación de información. Por ello, aunque el concepto de metadato nos acerca a una noción familiar desde hace mucho tiempo en el ámbito bibliotecario y documental, en realidad su campo de aplicación y significado se ha ampliado enormemente con la aparición de las tecnologías asociadas a la Web y más concretamente a la Web Semántica (Méndez, 2002).

La necesidad de caracterizar recursos de información aplicando metadatos ha impulsado iniciativas para crear modelos de metadatos adaptados a la Web que se puedan integrar en las nuevas tecnologías de la Web Semántica. Tal vez la iniciativa más conocida en el ámbito de los metadatos de propósito general es la DCMI¹⁹.

Su principal ventaja es su simplicidad ya que propone un modelo de metadatos independiente del lenguaje que se utilice para la Web.

19 DCMI: *Dublin Core Metadata Initiative*. Organización cuyo objetivo es el fomento del uso de metadatos, a través de la adopción de estándares interoperables con vocabularios especializados, para describir recursos de información. Su aportación más conocida es el modelo de metadatos Dublin Core.

Tipo	Definición/Aplicación	Ejemplos
Administrativo	Usados en la gestión y administración de recursos de información	<ul style="list-style-type: none"> ■ Adquisición de información ■ Derechos y reproducción ■ Requerimientos legales para el acceso ■ Localización de información ■ Criterios de selección para la digitalización ■ Control de la versión
Descriptivo	Utilizados para representar recursos de información	<ul style="list-style-type: none"> ■ Registros de catalogación ■ Proporcionar ayuda en la búsqueda ■ Índices especializados ■ Describir relaciones entre recursos usando enlaces explícitos ■ Anotaciones de los usuarios
Preservación	Para salvaguardar los recursos de información	<ul style="list-style-type: none"> ■ Informar sobre las condiciones de uso de los recursos físicos ■ Informar sobre las acciones llevadas a cabo para preservar versiones físicas y digitales de recursos
Técnico	Relativos a cómo funcionan los sistemas o el comportamiento de los metadatos	<ul style="list-style-type: none"> ■ Documentación de hardware y software ■ Digitalización de información (formato, compresión) ■ Autenticación y seguridad (encriptación, claves, etc.) ■ Control de tiempo de respuesta de sistemas
Uso	Relativos al nivel y tipo de uso que se hace con los recursos informativos	<ul style="list-style-type: none"> ■ Información sobre versiones ■ Reutilización del contenido del recurso

Tabla 3: Ámbitos de aplicación de los metadatos

La iniciativa DCMI ha realizado una importante labor para que esta propuesta se integre como parte esencial de la infraestructura de la Web Semántica y para alcanzar un alto grado de estandarización mediante las normas ISO²⁰ 15836-2003 y ANSI²¹/NISO²² Z3950.85-2001.

El modelo de abstracción de DCMI tiene como objetivo especificar los componentes y constructores usados en Dublin Core definiendo su naturaleza y describiendo como se combinan entre si para crear estructuras de información. Este modelo cubre tres aspectos claves:

20 ISO: *International Organization for Standardization*. Organización cuyo objetivo es la promoción del desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica.

21 ANSI: *American National Standards Institute*. Organización encargada de supervisar el desarrollo de estándares en Estados Unidos. Es miembro de la ISO y coordina la interrelación entre estándares internacionales y estándares estadounidenses.

22 NISO: *National Information Standards Organization*. Es una organización estadounidense de normalización, que opera en el campo de la publicación, así como en aplicaciones bibliotecarias y bibliográficas. Representa los intereses de ANSI en el Comité Técnico 46 (Información y Documentación) de la ISO.

- Cómo se describen los **recursos**.
- Cómo se estructuran las **propiedades** que describen los recursos.
- Cómo se organiza el **vocabulario** para especificar las propiedades que describen los recursos.

Los elementos descriptivos propuestos por DCMI se muestran en la siguiente tabla.

Elemento	Significado
Contributor	Colaborador
Coverage	Covertura
Creator	Creador/Autor
Date	Fecha
Description	Descripción
Format	Formato
Identifier	Identificador
Language	Idioma
Publisher	Editor
Relation	Recurso relacionado
Rights	Derechos de autoría, intelectuales, de explotación, etc.
Source	Fuente original de la que se deriva el recurso descrito
Subject	Tema o asunto del recurso
Title	Título
Type	Naturaleza o género del recurso: colección, imagen, recurso interactivo, servicios, software, sonido, texto, etc.

Tabla 4: Elementos de Dublin Core

Ninguno de estos elementos es obligatorio. Pueden incluirse varias entradas de un mismo elemento en la descripción de un recurso. DCMI ofrece múltiples documentos sobre esquemas de implementación (RDFS o XML Schema), línea de codificación, guías de usuario, etc. Resulta imprescindible distinguir entre propuestas de metadatos y su codificación en un lenguaje como XML, SGML²³, RDF o similar. En ocasiones se ha confundido la idea de metadato con la de aquellos lenguajes que se han utilizado para codificar dichas propuestas. Así pues, es importante distinguir entre los distintos

²³ SGML: *Standard Generalized Markup Language*. Es un metalenguaje normalizado (norma ISO 8879:1986 SGML) que permite definir lenguajes de marcas para la construcción de documentos. SGML proviene del lenguaje GML.

elementos de un modelo de metadatos y el lenguaje con el que se expresa la sintaxis de dicho modelo.

Los metadatos pueden convertir la gran esfera de datos en la que se ha convertido la Web en un núcleo de información estructurada y descrita de forma precisa para poder realizar búsquedas más eficaces. Para poder realizar procesos más complejos es necesario aplicar un concepto complementario al de metadatos con la finalidad de introducir en los recursos de información un conocimiento que explicita de forma estructurada, cómo realizar deducciones a partir de axiomas aplicables en diferentes dominios.

La Web Semántica necesita el conocimiento en un formato que pueda ser procesado por máquinas. Ello implica usar un modelo público, estándar e integrado con otras tecnologías existentes para la Web Semántica. Las ontologías ofrecen una estrategia para representar este conocimiento ya que permiten trabajar con conceptos, en vez de expresiones o palabras clave, realizando una descripción del contenido semántico de los recursos de información independientemente de su representación sintáctica.

El término “Ontología” se refiere a aquella disciplina, parte de la Metafísica que trata del ser en general y de sus propiedades trascendentales²⁴. Esta disciplina tiene como objetivo la descripción de tipos de entidades del mundo real y la forma en la que se relacionan entre si. Una ontología web realiza una descripción de objetos definiendo clases, propiedades, relaciones y axiomas. Las ontologías permitirían desarrollar estructuras complejas para simplificar tareas de comunicación entre personas y máquinas con la intermediación de aplicaciones informáticas, que realizan procesos de intercambio de datos y simplifican la interoperabilidad de diferentes representaciones de datos con herramientas de unificación, traducción y mapeado.

Tal vez una de las definiciones más concisas de ontología es la de Gruber (1993) para quien “una ontología es una especificación explícita de una conceptualización”. Desde este punto de vista una ontología sería una abstracción, más o menos esquemática, de algún objeto o hecho del mundo real. Según este mismo autor los componentes de las ontologías para representar el conocimiento de un dominio serían las siguientes:

24 Definición del Diccionario de Lengua Española, 22ª edición. [En línea]
<http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=ontolog%C3%ADa> (Consulta: 12 de junio de 2008)

- **Conceptos:** Son los elementos fundamentales que se intenta representar formalmente con una ontología. Estos conceptos en realidad pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc.
- **Relaciones:** Definen de forma explícita cualquier tipo de interacción o enlace entre los diferentes conceptos del dominio. Estas relaciones dan forma a esquemas o estructuras cuyo grado de complejidad puede variar dependiendo de la tipología de relaciones, restricciones semánticas de las mismas o el número de conceptos que intervienen. Las relaciones se definen usando clases, subclases, asociaciones de conceptos, etc.
- **Funciones:** Es un tipo de relación en la que se identifica un elemento a partir del cálculo de una función en la que intervienen varios elementos de la ontología.
- **Instancias:** Representan objetos determinados de un concepto.
- **Axiomas:** Teoremas definidos sobre la base de relaciones que deben cumplir ciertos elementos de la ontología. Por ejemplo: "Si A es de la clase Z entonces B es de la subclase W".

Así pues, una ontología es el resultado de seleccionar un dominio, y aplicar sobre el mismo un método para obtener una representación formal de los conceptos que contiene y de las relaciones de diverso tipo que existen entre los mismos, . El proceso de definición de una ontología conlleva una serie de tareas:

- Definición de las clases existentes en la ontología.
- Estructurar dichas clases en una jerarquía de taxonomías (clase-subclase).
- Definir las propiedades (slots) aplicables, así como el dominio y rango de valores permitidos.
- Rellenar los valores de las propiedades con ejemplos.

Un sistema de recuperación de información basado en ontologías puede realizar consultas que no se limitan únicamente a buscar elementos a partir del valor de sus propiedades. Permite comprobar que dichos elementos participan en ciertas relaciones e incluso realizar inferencias a partir de las propiedades de herencia de las relaciones especificadas en los axiomas. Para ello, toda ontología ha de poseer al menos una doble vertiente:

- **Semántica:** En la que se formaliza, expresa y estructura el contenido con niveles de detalle relativamente flexibles para una descripción de objetos y hechos del mundo real, tanto en sus propiedades como en su relaciones.
- **Pragmática:** En donde se definen las posibles aplicaciones de una ontología con métodos de diseño adecuados y teniendo en cuenta el contexto de uso aplicando criterios lógicos de razonamiento, de organización y de rigurosidad, todo ello para obtener dominios de conocimiento, funcionales y aplicables.

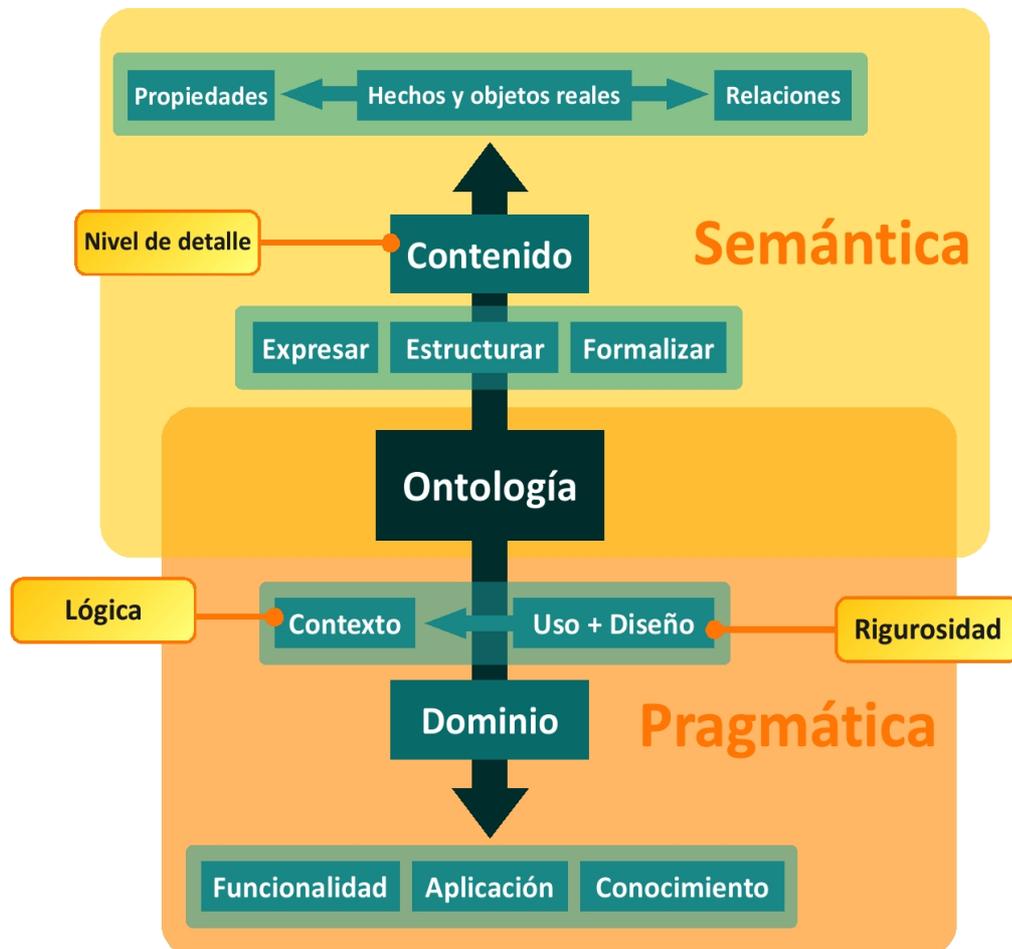


Ilustración 7: Vertientes Semántica y Pragmática de una ontología

Las ontologías necesitan revisarse para facilitar los procesos de mantenimiento a través de análisis de evaluación que midan su grado de pragmatismo y por lo tanto de adaptación a lo largo del tiempo. Como se ha indicado anteriormente, para llevar a la práctica la Web Semántica son necesarios lenguajes de marcado apropiados que

permitan crear documentos donde se represente el conocimiento definido en una ontología.

Con RDF y RDFS es posible caracterizar ciertas propiedades de conceptos de un dominio concreto al tiempo que es posible jerarquizar dichos conceptos. Aún así, es necesario alcanzar mayor grado de expresividad y abstracción para representar funciones y axiomas de una ontología. El lenguaje desarrollado por el W3C, que comienza a constituirse en un estándar para la representación de ontologías, es OWL que complementa y amplía a RDF y los esquemas RDF (RDFS).

No basta con crear un lenguaje de marcado, sino que también hay que desarrollar aplicaciones que exploten estos recursos interpretando los esquemas ontológicos y aplicando los axiomas definidos. De esta forma será posible comprobar y mantener una correspondencia coherente entre una ontología y las instancias de conceptos publicadas en la Web. Sería posible el desarrollo de herramientas para realizar búsquedas expandidas a partir de inferencias que se ejecutarían teniendo en cuenta los axiomas definidos en una o varias ontologías.

2.6. Tecnologías de la Web Semántica para la Representación de la Información

La Web Semántica no es una única tecnología, sino que es el resultado de la aplicación de diferentes tecnologías con un grado de abstracción cada vez mayor, desde el punto de vista de los procesos de Representación de la Información.

Son tres las tecnologías clave: XML, RDF y OWL. Ya se ha hecho mención a estas siglas en puntos anteriores de este proyecto. A continuación se incluye un análisis más detallado de su significado y papel en la Web Semántica, en especial de RDF.

2.6.1. XML

XML (sigla en inglés de *eXtensible Markup Language*)²⁵ es un metalenguaje para la definición de lenguajes que permiten el intercambio de datos estructurados correctamente con independencia de la plataforma en la que se ejecuten las aplicaciones.

²⁵ Puede encontrarse más información sobre XML en <http://www.w3.org/XML/>

El origen de XML se remonta a la creación del lenguaje GML²⁶ desarrollado en IBM durante 1969. El objetivo de este lenguaje era el almacenamiento de grandes volúmenes de documentos de naturaleza y estructuración muy heterogénea para su posterior procesamiento describiendo la estructura de un documento y no su formato de visualización.

En 1986 ISO normalizó GML con la norma 8879:1986 en donde se definía el lenguaje SGML (*Standard Generalized Markup Language*). La potencia de SGML se basa en la generalidad de uso sin que esté limitado por el contenido, formato, tamaño o estructura del documento representado. Por lo tanto SGML tiene una serie de características que aportan sus correspondientes ventajas en la elaboración de documentos digitales:

- **Legible:** El documento marcado con SGML puede ser leído por un humano o por una aplicación informática para su interpretación y procesamiento.
- **Extensible:** En SGML no existe un vocabulario definido de etiquetas para estructurar los documentos, sino que son los propios autores quienes definen las etiquetas que conformarán el vocabulario para estructurar los documentos.
- **Estructuración:** Los documentos se organizan siguiendo la estructura jerárquica de su contenido de manera que se pueden definir partes del mismo en función de partes más amplias.
- **Modular:** SGML permite separar la definición de la estructura de un documento de la información contenida en el mismo para que pueda ser procesada y visualizada de múltiples formas
- **Validación:** Cualquier documento SGML hace referencia a la descripción de su sintaxis y gramática de manera que pueda ser validado para comprobar que ha sido construido correctamente.

A pesar de estas ventajas, SGML no podía combinar elementos definidos por diferentes vocabularios. SGML se orientó en un primer momento a la elaboración

²⁶ GML: *Generalized Markup Language*. Lenguaje de marcas desarrollado por Charles Goldfarb, Edward Mosher and Raymond Lorie para SCRIPT, una aplicación de IBM para el formateo de textos. Como curiosidad indicar que las iniciales de los apellidos de los autores también conforman el término GML.

manual o asistida de documentos según una DTD²⁷ y no al intercambio de información entre aplicaciones. Pero la principal limitación de este lenguaje es su excesiva complejidad. El uso de SGML para la creación de un lenguaje para el marcado de documentos constituye un esfuerzo muy arduo debido a que se requiere un gran dominio y comprensión de su sintaxis y gramática.

Una de las aplicaciones más conocidas y aplicadas de SGML es el lenguaje HTML²⁸, desarrollado para dotar de formato visual a los datos con un conjunto definido de marcas (tipo de letra, énfasis, justificación, etc.). Se trata de un lenguaje totalmente ligado a la creación de la Web y que se ha venido desde entonces. Pero la estructuración de datos con HTML y la consiguiente representación de información conlleva un gran inconveniente: ya que no permite separar el modo en el que se muestra la información de la estructura del contenido informativo.

El uso conjunto de HTML y hojas de estilo CSS²⁹ simplifica la elaboración del código HTML permitiendo desarrollar proyectos de carácter modular. Esto es posible al aplicar un mismo estilo visual a un conjunto de documentos HTML de un modo sencillo ya que se mantiene un único fichero donde se definen los estilos visuales. Existen mecanismos para describir ciertos metadatos en la cabecera del documento o una estructuración basada en las marcas de encabezamiento (*headers: H1, H2, H3...*) en el cuerpo del mismo. Sin embargo en modo alguno son suficientes para expresar en su totalidad el contenido semántico y estructural de un documento.

Otro problema de HTML es el limitado número de marcas de las que dispone. Mediante SGML se ha definido un conjunto de etiquetas HTML que ha aumentado durante el ciclo de vida de este lenguaje. La excesiva complejidad de SGML y las limitaciones de HTML motivaron que el W3C desarrollara un nuevo estándar en 1996: XML. En realidad se trata de un lenguaje que no está únicamente orientado a su uso en el ámbito de la Web, sino que se diseñó pensando en la estructuración, intercambio y

27 DTD: *Document Type Definition*. Es una definición, de un documento SGML o XML, que especifica restricciones en la estructura y sintaxis del mismo.

28 HTML: *HyperText Markup Language*. Lenguaje de marcas desarrollado originalmente en el CERN (*Conseil Européen pour la Recherche Nucléaire*) por Tim Bernners-Lee dentro del proyecto World Wide Web. y orientado a la visualización de información en la Web a partir de una organización hipertextual de los documentos.

29 CSS: *Cascading Style Sheets*. Es un lenguaje usado para definir atributos de formato como color, tipo de letra o márgenes entre otros, asociados a la presentación visual de un documento HTML/XHTML o XML.

publicación de información con independencia del dispositivo, plataforma o aplicación de tratamiento.

Un documento XML contiene datos cuya estructura se define a partir de lo que se conoce como DTD que veremos más adelante. La estructura XML sigue un esquema jerárquico para que las diferentes etiquetas estén correctamente incluidas unas dentro de otras. Los documentos que siguen una determinada estructura definida en una DTD se dicen que son documentos XML bien formados.

Es posible incluir etiquetas sin contenido para dotar de mayor flexibilidad la representación de contenido. Los diferentes elementos se estructuran en torno a un elemento raíz que todo documento XML bien formado debe incluir. Los valores de los atributos se delimitan con comillas (dobles o simples). XML permite la inclusión de datos literales y comentarios.

```
<?xml version="1.0" ?>
<catalogo>
  <libro>
    <titulo>Moby Dick</titulo><autor>Herman Melville</autor>
  </libro>
  <libro>
    <titulo>Platero y yo</titulo><autor>Juan Ramón Jiménez</autor>
  </libro>
</catalogo>
```

Ejemplo 1: Código XML

Existen dos formas para declarar la estructura de un documento XML: Con una DTD (Document Type Definition) o con XML Schema. Una DTD define las restricciones de la estructura y sintaxis de un documento XML y SGML y no permite declarar elementos con el mismo nombre pero con validez diferente dentro de otros elementos. Además tampoco es posible indicar tipos de datos complejos para especificar cómo se codifican los datos.

```
<!DOCTYPE libreria [
<!ELEMENT libro (titulo+, autor+)>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
]>
```

Ejemplo 2: DTD correspondiente al ejemplo 1

XML Schema permite solventar las limitaciones de las DTD al ofrecer una definición mucho más expresiva y exacta aprovechando el propio lenguaje XML para integrar diferentes definiciones gracias al uso de los Namespaces.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www3.org/2001/XMLSchema"
  targetNamespace="http://www.dattatec.com"
  xmlns=" http://www.dattatec.com" elementFormDefault="qualified">
  <xs:element name="Libreria">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Libro" minOccurs="1"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Libro">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Titulo" minOccurs="1" maxOccurs="1"/>
        <xs:element ref="Autor" minOccurs="1"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Titulo" type="xs:string"/>
  <xs:element name="Autor" type="xs:string"/>
</xs:schema>
```

Ejemplo 3: Definición XML Schema equivalente a la definición de DTD del ejemplo 1

Aunque una declaración XML Schema es mucho menos compacta que una DTD, tiene la ventaja de una mayor expresividad y la reutilización de la definición de la estructura aplicando conjuntamente una o varias definiciones XML Schema.

Es posible convertir código XML directamente en un documento HTML o XHTML³⁰ (aplicando hojas de estilo XSL/XSLT sin que sea necesario elaborar complejos guiones de programación que realicen esta transformación. De forma complementaria, XPATH funciona como un lenguaje parecido al de manejo de expresiones regulares para recorrer el árbol de la estructura de un documento XML para seleccionar elementos y realizar ciertas operaciones básicas como la aplicación de estilos de transformación XSLT.

30 XHTML: *eXtensible Hypertext Markup Language* es un lenguaje para el marcado de documentos pensado para sustituir a HTML ya que ha sido definido a partir de XML y requiere un mayor grado de rigurosidad. Esto ayuda a la accesibilidad de los documentos en diferentes navegadores web al utilizar un lenguaje más estricto y menos ambiguo que HTML.

Con XML podemos describir cualquier tipo de información con una dinámica sencilla e intuitiva. En un principio, las diferentes tecnologías asociadas a XML, como XSL/XSLT, XPATH, XML Schema o DTD, permiten explotar de un modo considerable las posibilidades de XML. También pueden servir como un formato de intercambio de datos entre aplicaciones, de forma que se incrementa la interoperabilidad entre plataformas y sistemas aportando una herramienta válida en diversas situaciones y propósitos. En definitiva, XML constituye la base para el desarrollo de lenguajes orientados a la representación de información altamente estructurada y con diferentes niveles de abstracción. El uso conjunto de diferentes tipos de vocabularios permite representar diferentes tipos de objetos manejando especificaciones normalizadas para un uso posterior. XML es el salto cualitativo por el cual se consigue separar estructura, contenido y formato visual. Por todo ello, XML supone en la práctica el origen de la Web Semántica.

Cabría preguntarse sobre la causa que motiva que HTML continúe siendo tan utilizado. Es preciso aclarar que uno de los primeros desarrollos de envergadura con XML fue XHTML. Esta nueva especificación, más estricta, está definida con XML y constituye una herramienta de gran valor para que los documentos web cumplan con los requisitos de accesibilidad en cualquier entorno de consulta³¹. XHTML es mucho más estricto que HTML y obliga a que los documentos estén bien formados y sean válidos según la especificación correspondiente. Pese a todo la función de XHTML sigue siendo idéntica a la de HTML: La estructuración visual, diseño y maquetación de los documentos web.

La representación y almacenamiento de contenidos web con XML implica un análisis detallado y amplio de los mismos para obtener una serie de definiciones con las estructuras de los diferentes tipos de información. Esto implicaría un cambio de paradigma en la creación de contenidos para web ya que los procesos de elaboración de documentos se transformarían en procesos de gestión de información estructurada. Sería algo más parecido a la gestión de información aprovechando las ventajas inherentes al uso de bases de datos.

31 La Accesibilidad web ha sido utilizada de forma parcial ya que muchos autores relacionados con el diseño web han asociado este concepto al del desarrollo de contenidos adaptados para su consulta por parte de personas con deficiencia visual. El W3C propone una definición mucho más amplia asociando la Accesibilidad web a la idea de que los documentos puedan consultarse con independencia del dispositivo y aplicación cliente utilizados.

Para la mayoría de los usuarios esto sería algo complejo de asumir debido a que la obtención de resultados inmediatos en el navegador no está supeditada a requisitos muy estrictos en cuanto al grado de estructuración de los contenidos o la corrección del código HTML de los documentos. Estos mismos usuarios siguen teniendo una visión de la Web como un espacio en el que se pueden publicar contenidos de forma totalmente libre y sencilla siendo ajenos a las implicaciones que supone el uso y la potencialidad de XML. Es paradójico pensar que esta libertad extrema en la edición de contenidos que ha facilitado la difusión de la Web se haya convertido al mismo tiempo en el principal obstáculo en el desarrollo de la Web Semántica.

Pese a ello, de forma silenciosa y constante, se desarrollan muchas iniciativas que usan XML para estructurar información y desarrollar servicios. Sin ir más lejos, XML se utiliza de forma amplia en el almacenamiento de estructuras de información en sistemas de gestión de información basados en Web, para el intercambio de datos entre aplicaciones y bases de datos o para el desarrollo de especificaciones que como RDF/RDFS suponen un salto cualitativo respecto a alcanzar los objetivos de la Web Semántica.

2.6.2. RDF

RDF (sigla en inglés de *Resource Description Framework*)³² es un modelo para la representación de información estructurada en forma de metadatos con el objetivo de describir recursos disponibles en Internet o de otro tipo. En este modelo, la descripción de dichos recursos se realiza en forma de tripletas del tipo sujeto-predicado-objeto.

De esta forma el sujeto es el recurso a describir, el predicado es una propiedad o relación del recurso y el objeto es el valor asignado a esta propiedad o el recurso con el que establece la relación. El W3C ha redactado seis documentos que ofrecen una descripción completa de RDF y que cubren los siguientes aspectos:

- Manual de Introducción a RDF (*RDF Primer*). Un manual que ofrece el conocimiento básico requerido para un uso efectivo de RDF al tiempo que describe su sintaxis XML. Describe como definir vocabularios RDF con un Lenguaje de Descripción de Vocabularios (W3C, 2004b1).

³² Puede encontrarse más información sobre RDF en <http://www.w3.org/RDF/>

- **Conceptos y Sintaxis abstracta de RDF (*RDF Concepts and Abstract Syntax*).** Define la sintaxis abstracta en la que se basa RDF y su correspondencia con su semántica formal. Incluye también análisis acerca de objetivos del diseño con RDF, conceptos clave, normalización en la codificación de caracteres y manejo de referencias URI (W3C, 2004b2).
- **Especificación de la Sintaxis RDF/XML (*RDF/XML Syntax Specification*).** Define la sintaxis XML para RDF (W3C, 2004b3).
- **Lenguaje de descripción de Vocabularios RDF: Esquema RDF (*RDF Vocabulary Description Language: RDF Schema*).** Define un lenguaje para construir vocabularios RDF (W3C, 2004b4).
- **Semántica RDF (*RDF Semantics*).** Este documento ofrece una semántica precisa y un sistema de correspondencia completa de reglas de inferencia para RDF y RDFS (W3C, 2004b5).
- **Casos de prueba de RDF (*RDF Test Cases*).** Contiene una relación de pruebas de las diferentes características y elementos del vocabulario y sintaxis de RDF para comprobar que la especificación cumple con los requisitos y objetivos esperados (W3C, 2004b6).

Las declaraciones RDF pueden representarse visualmente con grafos donde sujeto y objeto adoptan la forma de nodos unidos por un arco dirigido desde el sujeto al objeto y que representa el predicado. La expresión “La película Brazil fué dirigida por Terry Gilliam” podría representarse gráficamente con la figura siguiente.



Ilustración 8: Grafo RDF básico

En la figura anterior puede observarse cómo los tres elementos de la tripleta hacen referencia a recursos con referencias URIs. En el caso del sujeto, la URI es la dirección del recurso sobre el que hacemos la declaración. En el caso del predicado (representado con un arco) la propiedad que hace referencia al creador de la página

apunta al elemento de Dublin Core “creator” cuya descripción en RDF está disponible en <http://purl.org/dc/elements/1.1/creator>. Finalmente, el objeto apunta a una URI que hace referencia al director de la película. RDF utiliza referencias URI para la localización de recursos, estén disponibles o no en red, proporcionando una forma más flexible y amplia que la proporcionada por URL.

De esta forma, sujeto, predicado y objeto hacen referencia a un recurso identificado con una URI. El objeto puede hacer referencia a una URI o tener asignado un valor de cadena literal o un literal tipificado cuyo dominio es definido en otro recurso. Esto permite aprovechar otros vocabularios XML e integrarlos dentro de declaraciones RDF. El grafo anterior puede representarse en forma de tripletas tal y como se muestra en la tabla siguiente.

Sujeto	Objeto	Predicado
http://www.cine.org/peliculas/brazil	http://purl.org/dc/elements/1.1/creator	http://www.cine.org/personas/terrygilliam

Tabla 5: Equivalencia en tripletas de la ilustración 8

Para simplificar las referencias URI, pueden definirse prefijos que permitan acortar considerablemente las declaraciones RDF. A continuación se incluye un ejemplo.

```
prefix rdf:, namespace URI: http://www.w3.org/1999/02/22-rdf-syntax-ns#
prefix rdfs:, namespace URI: http://www.w3.org/2000/01/rdf-schema#
prefix dc:, namespace URI: http://purl.org/dc/elements/1.1/
prefix owl:, namespace URI: http://www.w3.org/2002/07/owl#
prefix ex:, namespace URI: http://www.cine.org/
prefix extermns, namespace URI http://www.cine.org/terms/
prefix expersonas, namespace URI http://www.cine.org/personas/
prefix expeliculas, namespace URI http://www.cine.org/peliculas/
prefix xsd:, namespace URI: http://www.w3.org/2001/XMLSchema#
```

Ejemplo 4: Definición de prefijos

El siguiente ejemplo más complejo incluye literales tipificados y su correspondiente representación con tripletas RDF haciendo uso de prefijos:

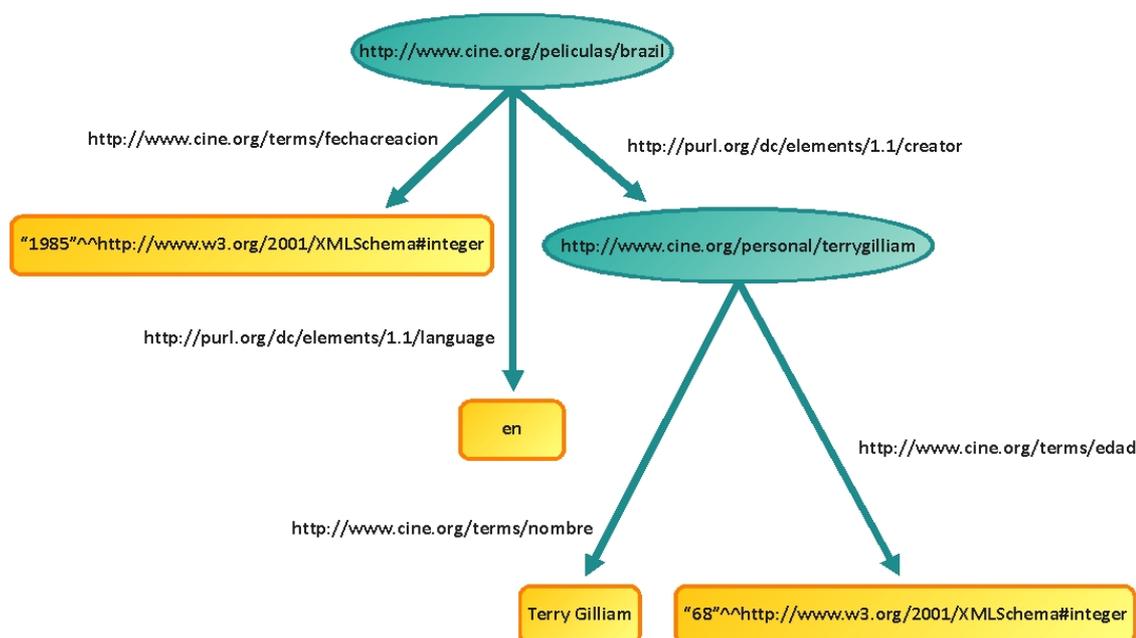


Ilustración 9: Grafo RDF complejo

La representación en forma de triplas de la ilustración anterior sería la siguiente.

Sujeto	Objeto	Predicado
expeliculas:brazil	dc:creator	expersonas:terrygilliam
expeliculas:brazil	exterm:fecha creacion	"1985"^^xsd:integer
expeliculas:brazil	dc:language	"en"
expersonas:terrygilliam	exterm:nombre	"Terry Gilliam"
expersonas:terrygilliam	exterm:edad	"68"^^xsd:integer

Tabla 6: Equivalencia en triplas de la ilustración 9

RDF también contempla el uso de valores estructurados de datos para realizar declaraciones de manera que la representación de objetos reales puede realizarse de un modo más sencillo. Esta estructuración de datos puede construirse directamente estableciendo múltiples predicados que definan diversas propiedades de un recurso e incluso nodos en blanco (a los que se les asigna un identificador) para agrupar lógicamente valores agregados. Para dotar a RDF de una sintaxis existe un vocabulario que permite la codificación de RDF a través de XML denominado RDF/XML³³. Las

33 Existen otras codificaciones de RDF como N3, N-TRIPLE o TriX entre otras. Para el presente trabajo se utilizará únicamente la codificación RDF/XML, menos compactas que otras soluciones, pero más

declaraciones de la figura anterior se podrían codificar según la tabla siguiente.

```
1. <?xml version="1.0"?>
2. <!DOCTYPE rdf:RDF [<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
3. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.   xmlns:dc="http://purl.org/dc/elements/1.1/"
5.   xmlns:exterm="http://www.cine.org/terms/">
6.   <rdf:Description rdf:about="http://www.cine.org/peliculas/brazil">
7.     <dc:creator rdf:Resource="http://www.cine.org/personas/terrygilliam"/>
8.     <exterm:fechacreacion
rdf:datatype="&xsd;integer">1985</exterm:fechacreacion>
9.     <dc:language>en</dc:language>
10.  </rdf:Description>
11.  <rdf:Description rdf:about="http://www.cine.org/personas/terrygilliam">
12.    <exterm:nombre>Terry Gilliam</exterm:nombre>
13.    <exterm:edad rdf:datatype="&xsd;integer">68</exterm:edad>
14.  </rdf:Description>
15. </rdf:RDF>
```

Ejemplo 5: Código RDF/XML correspondiente a la ilustración 9 y a la tabla 6

En la línea 1 del ejemplo anterior se indica que código sigue la sintaxis XML. En la línea 2 se incluye una definición el tipo de documento (RDF) y la creación de la entidad “xsd” asociada a la especificación de XML Schema con la finalidad de hacer referencia a diferentes tipos de datos acertando su referencia. Las líneas 3, 4 y 5 definen los Namespaces para “dc” (Dublin Core) y “exterm” (Definición de atributos propios). Desde la línea 6 a la 10 se describe el recurso “http://www.cine.org/peliculas/brazil” indicando que el creador es el recurso “http://www.cine.org/personas/terrygilliam”, su fecha de creación (puede observarse el uso de la entidad “xsd” para indicar que el tipo de dato es entero) y el idioma original. También se realiza una descripción del recurso “http://www.cine.org/personas/terrygilliam” indicando su nombre y su edad.

Una de las ventajas de XML, y por tanto de RDF/XML, es la adaptación del nivel de abstracción para facilitar su legibilidad por las personas. A mayor nivel de abstracción el código se compacta (tanto las referencias URI como la sintaxis de las tripletas) pero su legibilidad se vuelve más compleja. En ocasiones el diseño puede estar orientado a la lectura del código por parte de máquinas por lo que se hace necesario desarrollar documentos más compactos para optimizar su transmisión y almacenamiento. RDF también dispone de un lenguaje de descripción de vocabularios RDF conocido como Esquema RDF (RDF Schema ó RDFS). Con RDFS pueden describirse esquemas sencillos usando clases y subclases así como propiedades, su dominio de aplicación y su rango

expresiva y con más posibilidades de integración con otros vocabularios XML.

de valores posibles. Se trata de un lenguaje muy básico para la descripción complementado por OWL para esquemas más complejos.

Otras estructuras o declaraciones complementarias que ofrece RDF son las siguientes:

- **Contenedores:** Permiten la descripción de grupos de elementos. Pueden describir grupos en donde el orden de los elementos no es importante (*Bag*), otros contenedores en donde el orden de los elementos es relevante (*Sequence*) y grupos con elementos alternativos de los cuales puede seleccionarse uno para asignarlo a una propiedad (*Alternative*).
- **Colecciones:** Permiten crear grupos cerrados de elementos. Esta estructura al contrario que un Contenedor, que es abierto, permite enumerar de un modo preciso los elementos que la componen.
- **Modelización (*Reification*):** Permiten describir con RDF declaraciones de alto nivel con la intención de expresar conocimiento sobre otras declaraciones. En este caso la declaración se contempla como un recurso, representando explícitamente al sujeto, predicado, objeto y tipo de la afirmación.

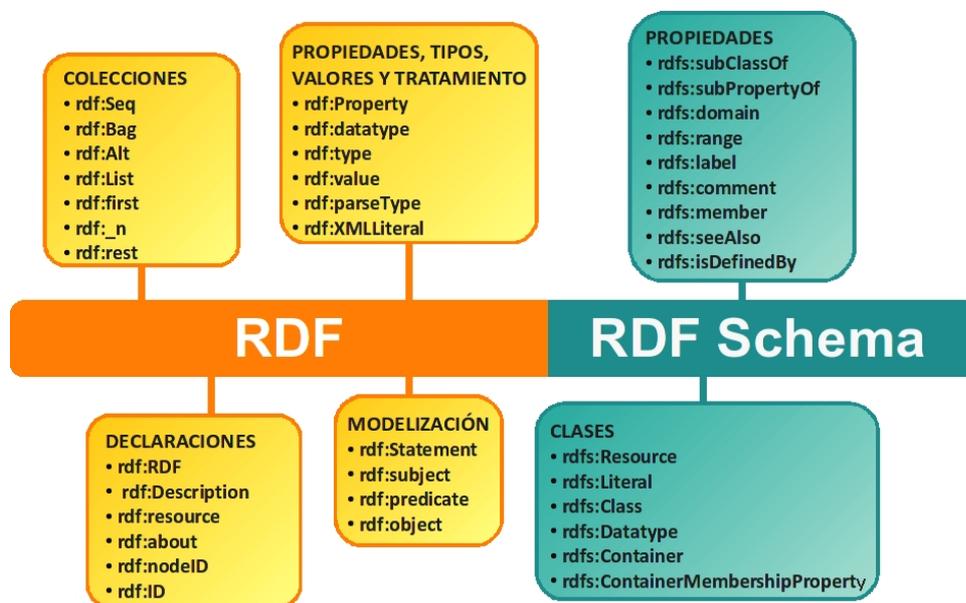


Ilustración 10: Constructores de RDF y RDF Schema

Se han construido multitud de vocabularios RDF para su aplicación en distintas

disciplinas. El ámbito de actuación de estos vocabulario cubre se usa tanto en la descripción de recursos de Internet y también en otros campos tales como el de componentes eléctricos y electrónicos, información médica, distribuciones de software o componentes y características de dispositivos informáticos.

Algunos ejemplos de aplicaciones de RDF son las siguientes:

- **Dublin Core:** Existe un vocabulario RDF que implementa el modelo de metadatos DCMI. Es importante apuntar que Dublin Core forma parte funcional de muchos vocabularios RDF para el almacenamiento de metadatos en documentos, lo que hace innecesaria la declaración de propiedades que ya incluye el vocabulario Dublin Core.
- **PRISM (*Publishing Requirements for Industry Standard Metadata*):** Este vocabulario trata de cubrir las necesidades de descripción de metadatos de cualquier tipo de contenido asociado a la industria editorial.
- **RSS (*RDF Site Summary*):** Se trata de un vocabulario RDF cuyo objetivo es la descripción de aspectos básicos y sencillos de recursos de información (principalmente páginas web). Este vocabulario es mucho más simple que Dublin Core pero lo suficientemente funcional como para ser aplicado en procesos de sindicación de contenidos.
- **CIM/XML (*Common Information Model/XML*):** Es un subconjunto reducido del vocabulario de RDF para su aplicación en la descripción de modelos de unidades de sistemas de potencia eléctrica con la finalidad de describir e intercambiar información acerca de estas unidades.

2.6.3. OWL

OWL (*Web Ontology Language*)³⁴ está diseñado para su uso en aplicaciones que requieran realizar funciones de procesamiento del contenido informativo y no limitarse únicamente a la presentación de la misma para su consulta por parte de personas.

Este lenguaje facilita una mayor capacidad de interpretación por parte de máquinas

³⁴ Puede encontrarse más información sobre OWL en <http://www.w3.org/2004/OWL>

del contenido Web que el que ofrecen XML, RDF o RDFS, proporcionando un vocabulario adicional y un semántica formal. En definitiva OWL es un lenguaje para definir ontologías web.

El lenguaje OWL se describe a lo largo de seis documentos:

- Introducción a OWL (*OWL Web Ontology Language Overview*). Ofrece una breve introducción sobre las principales características del lenguaje (OWL, . 2004a2)
- Guía de OWL (*OWL Web Ontology Language Guide*). Presenta una demostración del uso de OWL con un ejemplo bastante extenso (OWL, . 2004a1).
- Guía de referencia de OWL (*OWL Web Ontology Language Reference*). Ofrece una descripción sistemática y resumida de las primitivas de modelado de OWL, con la sintaxis de intercambio de RDF/XML para OWL. Este documento sirve como una guía de referencia para usuarios del lenguaje OWL (OWL, .2004a3).
- Semántica y sintaxis abstracta de OWL (*OWL Web Ontology Language Semantics and Abstract Syntax*). Definición formal del lenguaje (OWL, .2004a4).
- Casos de prueba de OWL (*OWL Web Ontology Language Test Cases*). Este documento incluye una extensa relación de casos de prueba de OWL (OWL, . 2004a5).
- Casos de uso y requisitos de OWL (*OWL Web Ontology Language Use Cases and Requirements*). Contiene una amplia relación de casos de pruebas (OWL, . 2004a6).
- Sintaxis de presentación XML de OWL (*OWL Web Ontology Language XML Presentation Syntax*). Proporciona esquemas XML para la sintaxis de presentación correspondiente a los tres sublenguajes de OWL: Lite, DL y Full (OWL, .2003).

El diseño de OWL se orienta al crecimiento de las posibilidades de la Web Semántica. Recordemos que XML únicamente ofrece una sintaxis para documentos estructurados y que no tiene en cuenta el contenido semántico de los mismos. XML Schema permite

añadir ciertas restricciones a la estructura de dichos documentos XML y amplía las posibilidades de este lenguaje con los tipos de datos. RDF es un modelo de datos para la representación de recursos web y las relaciones entre los mismos con una semántica muy sencilla que puede ser codificada con XML, al tiempo que se utiliza RDFS para describir las propiedades y clases para generar estructuras jerárquicas.

OWL añade más posibilidades para describir propiedades y clases, como relaciones entre clases, cardinalidad, clases enumeradas o características de propiedades. Ofrece tres sublenguajes con distintos niveles de expresividad adaptados a las necesidades de uso:

- **OWL Lite:** Orientado para la elaboración de clasificaciones jerárquicas y restricciones simples. OWL Lite ofrece una forma sencilla de implementar tesauros, taxonomías y otros esquemas conceptuales.
- **OWL DL:** Ofrece una expresividad máxima garantizado que todos los resultados puedan ser computados en un tiempo finito. OWL DL supedita el uso de los constructores de OWL a la verificación de ciertas restricciones.
- **OWL Full:** Permite la máxima expresividad con la libertad sintáctica de RDF sin garantizar que los resultados puedan ser computados. OWL Full posibilita que una ontología incremente el significado de un vocabulario predefinido.

Los diferentes constructores de OWL Lite, OWL DL y OWL Full puede agruparse en una serie de categorías funcionales tal y como se muestra en la siguiente ilustración.



Ilustración 11: Constructores de OWL Lite, DL y Full

Podría decirse que OWL Full se contempla como una extensión de RDF mientras que OWL Lite y OWL DL pueden considerarse extensiones de una visión restringida de RDF. Es decir, todo documento de OWL es un documento RDF y cada documento RDF es un documento OWL Full, pero sólo algunos documentos RDF pueden considerarse documentos OWL Lite o DL.

2.7. Web 2.0 y Web Semántica: el camino hacia la Web Ubícua

Es posible encontrar, en algunas ocasiones, una equivalencia terminológica entre Web Semántica y Web 2.0³⁵. También se está comenzado a hablar de Web 3.0 igualmente para referirse al concepto de Web Semántica. Pensamos que expresarse con neologismos tan efectivas como éstos puede crear más confusión que ayudar a comprender los conceptos que quieren expresar.

Ya hemos comentado que la Web se diseñó en un principio como una herramienta de

³⁵ Aunque el origen del término Web 2.0 se atribuye a Tim O'Reilly en realidad fue acuñado por Dale Dougherty de O'Reilly Media en una lluvia de ideas con Craig Cline de MediaLive para desarrollar ideas para una conferencia en 2004

publicación y comunicación. Posteriormente se incorporaron nuevas funcionalidades con la aparición de tecnologías que permitieron la interacción entre contenidos y usuarios, así como la explotación de bases de datos para la elaboración de páginas web dinámicas.

Éste fue el detonante principal para el desarrollo de lo que se conoce como Web 2.0, o tal vez de un modo más acertado y descriptivo lo que se denomina Web Social³⁶. La idea fundamental es la del establecimiento de redes o comunidades de usuarios que trabajan con una serie de servicios basados en aplicaciones web como son los blogs, los servicios de publicación de contenidos multimedia, las redes sociales, los wikis o los buscadores sociales basados en folksonomías.

El objetivo fundamental de la Web 2.0 es la de fomentar la colaboración con la finalidad de difundir e intercambiar información de forma rápida y sencilla. Se trata en realidad de una aplicación concreta de la Web convencional ya que no todos los desarrollos web actuales tienen que basarse en conceptos asociados a la Web 2.0. Con la Web Semántica sucede algo similar.

La Web Semántica supondrá añadir metadatos descriptivos a los recursos de información, incluso algunos de estos recursos estarán definidos con lenguajes que permitirán expresar su contenido semántico. Esto no supondrá la desaparición de la Web Social sino la aparición de una nueva tecnología que reorientará, adaptará y mejorará desarrollos previos.

Para Davis (2008) la evolución de la Web no se detiene en la Web Semántica. Propone un horizonte alrededor del año 2020 en el que las tecnologías asociadas al desarrollo de la Web Social se combinarían con las de la Web Semántica. Este proceso conformaría una nueva orientación en el desarrollo de sistemas y software, en la que la propia Web estaría presente en el núcleo de entornos de trabajo que hoy día la consideran como una extensión para la conectividad o como repositorio de información.

La Web Ubícua integraría los aspectos de conectividad social de la Web 2.0 y los de formalización de la información de la Web Semántica. Todo ello dentro de un nuevo paradigma en donde la Web no sería una aplicación más, sino que constituiría el

³⁶ Aunque tal vez sea más exacto utilizar la expresión "Web Social" utilizaremos la de "Web 2.0", por tratarse del término más difundido para referirse al concepto que estamos tratando.

centro en torno al cual se diseñarán sistemas operativos, aplicaciones y nuevos mecanismos para almacenar y recuperar la información que trasciendan el disco duro de un ordenador personal. Es posible que este escenario marque el comienzo de la desaparición del concepto de “Ordenador Personal”, como objeto físico en el que cada usuario almacena la información, así como las aplicaciones que la gestionan, deslocalizándose todo ello hacia la Web.

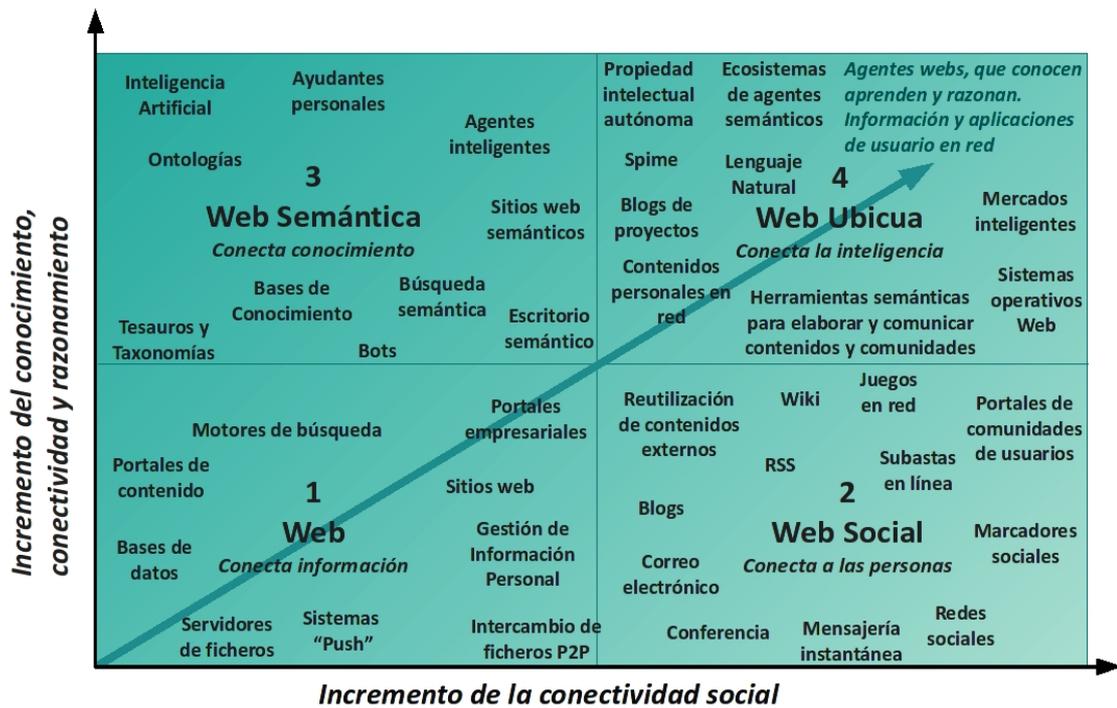


Ilustración 12: La Evolución de la Web. Traducido y adaptado de Davis (2008)

Desde este punto de vista, tanto la Web Social, como la Web Semántica, constituyen estados intermedios y necesarios en la evolución de la Web en una dirección en la que los metadatos sean utilizados ya no solo por personas o aplicaciones (supervisadas por personas en última instancia) sino por aplicaciones de forma autónoma, tomando decisiones e incluso elaborando nuevos recursos de información y ontologías.

2.8. Esquemas conceptuales en la Recuperación de Información

Los actuales sistemas de recuperación de información web son el resultado de una evolución continua de la propia Web desde su aparición. Los directorios web ofrecen

posibilidades de búsqueda de información alternativas mediante la navegación por categorías. Ciertos motores de búsqueda incluyen una clasificación de recursos seleccionados, lo cual permite a los usuarios sin experiencia encontrar recursos de información destacados. Otros motores, siguiendo la línea de Google, explotan la estructura de enlaces para la obtención de la relevancia de las páginas recuperadas. La utilidad de correo Gmail ofrece la posibilidad de gestionar etiquetas para clasificar los mensajes de correo del usuario para su posterior búsqueda y recuperación.

A los problemas propios de la recuperación de información, como relevancia, pertinencia, exhaustividad y selección de los términos de búsqueda, se unen otros, inherentes a la naturaleza de la Web. Algunos autores, como Martínez Méndez (2004), han realizado modelos para la evaluación de los principales motores de búsqueda Web, concluyendo que la dinámica de funcionamiento y resultados obtenidos en dichos sistemas son similares. Por tanto puede afirmarse que las dificultades a las que se enfrentan los motores de búsqueda también son comunes y que se han agravado progresivamente con la aparición de la Web 2.0.

En efecto, la calidad, estructuración y originalidad de contenidos no han evolucionado paralelamente a la facilidad de publicación en la Web. Existen gran cantidad de páginas duplicadas ya que muchos usuarios prefieren copiar contenidos en vez de referenciarlos con enlaces de hipertexto. Multitud de páginas hacen un uso incorrecto de metadatos HTML e incluso simplemente incluyen una lista con multitud de términos de todo tipo para confundir a los robots de indización. Tampoco es posible distinguir el tipo de referencia que se ha obtenido en el proceso de búsqueda, es decir, cuando se recupera un documento informativo, una ficha de una aplicación en un servicio de descarga de pago, una entrada en un foro de debate, etc. En este escenario los buscadores web son incapaces en ocasiones de ofrecer unos resultados útiles.

Por tanto, coincidimos con la observación realizada por Spivack (2005), según la cual disminuye el nivel de eficacia de la consultas en texto libre y por tanto de la productividad de las búsquedas. A lo anterior se añade la dificultad que supone trabajar con conjuntos de recursos heterogéneos en cuanto a su formato.

Así pues, podemos considerar que el incremento en el volumen de datos, información o conocimiento no implica el paso a un estadio superior de organización. El factor determinante en la evolución de un sistema de información es el grado de estructuración, formalización e integración de sus elementos (Schlögl, 2005). Además,

en la Web, existe otro considerando a tener en cuenta: la conectividad.

La comunicación resulta fundamental en los procesos de organización de datos, información o conocimiento (García Marco, 1995b) y por ende, los procesos de recuperación de información deben adaptarse a una nueva realidad, en la que las redes sociales y las comunidades de usuarios forman parte esencial del desarrollo de la Web. La conectividad debe acompañarse de una adecuada descripción de los recursos que utilizan los usuarios. La conexión de las personas no garantiza una mayor calidad de los contenidos y en ocasiones cuando no disponen de herramientas y recursos adecuados se realiza un uso de la Web ineficaz e incluso contraproducente para los procesos de recuperación de información.

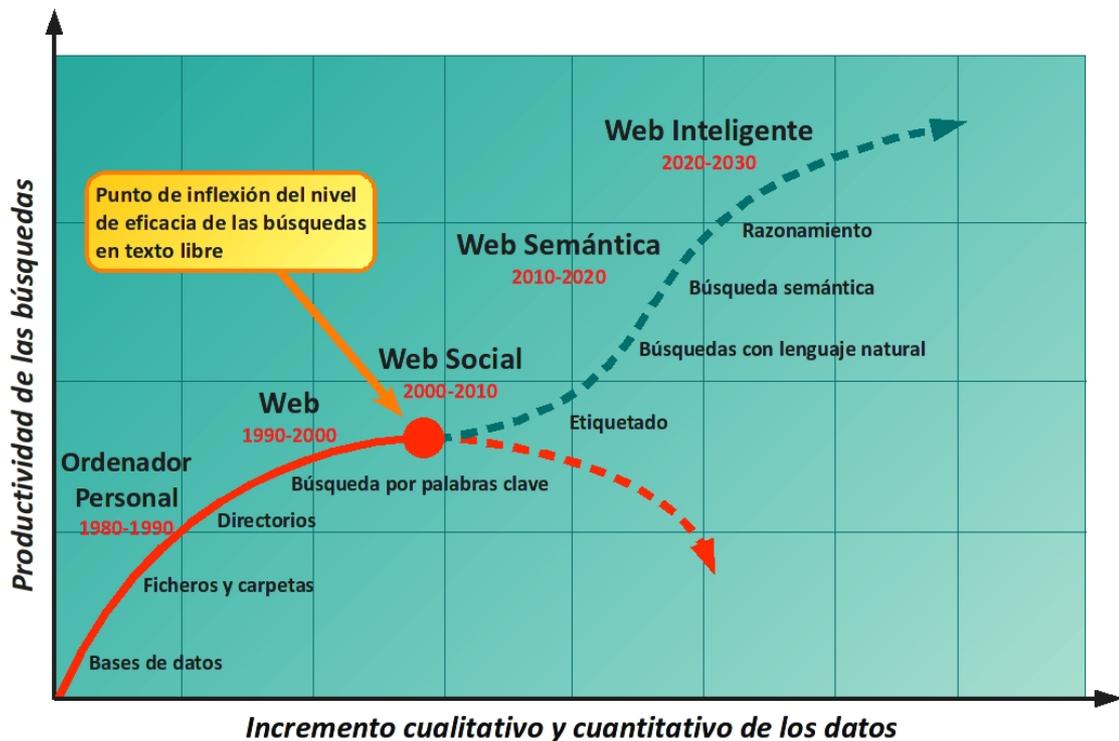


Ilustración 13: Eficacia de la búsqueda en texto libre en relación a la evolución de la Web. Traducido y adaptado de Spivack (2005)

Es evidente que la situación mejoraría con el uso de modelos de metadatos normalizados, como Dublin Core, para la descripción de recursos de información, junto con la aplicación de esquemas conceptuales. La indexación de páginas web con un

tesauro hace posible plantear consultas sin que el usuario tenga que realizar una selección predictiva de términos. Se amplían las posibilidades de búsqueda ya que el usuario consultaría una red de términos, los cuales podrían ser combinados con un nuevo lenguaje de consulta. Se trata de una búsqueda basada en la exploración de una red de conceptos, que guiaría al usuario en la selección de los términos de la consulta, siendo posible incrementar su eficacia estableciendo filtros según el contenido de ciertos metadatos. Esta evolución de los SRI se debe fundamentalmente al carácter dinámico de la Web desde el punto de vista tecnológico, de contenidos y de servicios ofertados (Savy y Picard, 2001).

Las nuevas tecnologías como los editores web o los gestores de contenidos y la aplicación de nuevos estándares como XML, XHTML y RDF permiten una mayor flexibilidad y sencillez en la creación de páginas Web. La facilidad para crear servicios web que hacen uso de sistemas de bases de datos, permite que cada servicio busque la manera de diferenciarse del resto con nuevas características que cada día apuntan más a la creación de redes sociales que permiten a los usuarios colaborar entre sí.

Por ello, pensamos que en un entorno tan dinámico como la Web asistimos a un proceso evolutivo en donde los nuevos SRI integrarán características que actualmente ofrecen de forma individual los diversos servicios de búsqueda de información.

Los metadatos son el elemento catalizador de la Web Semántica. Según han ido ganando en capacidad descriptiva y ampliando las posibilidades de estructuración e interrelación se han obtenido nuevas tecnologías que han permitido pasar de sistemas de recuperación de información basados en simples listas de datos hasta aplicaciones en donde se contemplan otros más complejos donde se aplican procesos lógicos para definir y ejecutar tareas cercanas al razonamiento (Davis, 2006).

Al mismo tiempo han evolucionando las herramientas y lenguajes usados para codificar dichos metadatos y adaptarse al incremento de las necesidades de estructuración y abstracción. El nivel más bajo de estas herramientas, que permiten el desarrollo de aplicaciones con funciones de interoperabilidad sintáctica, lo constituyen los vocabularios controlados y los glosarios que dan paso a una organización básica de datos con XML y al Modelo Relacional.

En un nivel posterior pueden desarrollarse operaciones orientadas a la interoperabilidad estructural basadas en el uso de esquemas de bases de datos, la

definición de entidades, relaciones y atributos con el Modelo Entidad-Relación y la utilización de Esquemas XML, así como los Topic Maps para el desarrollo de soluciones que giran en torno al concepto de Taxonomía. El siguiente nivel tiene como objetivo la consecución de la interoperabilidad semántica para el desarrollo de esquemas conceptuales a partir de modelos avanzados. Estos esquemas sirven como base para definir relaciones lógicas entre conceptos con la finalidad de realizar inferencias.

En cualquier caso, de un modo u otro, los SRI para la Web tienden a la aplicación de algún tipo de clasificación, tesoro o esquema conceptual ya sea como parte esencial del sistema o complementando grandes índices de documentos que explotan los motores de búsqueda. Estos esquemas conceptuales se aplicarían tanto en los procesos de clasificación de documentos como en los de búsqueda de información por parte del usuario. Permitirían expandir y redefinir búsquedas o complementar la extracción del contenido textual de los documentos con la indización de éstos por medio de tesauros o clasificaciones.

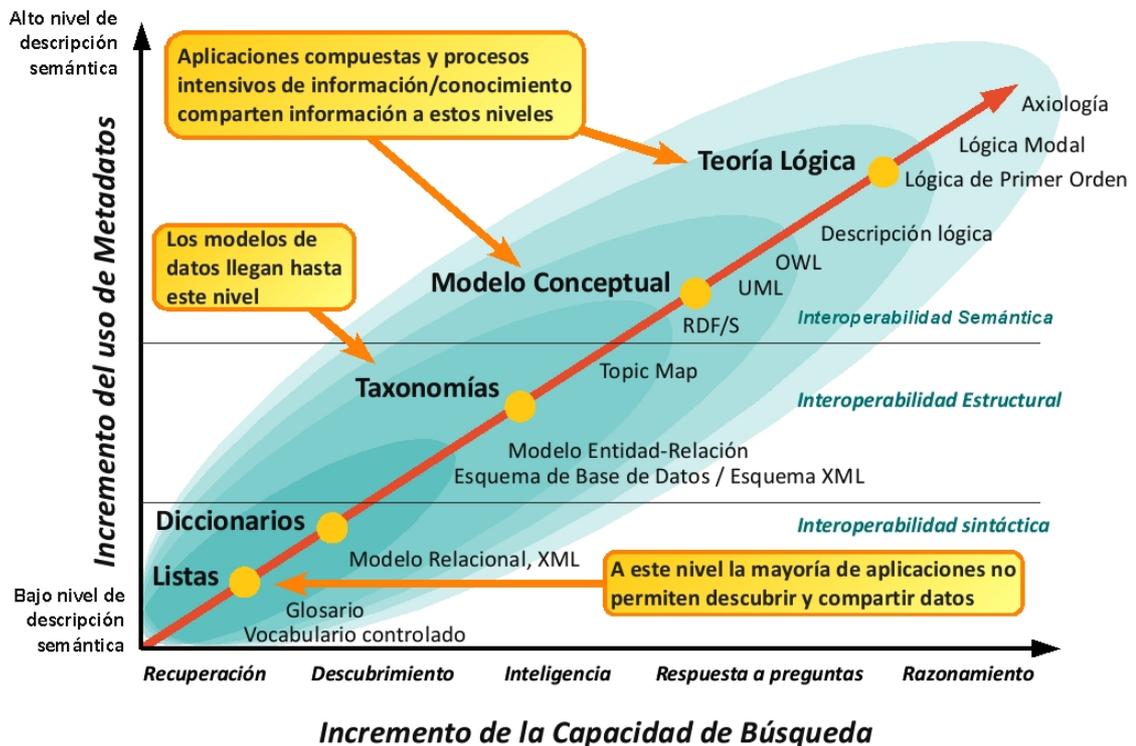


Ilustración 14: Relación entre el incremento de la capacidad de búsqueda y el uso de metadatos. Traducido y adaptado de Davis (2006)

Esta perspectiva se puede complementar con la visión de Miller (1997), para el que resulta imprescindible el uso de herramientas de este tipo, con la finalidad de realizar un control léxico. Para este autor el éxito de una búsqueda en texto libre implica la conveniente selección de términos. En ocasiones dicha selección se desarrolla en un proceso de “ensayo y error” por parte del usuario ya que desconoce las peculiaridades del lenguaje de consulta y el entorno de trabajo. Esta situación se agrava con las peculiaridades léxicas y terminológicas del propio lenguaje natural, tales como sinonimia, polisemia o variaciones lingüísticas entre otras. Resulta pues evidente, que la aplicación del tesoro, o algún otro tipo de esquema conceptual, como herramienta de control terminológico, incrementaría las posibilidades de éxito en las operaciones de búsqueda.

Otras aplicaciones de interés serían las de mostrar referencias de páginas cuyo contenido está relacionado con el de aquellas que se han recuperado directamente en la búsqueda o la de sugerir nuevos términos de consulta. Incluso sería posible desarrollar una interfaz donde el usuario no tenga que introducir palabras clave sino navegar por una red de conceptos y seleccionar aquellos que son de interés en su búsqueda.

Resulta interesante la reflexión de Pérez Agüera (2004) apuntando que, los tesauros, aunque se han integrado en entornos de gestión y recuperación de información documental, no siempre se han utilizado en el ámbito de la recuperación de información automática. En efecto, existen un gran número de sistemas que aplican únicamente cálculos estadísticos y de frecuencia de aparición de palabras. Siguiendo con el mismo autor, unas de las principales dificultades con las que se encuentran los motores de búsqueda es que los procesos de recuperación de información en Internet se producen de forma distribuida, con lo cual resulta de interés utilizar una herramienta compartida para la desambigüación y normalización a nivel semántico de los términos en los procesos de indización y la elaboración de las consultas.

Uno de los obstáculos que hasta ahora ha encontrado la aplicación de este tipo de esquemas conceptuales en los SRI es la representación de los mismos con un modelo adecuado, aplicando una tecnología que permita creación y mantenimiento sencillos, interoperabilidad, reutilización e incluso la integración de diferentes esquemas. Ya hemos visto anteriormente como en el ámbito de la Web Semántica se han desarrollado tecnologías capaces de realizar satisfacer estas necesidades,

fundamentalmente RDF y XML. Veamos a continuación las soluciones que aportan dichas tecnologías para la representación de esquemas conceptuales entre los que se encuentran los tesauros.

3

*Representación de tesauros
con SKOS*

3. Representación de Tesauros con SKOS

"La palabra es el índice del pensamiento" (Lucio Anneo Séneca)

Resumen: En este capítulo realizaremos un análisis de la evolución del concepto de tesauro seguida de una breve descripción de algunas iniciativas para la representación de tesauros usando vocabularios RDF y por ende para su aplicación en la Web Semántica. Este breve análisis nos permitirá conocer algunos puntos de interés de estas propuestas para plantear aquellos aspectos que debe contemplar un modelo de representación de tesauros en la Web Semántica. Finalmente se realizará un estudio detallado de los distintos elementos SKOS y de sus principales características.

3.1. El concepto de Tesauro

El concepto de tesauro ha seguido una evolución a lo largo del tiempo desde una simple lista de palabras relacionadas entre si por un concepto, hasta los actuales vocabularios controlados en donde los términos conforman estructuras complejas a través de relaciones semánticas. Aunque no es objeto de este trabajo una revisión exhaustiva de las diferentes definiciones existentes sobre el concepto de Tesauro, ni abordar el tema de los lenguajes documentales, creemos imprescindible exponer algunas de las definiciones y propuestas que, a nuestro parecer, son más relevantes para identificar las principales características y funciones de un tesauro.

El término Tesauro, procede de la voz latina **thesaurus** y del griego $\theta\eta\sigma\alpha\upsilon\rho\acute{o}\varsigma$. Dicho vocablo significa "tesoro" según la Real Academia³⁷, en cuyo diccionario encontramos una entrada con la siguiente definición: *"Nombre dado por sus autores a ciertos diccionarios, catálogos o antologías"*.

Siguiendo a Laureilhe (1977) fue Robert Etienne el primer autor que uso el término "Tesauro" en su diccionario de latín "Thesaurus linguae latinae" en el que ubicaba, de

³⁷ Definición del Diccionario de Lengua Española, 22ª edición. [En línea]
<http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=tesauro> (Consulta: 1 de septiembre de 2008)

forma ejemplificada, las entradas de palabras en su contexto semántico correspondiente. El nivel de detalle aumenta en la obra de Peter Mark Roget, que data de 1852, "*Thesaurus of English Words and Phrases*" en la que ofrece una herramienta lingüística para identificar un concepto concreto que agrupa una o varias palabras sinónimas.

Según Currás (2005) tenemos que remontarnos a finales de los años 50 para poder encontrar el origen del concepto actual que se tiene del tesoro en la gestión de información documental. El incremento de la productividad científica, y el consiguiente aumento en la actividad de edición y publicación de documentos, hizo necesario el desarrollo de sistemas de indización a partir de palabras clave. En aquel entonces Howerton hablaba de listas controladas para referirse a conceptos, entre los cuales se establecen relaciones heurísticas o intuitivas. No hay que olvidar que previamente, según Roberts (1984), Mooers ya había asociado los tesauros con los sistemas automatizados para la recuperación de información, mientras que Taube había sentado las bases de la postcoordinación y Luhn ya había abordado, a un nivel muy básico, la elaboración de tesauros con técnicas automáticas. Brownson (1957) usa por primera vez el término tesoro, dentro del campo de la Biblioteconomía y Documentación, haciendo mención a la problemática de traducir los conceptos y sus relaciones expresados en los documentos, a un lenguaje más preciso y carente de ambigüedades, para facilitar la recuperación de información. No sería hasta principios de los 60 cuando se publicaría el Tesoro ASTIA³⁸ con las características de los tesauros actuales. En este momento el concepto de tesoro evoluciona desde una herramienta lingüística hacia una herramienta de análisis y organización de información.

Es fácil deducir que en aquel momento ya se había detectado la necesidad de emplear una herramienta, que gestione un vocabulario controlado, en las tareas de indización para los procesos de análisis documental. Surge el concepto de lenguaje documental que intenta resolver los problemas asociados al uso del lenguaje natural (homonimia, polisemia, sinonimia) en sistemas de gestión de información. Básicamente, un lenguaje documental se compone de una lista de términos o vocabulario y de una serie de relaciones entre dichos términos establecidos a priori.

A este respecto García Gutiérrez (1984) nos ofrece una visión bastante precisa del significado que encierra la idea de Lenguaje documental: "*conjunto normalizado y*

38 ASTIA: Agencia de Servicios Técnicos de Información de la Armada de Estados Unidos.

normativo de términos relacionados por principios comunes, declarados portavoces preferenciales de los mensajes encerrados en un colectivo documental con el fin de provocar una recuperación pertinente de la información por aproximación temática. El lenguaje documental no se refiere, pues, a otros criterios utilizados en la búsqueda documental; autor del documento, lengua del texto, fecha de publicación, etc.”. Para este autor existen dos procesos claramente subyacentes a la propia existencia de un lenguaje documental: la recuperación de información y la representación del conocimiento de forma normalizada, partiendo generalmente de un corpus documental.

Gilchrist (1971) define un tesoro como *“una lista autorizada de léxicos, sin notación, que difiere de una lista de encabezamientos de materia, en el que las unidades léxicas, son más pequeñas y manejables y se aplican en la indización coordinada”*. Casi simultáneamente otro autor Wersig (1971) indica que los tesauros son *“listas de términos, prefijados con anterioridad, aunque sacados del texto de los propios documentos, y que replica los conceptos en unidades simples que se coordinan posteriormente para evitar ambigüedades. Entre ellas se establecen relaciones jerárquicas, asociativas y de equivalencia”*. Ambos autores coinciden en afirmar la simplicidad de los elementos del tesoro y su posterior coordinación en el proceso de indización, si bien Wersig hace especial hincapié en la existencia de relaciones semánticas entre dichas unidades. A lo largo de la década de los años 1970 se desarrollan actividades orientadas a la creación de normas sobre la estructura de tesauros (como la primeras normas ANSI e ISO), así como la aparición de una serie de tesauros entre los que cabe destacar la primera edición del *“Tesoro de la UNESCO”* en 1977.

En la década de los 80 se incrementa la labor de normalización. Ya en la segunda edición de la norma ISO 2788:1986 (ISO, 1986) sobre tesauros monolingües encontramos una definición en la que se considera a un tesoro como un *“vocabulario de un lenguaje controlado de indización, con una organización formal con la finalidad de explicitar las relaciones a priori entre conceptos”*. De esta misma época data la definición propuesta por Aitchinson y Gilchrist (1987), similar a la de la norma ISO pero en la que introducen expresamente la función del tesoro en los procesos de recuperación de información.

No hay que olvidar tampoco la definición de Slype (1991) para el que un tesoro se

trata de una *“Lista estructurada de conceptos destinados a representar de manera unívoca el contenido de los documentos y de las consultas dentro de un sistema documental determinado y a ayudar al usuario en la indización de los documentos y las consultas”*. Lancaster (1992) también participa de este puntos de vista ya que considera al tesauro como una herramienta para el control terminológico, resaltando su carácter instrumental orientado fundamentalmente a la recuperación de información. Una definición muy interesante es la que nos ofrece Currás (1998) en la que habla del Tesauro como *“un lenguaje especializado, normalizado, post-coordinado, usado con fines documentarios, donde los elementos lingüísticos que lo componen -términos, simples o compuestos-, se hallan relacionados entre sí sintáctica y semánticamente”*.

Al respecto de las relaciones semánticas, tal y como indican Jurafky y Martin (2000), se identifican las siguientes:

- Homonimia: Cuando dos unidades léxicas con origen etimológico diferente, y por lo tanto con significados distintos, se escriben y/o pronuncian de forma idéntica.
- Polisemia: Cuando una unidad léxica tiene diferentes significados dependiendo del contexto en el que se use.
- Sinonimia: Se refiere a unidades léxicas que se escriben y pronuncian de forma distintas pero se refieren a un mismo significado.
- Antonimia: Es la relación semántica existente entre dos palabras con significados contrarios.
- Hiperonimia: Relación que se establece entre unidades léxicas, referida al hecho de que el significado de una de ellas engloba al de la otra.
- Hiponimia: Es la relación inversa a la Hiperonimia que se produce cuando el significado de una unidad léxica está englobado por el de otra.
- Holonimia: Se trata de una relación similar a la de Hiperonimia, pero no se refiere tanto a una inclusión conceptual como a una inclusión material.
- Meronimia: Es la relación inversa de la Holonimia

Es de interés describir los elementos que componen la estructura de un tesauro según las normas ISO 2788:1986 (ISO, 1986) y ANSI/NISO Z39.19 (ANSI, 2005) ya que ofrecen un enfoque que ha perdurado hasta la actualidad. Para estas normas un tesauro se compone de dos tipos de unidades léxicas: términos preferentes y términos no preferentes. Los términos no preferentes se asocian a un único término preferente a través de una relación de reenvío (USE), mientras que la relación inversa denota los términos equivalentes (no descriptores) para un término preferente (UF, *Used For*). Para evitar ambigüedades en casos de homonimia, ambas normas introducen el uso de términos cualificadores entre paréntesis a continuación del término.

Entre los términos preferentes se establecen relaciones jerárquicas y asociativas. Las relaciones son similares en ambas normas. Se incluyen relaciones para denotar cuando un término es más genérico que otro (BT, *Broadern Term*) cuando un término es más específico que otro (NT, *Narrower Term*) o cuando dos términos se relacionan entre sí (RT, *Related Term*). La norma ANSI/NISO Z39.19, además de las relaciones jerárquicas genéricas, añade otras para expresar relaciones jerárquicas todo/parte y entre un sustantivo común y un nombre propio (instancia). La norma ISO solo contempla el caso de relaciones jerárquicas todo/parte, pero incluye una relación que no está presente en la norma ANSI/NISO, la relación entre un término y el término cabecera de la jerarquía (TT, *Top Term*). Asimismo, ambas normas incluyen la posibilidad de asociar, a los términos preferentes, notas de alcance para aclarar aspectos de uso, generalmente en los procesos de indización.

Hacemos notar que los elementos centrales del modelo de tesauro definido por las normas ISO 2788:1986 y ANSI/NISO Z39.19 son las unidades léxicas. Este modelo tiene una orientación principalmente lexicográfica. Por ello, es de gran interés la revisión que hace Miller (1997) con respecto a la naturaleza de los tesauros, en contraposición con la de los esquemas de clasificación. Hace evidente que la naturaleza funcional de estas herramientas, condiciona en gran medida su origen y las experiencias reales de aplicación, orientan su evolución en un sentido u otro. Para este mismo autor, en ocasiones, no es posible una separación entre tesauros y esquemas conceptuales, y no es posible considerar, en todos los casos, que el campo de aplicación del tesauro se ciña únicamente a la recuperación de información, dejando los aspectos de organización del conocimiento a los esquemas de clasificación. Esta reflexión es aún

más destacable, considerando que la recuperación de información en la Web Semántica tiende a combinar las técnicas estadísticas de recuperación de información y la representación de la información por medio de metadatos (Díaz Ortuño, 2003), organizados en estructuras definidas por esquemas conceptuales (García Jiménez, 2004).

A lo anterior se añade la observación efectuada por Pizzato (2003), en relación al empleo de técnicas heurísticas conjuntamente con un tesoro. El objetivo es la obtención de modelos destinados a refinar, ampliar o expandir consultas, tal como apuntaron Baeza-Yates y Ribeiro-Neto (1999). La finalidad de esta técnica es el consiguiente incremento de la efectividad en los procesos de recuperación de información. El usuario también interviene en este proceso, con independencia del uso de técnicas heurísticas, de forma que puede navegar por la estructura del tesoro, seleccionando términos para construir la consulta.

Al respecto no hay que olvidar el estudio realizado por Shiri, Revie y Chowdhury (2002) que analiza las estrategias de búsqueda por parte de los usuarios, demostrando que la expansión de consultas mediante tesauros y su integración en la recuperación de información es totalmente viable.

Finalizamos esta introducción al concepto de tesoro con una definición más extensa que las anteriores realizada por Arano (2005): *“Un tipo de lenguaje documental que representa la estructuración conceptual de un determinado campo del conocimiento. El tesoro proporciona una organización semántica principalmente a través de la explicitación de las relaciones establecidas entre dichos conceptos y, eventualmente, a través de un significado restringido de los términos que los representan. Los tesauros, para la Documentación y la Biblioteconomía, son por lo tanto una herramienta de control terminológico, puesto que la estructuración del tesoro es en base a conceptos. Los conceptos, son representados por términos seleccionados que evidencian ese control terminológico. Con dicho control se busca neutralizar la sinonimia y la polisemia, ambas características naturales de la lengua, que dificultan la precisión de la indización y recuperación de la información, ambas funciones básicas de los tesauros. Los tesauros son herramientas construidas para el auxilio tanto de los profesionales de la información como de los usuarios finales”.*

3.2. Características principales de los tesauros

Empleando como base el análisis elaborado en el apartado anterior, es posible ofrecer una relación de las principales características estructurales y funcionales de los tesauros. El propósito de dicha relación es elaborar más adelante, una propuesta de un modelo de representación de tesauros en la Web Semántica.

Entre las características estructurales podemos destacar las siguientes:

- Se trata de un vocabulario controlado y postcoordinado.
- Las unidades léxicas pueden ser términos preferentes (descriptores) o no preferentes (no descriptores).
- Los términos no preferentes se asocian siempre a un único término preferente. Por lo tanto el tesoro ofrece múltiples puntos de entrada hacia un mismo concepto etiquetado con un único término preferente.
- Es posible asociar a los términos preferentes notas de alcance, para explicar el ámbito de uso en las tareas de indización.
- El tesoro muestra las unidades léxicas a través Las unidades léxicas forman estructuras jerárquicas y asociativas a través del establecimiento de relaciones entre los términos preferentes.
- Existen diferentes tipologías de relaciones jerárquicas: genéricas, todo/parte y nombre común/instancia individual.

Desde el punto de vista funcional:

- Es un lenguaje documental, que utiliza un vocabulario controlado, ideado para solventar los problemas de ambigüedad del lenguaje natural en los procesos de indización y recuperación documental.
- Los tesauros han evolucionado desde herramientas de carácter esencialmente lexicográfico, hacia otras orientadas a la organización de información y la representación del contenido de un corpus de documentos, del que se vale, en muchos casos para la extracción de términos.

- Los términos preferentes se utilizan en los procesos de indización y selección de términos para la búsqueda de información. Por su parte los términos no preferentes aumentan la capacidad de previsión terminológica del tesauro, de forma que el usuario pueda utilizarlos para iniciar su búsqueda.
- Los tesauros Incrementan la capacidad de comunicación entre usuarios y el sistema de recuperación de información. Constituyen un nexo entre el lenguaje natural y la indización, como representación de la información utilizada en el sistema.
- Por regla general un tesauro general ofrece una perspectiva más cercana a la organización del conocimiento que la de un tesauro específico orientado hacia la representación del contenido de un corpus documental concreto.
- Durante la exploración de la estructura del tesauro, el usuario puede determinar, al mismo tiempo, los términos de búsqueda. Este proceso de selección se retroalimenta, ampliando, refinando o expandiendo los términos que componen la consulta.

3.3. La evolución del concepto de Tesauro

Resulta lógico que el concepto de Tesauro sufra un proceso de evolución a lo largo del tiempo. El incremento del volumen de información, la diversidad de formatos de la misma, los cambios en el modo de acceso a la información o las nuevas tecnologías de la información son algunas de las causas que han motivado dicha evolución. Los avances en equipamiento informático y la aparición de Internet hacen posible un incremento en la capacidad de almacenamiento y publicación de información, así como un acceso más rápido y mayor disponibilidad de la misma.

El soporte también han cambiado y se ha pasado del papel a diversos formatos digitales que permiten el archivo de grandes volúmenes de información multimedia. El modo en el que se accede a la información también ha evolucionado. Las clásicas estructuras de acceso secuencial, han dado paso al hipertexto, con el objetivo de ofrecer al lector una forma más dinámica para consultar y relacionar contenidos.

Las tecnologías de la información ya no se usan únicamente para controlar los documentos en papel. También se emplean para la elaboración de documentos cuyo

ciclo de vida no hace imprescindible la impresión, ya que los usuarios pueden consultarlos directamente en formato digital, como por ejemplo a través de la Web. En un principio las tecnologías de la información se utilizaban para construir grandes índices de documentos en papel. Hoy día los documentos ofimáticos y las bases de datos constituyen la principal fuente de información en muchos ámbitos de trabajo.

Vemos pues, que el entorno y las circunstancias, que dieron origen al tesauro, han cambiado en cierta medida. Las ediciones en papel realizadas para tesauros de diversos ámbitos temáticos ya no tienen una utilidad práctica inmediata. El rápido crecimiento del volumen de información en formatos digitales, hace inviable la aplicación de tesauros estáticos en sistemas de gestión y/o recuperación de información (Méndez, 2000). Resulta imprescindible que las herramientas de recuperación de información puedan adaptarse rápidamente a los contenidos.

Los tesauros, concebidos como instrumentos léxicos, comenzaron a principios de la década los años 90 a integrarse en los sistemas automatizados de recuperación de información. Los modelos de datos comenzaron a contemplarse para la representación de tesauros en entornos informáticos y desde el punto de vista de la creación y mantenimiento de tesauros, Rodríguez Muñoz (1990 y 1992) propuso un esquema entidad-relación, para la representación de tesauros, que permitía un rápido desarrollo utilizando sistemas de gestión de bases de datos relacionales (Jones, 1993). Los tesauros también fueron incluidos en distribuciones de bases de datos bibliográficas, como herramienta de ayuda en la selección de términos para diseñar las consultas de búsqueda. La relación entre hipertexto y tesauros no tardó mucho en ser identificada. Precisamente Rada (1991) propone un sistema en donde la información se elabora de forma colaborativa en un entorno hipertextual, al tiempo que se crea una red de enlaces a partir de los conceptos contenidos en los documentos. Dicha red de enlaces supone una estructura paralela a la de los vínculos que de forma explícita se insertan en el texto de los propios documentos de hipertexto.

También ha sido fructífera la idea de utilizar el hipertexto como marco de desarrollo para aplicaciones de gestión y consulta de tesauros, tal y como propusieron Pastor y Saorín (1993). La propuesta de estos mismos autores, cristalizó en lo que denominaron "Hipertexto Documental" (Pastor y Saorín, 1995a y 1995b). En estos trabajos daban forma a un modelo, donde la estructura paralela de enlaces propuesta por Rada se materializaba en una Red Semántica representada a través de un tesauro,

conformando un entorno de trabajo basado en el hipertexto y que posteriormente ampliarían hacia entornos personales de gestión integral de información documental (Pastor y Saorín 1998).

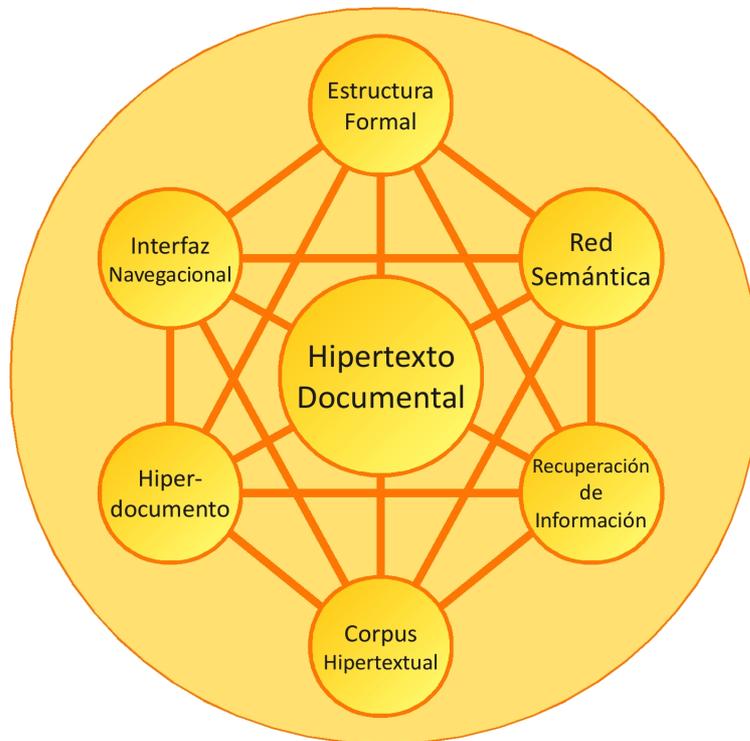


Ilustración 15: La geometría del Hipertexto Documental. Adaptado de Pastor y Saorín (1995b)

Al respecto de la aportación del hipertexto en los lenguajes documentales, y por ende a los tesauros, García Marco (1994) escribe: *“Creemos que los entornos de programación hipertexto pueden constituir plataformas muy adecuadas para desarrollar sistemas de construcción y gestión de lenguajes documentales, así como para generar interfaces capaces de facilitar su uso por parte del usuario”*. El mismo autor, señala que el hipertexto incrementa el grado de dinamismo y potencialidad en entornos de trabajos cooperativos de una forma amigable, dentro de procesos de procesamiento automático del conocimiento, al tiempo que abre nuevas líneas de trabajo para profundizar en cómo las estructuras de organización del conocimiento jerárquicas y asociativas pueden complementarse entre sí.

Kalervo (1996) elaboró un modelo deductivo para la expansión de consultas basadas

en conceptos, partiendo de tres niveles de abstracción (conceptual, lingüístico y de ocurrencias).

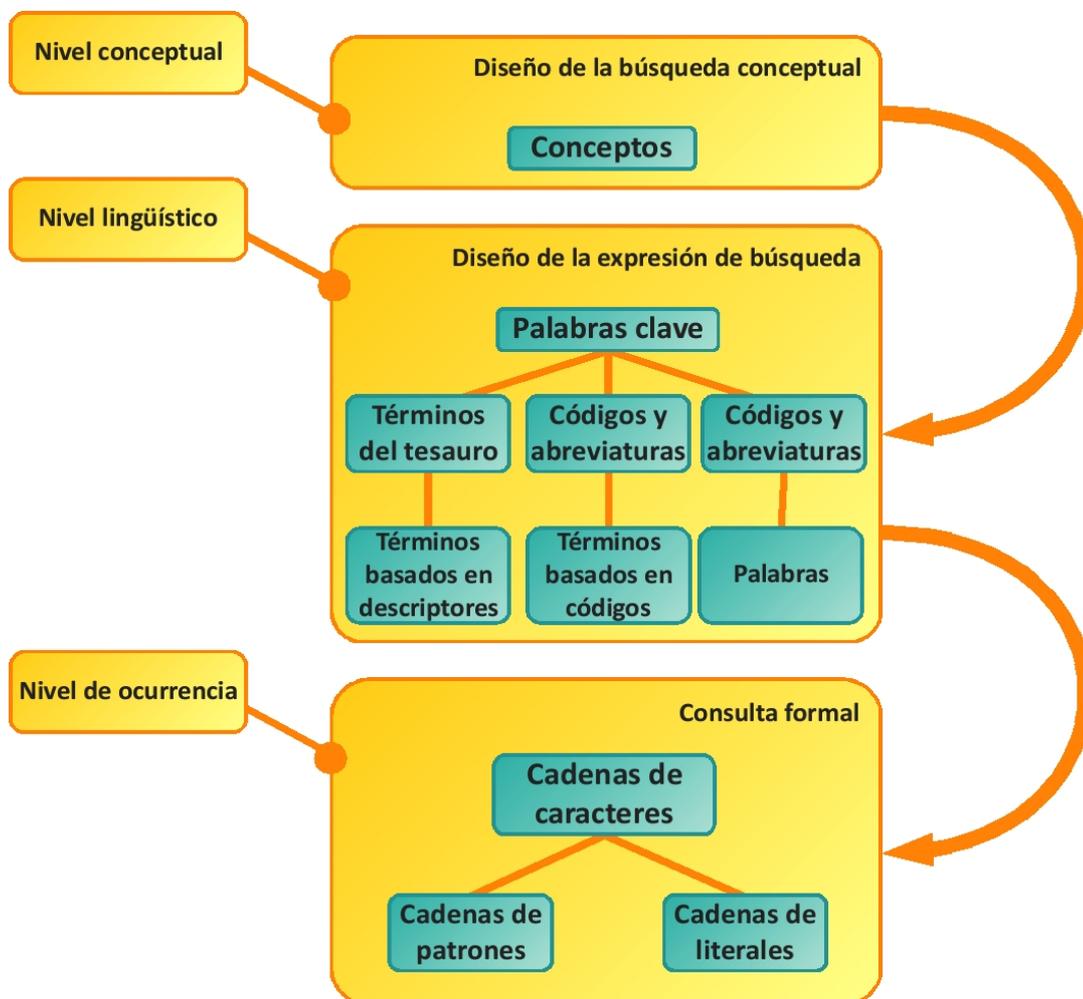


Ilustración 16: Niveles de abstracción del tesauro aplicados a la formulación de consultas. Adaptado y traducido de Kalervo (1996)

Los conceptos y sus relaciones se sitúan en el nivel conceptual. El nivel lingüístico representa los conceptos a través del lenguaje natural cuyas expresiones pueden tener varias representaciones en el nivel de ocurrencia. Nos encontramos ante una visión del tesauro donde prevalecen los aspectos conceptuales, antes que los léxicos, dotando a los tesauros de mayor flexibilidad y capacidad de adaptación. Esta afirmación coincide con los argumentos de López-Huertas (1999), apuntando que el tesauro encierra un gran potencial de evolución desde un recurso léxico en la recuperación de información, hacia un instrumento de gran capacidad para la representación conceptual,

convirtiéndose en una base de conocimiento experto para la recuperación de información.

Desde un punto de vista más general, algunos autores como Lopez-Huertas (1997) y de la Cueva Martín (1999) ven claramente los beneficios de las tecnologías de la información en el campo de los tesauros: mayor facilidad para crear y mantener tesauros, participación de los usuarios en el proceso de elaboración, integración en entornos de gestión de información, interoperabilidad de tesauros y por supuesto tal y como se ha apuntado anteriormente, aportar al tesoro una estructura de hipertexto para su consulta y uso.

Sin duda, la aparición de la Web marcó un antes y un después en la aplicación de los tesauros. Hasta aquel momento habían mostrado su eficacia en entornos controlados, generalmente dentro de bibliotecas, centros de documentación o bases de datos bibliográficas, e incluso en ciertos servicios ofrecidos a través de la Web. En estos entornos, la estructuras y rutas de acceso a la información están claramente definidas. Los flujos de información están diseñados y supervisados, así como los diversos procesos de almacenamiento, análisis, descripción, indización, búsqueda y recuperación. Generalmente la actualización de contenidos se efectúan mediante la adición de nuevos elementos y su ulterior procesamiento. La indización por medio de descriptores de un tesoro constituye un proceso lineal, desarrollado prácticamente en serie debido a la homogeneidad en la estructura de los contenidos.

En la Web la heterogeneidad en cuanto a formatos y estructuras no permite aplicar el mismo modelo de trabajo descrito anteriormente. Todo ello sin añadir el enorme crecimiento de recursos y contenidos desde la aparición de los sistemas de gestión de contenidos. Las rutas de acceso a los elementos informativos no están delimitados ni establecidos de antemano y la constante actualización de contenidos ha hecho prácticamente imposible una aplicación directa de los tesauros en la Web. El mantenimiento de índices globales, que reflejen la asignación de descriptores de un tesoro, es una labor totalmente imposible hoy día.

Junto con el volumen de información y su constante actualización, hay que resaltar la inexistencia hasta ahora de herramientas para la representación de tesauros, así como su gestión, uso e integración de forma distribuida. Algunas iniciativas, como los directorios de páginas web, se han rendido a la evidencia de que en la época de la Web 2.0, resulta prácticamente imposible realizar un control mediante la aplicación de un

tesauro, u otro tipo de herramienta que implique una indización manual. La solución adoptada ha sido la reducción de la estructura del directorio y/o la incorporación de sistemas de recuperación de información a través de motores de búsqueda.

Los tesauros conceptuales son herramientas más flexibles, cuya aplicación a la Web ofrecen una serie de ventajas que analizaremos más adelante. No obstante, en el momento de desarrollo y expansión de la Web, los tesauros se definían y elaboraban bajo el paradigma léxico, mucho más limitado que el conceptual y no adecuado para el diseño de procesos eficaces de recuperación de información que abarquen la globalidad de la Web. La ausencia de modelos y estándares para la representación de tesauros en aquel momento hacía posible la adopción de iniciativas.

La alternativa, tal y como se ha señalado anteriormente, es el uso de motores de búsqueda, que durante mucho tiempo demostraron ser eficaces. A pesar de que los tesauros se han adaptado a su publicación y consulta en entornos digitales de un modo bastante rápido (Laguens García, 2006), su aplicación real y eficaz en la Web aún deja bastante que desear. Esta presencia limitada de los tesauros en la Web se debe a una serie de dificultades entre las que destacamos las siguientes:

- Escaso desarrollo y delimitación de la orientación conceptual de los tesauros, prevaleciendo la visión léxica de los mismos. La aplicación de los tesauros léxicos a un entorno tan dinámico como la Web no obtenía resultados adecuados, en relación al esfuerzo realizado en los procesos de indización.
- Ausencia de modelos y estándares adecuados, para la representación de diferentes niveles de abstracción en la Web, como XML, RDF o SKOS. No hay que olvidar que XML vio la luz a principios de 1998.
- El acceso a las tecnologías de desarrollo Web (JAVA, PHP, MySQL) era costoso y en ocasiones ni se habían desarrollado. Esto hizo que los desarrollos se limitaran a entornos corporativos con una finalidad eminentemente pragmática, olvidando la necesidad de investigación, experimentación y estudio de dinámicas de recuperación de información por parte de los usuarios con herramientas alternativas.
- El rápido éxito de los motores de búsqueda, que en un primer momento se demostraron eficaces, utilizando técnicas automáticas de exploración de la

Web y construcción de índices.

- El hecho de que muchos autores consideraran al tesauro como una herramienta obsoleta, o de utilidad limitada a entornos cerrados o de gestión documental. Esto motivo a que comenzaran a estudiarse otras propuestas que aparentaban ser más adecuadas para la Web y su estructura hipertextual.

Al respecto del último punto, hay que destacar el trabajo de Shiri y Revie (2000) en el que apuestan por ampliar el concepto de tesauro, e incluso hacerlo permeable a otras herramientas. Incluso plantean nuevas vías de trabajo hacia posibles usos del tesauro, tales como la creación de sistemas de navegación, todo ello en un entorno en el que los metadatos son fundamentales para la reutilización e intercambio de tesauros. Por lo tanto el papel de los tesauros en entornos digitales debería ir más allá de un cambio de soporte, lo que conllevaría una auténtica redefinición de los planteamientos conceptuales de estas herramientas, tal como nos dan a entender Arano y Codina (2004) al afirmar que *“lo digital muchas veces pasa por cambiar solamente el soporte, en este caso concreto de los tesauros, pero no se ha generado una reflexión sobre los métodos de construcción de los mismos”*.

Tudhope (2001) abordó este problema planteando nuevas relaciones para los tesauros con el fin de incrementar las posibilidades de búsqueda, en una solución cercana a las ontologías. Por su parte, Arano (2005) ve en las ontologías nuevas posibilidades para el diseño de lenguajes documentales destinados a la recuperación de información. La ampliación de la tipología de relaciones permitiría desarrollos más adaptables y reutilizables, desde el punto de vista de diferentes dominios de conocimiento y a lo largo del tiempo. Algunos trabajos como los de la García Torres, Pareja Lora y Pradana López (2008) apuntan en este sentido, observando que las ontologías tienen una mayor capacidad expresiva para modelar sistemas de información que permitan enfrentarse a situaciones más complejas.

Otros autores, han optado por el análisis de herramientas que pudieran constituir una alternativa a la aplicación de los tesauros. El trabajo de Moreiro González et al (2004) analiza tres opciones para la organización y representación de la información: mapas conceptuales, tesauros y topic maps³⁹, apostando claramente por éstos últimos,

³⁹ Topic Maps es un modelo para la representación de información fundamentalmente a través de conceptos, ocurrencias y asociaciones. Se trata de un estándar definido por la norma ISO/IEC 13250:2003. Dispone de una sintaxis XML (XTM) y de un lenguaje de consulta (TMQL).

debido fundamentalmente al carácter extensible de las posibles relaciones de asociación entre conceptos. Otra de las ventajas de los topic maps es el origen del vocabulario, elaborado a partir de los propios documentos del sistema. No obstante, los últimos desarrollos alcanzados en la Web Semántica, más concretamente con RDF, han reavivado el concepto de tesauro en el ámbito de la recuperación de información para la Web.

Hemos hablado de las aportaciones del hipertexto a los tesauros. También hemos de tratar la nueva perspectiva que de la Web pueden ofrecer los tesauros. Es imprescindible, para ello, tener en cuenta que en realidad la Web no es una aplicación completa de todos los mecanismos de acceso a la información incluidos en la teoría del Hipertexto. La Web únicamente nos ofrece relaciones unidireccionales que nos conducen de un nodo a otro (páginas o bloques dentro de una misma página).

Las funcionalidades de enlaces multidireccionales, de expansión, sustitución, anotaciones o de otro tipo (Pastor y Saorín, 1995a), no están disponibles directamente en la Web a nivel conceptual, teniendo que ser diseñados a nivel de aplicación. Por ello, pensamos que los tesauros constituyen una herramienta de fácil aplicación y comprensión, por parte de profesionales y usuarios en los procesos de gestión y recuperación de información. Pueden dotar a los usuario de un nivel de abstracción que permita un acceso más sencillo a los contenidos que aquel que ofrecen la red de hipervínculos de la Web. Esta idea de "Hipertexto a dos niveles", propuesta por Mayfield (1995), constituye una línea de trabajo, en la que los tesauros pueden tener una clara aplicación para la recuperación de información en la Web. Este modelo puede complementarse con otro nivel donde se organiza la estructura de cada documento.

De esta forma, se obtendría un modelo en el que la Web se estructuraría en varios niveles: conceptual, navegacional y estructura documental. Los tesauros organizarían el nivel conceptual, constituyendo la solución empleada para los procedimientos de recuperación de información. Seguirían conservándose la navegabilidad que proporcionan los hipervínculos de las páginas Web consultas. Incluso en la estructura del documento, donde se refleja la organización de contenidos de una página, podrían aplicarse tesauros para indizar contenidos específicos, aunque tal vez lo más adecuado sea la aplicación de topic maps en este nivel.

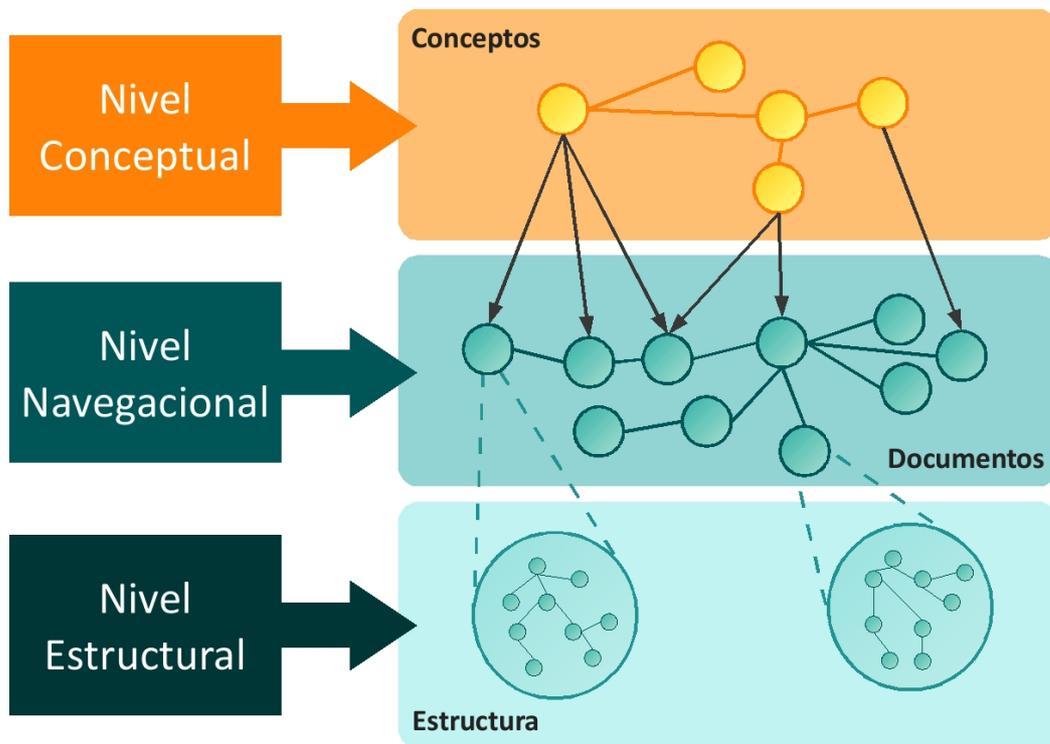


Ilustración 17: La Web estructurada en tres niveles: conceptual, navegacional y estructura documental

Resulta obvio que los tesauros léxicos no son adecuados para este cometido. Por su parte, las ontologías y los topic maps resultan excesivamente complejos, para su aplicación inmediata, de forma eficaz, en todos los niveles de la Web y a gran escala.

Por ello, pensamos que el futuro del Tesauro en la Web Semántica pasa, necesariamente, por la permeabilidad con otras propuestas (ontologías y topic maps) e incluso por el uso combinado de las mismas, que afiancen un cambio de paradigma en la elaboración de tesauros, dirigiéndolo hacia los tesauros conceptuales. También podrían comenzar a aplicarse en campos que, en un principio, aparentan ser opuestos al propio concepto de tesauro como herramienta de control terminológico, como son las Folksonomías (Moreiro González, 2006). Ésto proporcionaría mecanismos para la creación de vocabularios semi-controlados cuyos elementos se estructurarían para facilitar su consulta y uso en tareas de recuperación de información.

3.4. Antecedentes de representación de tesauros en la Web con RDF

Se han desarrollado trabajos para la representación de tesauros y esquemas conceptuales en formato XML que han tenido aplicación en organizaciones y ámbitos concretos. Algunos trabajos en este sentido como Zthes⁴⁰, MeSH⁴¹ o Topic Maps han tenido continuidad a lo largo del tiempo. Sin embargo, desde hace algunos años la tendencia es usar el modelo propuesto por RDF y su correspondiente codificación en XML para este fin. En realidad se trata de una evolución lógica en las líneas de trabajo dentro de la representación tesauros para la Web Semántica. El motivo fundamental de este cambio se debe a las grandes posibilidades de integración, reutilización y ampliación que ofrecen RDF y OWL en el ámbito de la Web Semántica. En efecto, cada vez más iniciativas adoptan el modelo que ofrece RDF para la representación de información en la Web y se utiliza XML para codificar sus modelos de representación de información ya que parte del potencial de este modelo consiste en la posibilidad de combinar diferentes especificaciones.

Es necesario puntualizar que el desarrollo de un vocabulario RDF/XML para la representación de tesauros implica superar el modelo que nos proponen las normas ISO 2788:1986 (ISO, 1985 y 1986) o ANSI/NISO Z39.19 (ANSI, 2005) sobre construcción y mantenimiento de tesauros. Estas normas describen un tesauro como un conjunto de términos de diverso tipo que entre los que se establecen relaciones semánticas. Pero la Web precisa de una visión más amplia que supere la idea de término como elemento central del tesauro y que amplíe el número, tipo y significado de las relaciones. Existen, por lo tanto, múltiples iniciativas dirigidas a la definición de vocabularios RDF para la representación de tesauros (W3C, 2001). Algunas siguen conservando como elemento principal la idea de término. Otras, en cambio, introducen la noción de concepto a los que se les asignan los términos correspondientes. Analizaremos, de un modo breve y descriptivo, algunas de estas iniciativas que servirán de referencia en la definición de un modelo empírico.

40 Zthes es un modelo para la representación de tesauros terminológicos descritos por las normas ISO 2788:1986 y ANSI/NISO Z39.19. Utiliza XML para representar los tesauros y está pensado para integrarse con el protocolo de recuperación de información Z39.50.

41 MeSH: *Medical Subject Heading*. Es un vocabulario controlado especializado en Medicina, Biología y Ciencias de la Vida. La Biblioteca Nacional de Medicina de EE.UU. creó y actualiza MeSH y desarrolló un vocabulario XML específico para su representación en entornos informáticos.

3.4.1. LIMBER

LIMBER (*Language Independent Metadata Browsing of European Resources*) se trata de una propuesta en el ámbito del proyecto del mismo nombre desarrollada por Miller y Matthews (2001). Su objetivo es el uso de RDF para la construcción de un sistema de información integral que utiliza un gestor de tesauros para la indización de recursos. Debido a que RDF está presente en la totalidad de elementos del sistema, también se desarrolló un vocabulario RDF para la representación de tesauros.

En LIMBER, los elementos de un tesauro son los conceptos, términos y notas de alcance. Se pueden establecer relaciones semánticas y de equivalencia entre conceptos y relaciones léxicas entre conceptos y términos. Las relaciones semánticas pueden ser jerárquicas, asociativas o indicar cuando un concepto constituye el inicio de la jerarquía (por lo que existe una subclase de conceptos para ello). Las relaciones de equivalencia permiten indicar cuando se trata de equivalencias exactas, inexactas, parciales o de uno a muchos. Existen varios tipos de notas de alcance y es posible el desarrollo de tesauros multilingües asociando un código de idioma tanto a conceptos como a notas de alcance.

3.4.2. ILRT

El ILRT (*Institute for Learning & Research Technology*) desarrolló en el año 2000 un vocabulario RDF con la finalidad de construir esquemas conceptuales que ayudaran al mantenimiento de redes de información para la investigación. Los elementos de este vocabulario son conceptos, términos, notas de alcance y tipo de uso de los términos. Las relaciones entre conceptos pueden ser asociativas o genéricas (las relaciones específicas son inferidas por la aplicación). Los términos, que pueden tener asignados un código de idioma, se clasifican en preferentes o no preferentes y se asocian a los conceptos.

3.4.3. CERES

CERES (*California Environmental Resources Evaluation System*) es un sistema de información desarrollado por la Agencia de Recursos de California para facilitar el acceso a recursos de información electrónicos sobre Medio Ambiente. De forma conjunta con el NBII (National Biological Information Infrastructure) se elaboró un tesauro para cuya publicación en Internet se diseñó una especificación RDF. La

especificación expresa bastante fielmente el concepto de Tesauro que reflejan las normas ISO y ANSI/NISO para su creación, mantenimiento y visualización. En este vocabulario se identifican términos que a su vez se dividen en categorías, descriptores y términos de entrada (no descriptores). Existen propiedades apropiadas para representar las relaciones de cada tipo de término. Por ejemplo, para un descriptor las propiedades disponibles son SN (para las notas de alcance), BT (para las relaciones genéricas), RT (para las relaciones asociativas), UF (para las relaciones con los no descriptores), etc.

3.4.4. GEM

Otro vocabulario RDF que también está basado en la estructura propuesta por las normas ISO y ANSI/NISO es el propuesto por *The GEM Consortium*. Este vocabulario creado en 2001 se limita a definir los distintos tipos de relaciones del tesauro en forma de propiedades. No se realiza definición de ningún otro tipo de elemento dejándola abierta con la posible empleo de otro vocabulario complementario.

3.4.5. CALL Thesaurus Ontology

Este vocabulario fué desarrollado por DRC (*Dynamics Research Corporation*) en el año 2002 dentro de un proyecto para la representación de un tesauro de más de 18.000 términos para el CALL (*Center for Army Lessons Learned, Fort Leavenworth, Kansas*). En realidad se trata de una ontología DAML definida con RDF. Los tesauros que aplican este vocabulario se estructuran en términos a los que se asocian propiedades que representan las relaciones semánticas. Las restricciones semánticas del tesauro se definen aprovechando las características de DAML

3.4.6. AGROVOC/KAON

A partir del modelo de ontología KAON desarrollada por el Centro de Investigación de Tecnologías de la Información de la Universidad de Karlsruhe en 2002 se realizó una aplicación para la representación del tesauro AGROVOC desarrollado por la FAO (Organización para la Alimentación y la Agricultura) y por la Comisión de las Comunidades Europeas. Cada término se modela como `rdfs:Class`, las relaciones jerárquicas se representan con la propiedad `rdfs:subClassOf`. Se definen propiedades para representar las relaciones asociativas, los términos preferentes y los

términos equivalentes. La etiqueta asociada a cada término es el otro elemento de este modelo que incluye propiedades para representar la cadena de texto de la etiqueta, el concepto al que se asocia y el idioma utilizado.

3.4.7. ETB: European Treasury Browser

ETB es un esquema RDF creado para el *European Treasury Browser Multilingual Educational Thesaurus*. Esta propuesta se basa en el concepto de nodos etiquetados, es decir, tanto los términos, como las relaciones, como el propio tesoro son nodos que se etiquetan de diferente forma para diferenciarse unos de otros. Aunque se trata de un tesoro multilingüe en realidad ETB trabaja con la suma de varios tesauros monolingües.

Los tipos de nodos definidos en ETB representan la totalidad del tesoro (*Thes*), cada uno de los tesauros monolingües (*Tmono*), campos semánticos (*MT*), nodos de términos (*MTNode*), nodos de equivalencia entre idiomas (*ENode*), términos no preferentes (*UNode*) y relaciones semánticas (*RNode*). En torno al nodo tesoro se organizan el resto de nodos correspondientes a cada uno de los tesauros monolingües. Cada tesoro monolingüe tiene asociados una serie de nodos de campos semánticos a partir de los cuales se van estructurando los nodos de términos y de relaciones semánticas

3.5. Modelo de representación de tesauros para la Web Semántica

Como hemos visto en el apartado anterior los vocabularios RDF propuestos para la representación de tesauros proponen distintas soluciones a un mismo problema, debido principalmente a las diferencias entre los modelos conceptuales propuestos por los autores. En dichas propuestas predominan mayoritariamente tesauros donde el elemento central son los términos. Esta visión clásica de un tesoro no es adecuada para la Web Semántica ya que conlleva una serie de dificultades en los procesos de mantenimiento del tesoro e indización de documentos.

A continuación desarrollaremos ciertos aspectos de desarrollo que implican una evolución del modelo conceptual del tesoro, aportando una aproximación más novedosa y adaptable a las posibilidades de recuperación de información que ofrece la Web Semántica.

3.5.1. Tesauros basados en conceptos y tesauros basados en términos

Tal vez sea éste el aspecto clave en el desarrollo de un modelo de representación de tesauros, siendo determinante en el correspondiente desarrollo del vocabulario RDF/XML que será adoptado para este propósito. Existen dos aproximaciones en la consideración del núcleo central del tesoro en torno al cual se estructuran el resto de elementos.

La primera de ellas es contemplar a un tesoro como un conjunto de términos que se vinculan entre sí con una red de relaciones semánticas de distinto tipo. Se trata de la representación clásica de las normas ISO y ANSI/NISO en las que se describe un tesoro como un conjunto de términos entre los que se establecen relaciones jerárquicas (término genérico y término genérico), asociativas (termino relacionado) y de equivalencia (término equivalente, término preferente). La segunda alternativa concibe al tesoro como un conjunto de conceptos entre los que se establecen relaciones semánticas similares a las que se establecen en los tesauros basados en términos. Los conceptos se vinculan con términos, a modo de etiquetas léxicas, entre los que pueden o no establecer relaciones léxicas.

Como se observa en la ilustración siguiente, en un tesoro basado en términos las relaciones semánticas se establecen entre los propios términos. Por contra, en un tesoro basado en conceptos se establecen ciertas relaciones semánticas entre conceptos y otras entre términos. Las relaciones entre conceptos suelen incluir las que desarrollan las estructuras jerárquica y asociativa, mientras por otro lado se denotan las relaciones existentes entre términos asociados a un mismo concepto (abreviaturas, uso común, tecnicismos, etc.)



Ilustración 18: Tesoro basado en términos frente a un tesoro basado en conceptos

Por lo tanto un tesoro basado en términos se estructura en un único nivel léxico-terminológico mientras que un tesoro basado en conceptos lo hace en tres niveles:

- Nivel conceptual en el que se identifican los conceptos y se establecen las relaciones entre los mismos.
- Nivel de correspondencia terminológica en el que se asocian los términos (preferentes o no preferentes) a sus respectivos conceptos.
- Nivel léxico donde se definen relaciones léxicas para vincular los términos entre sí.

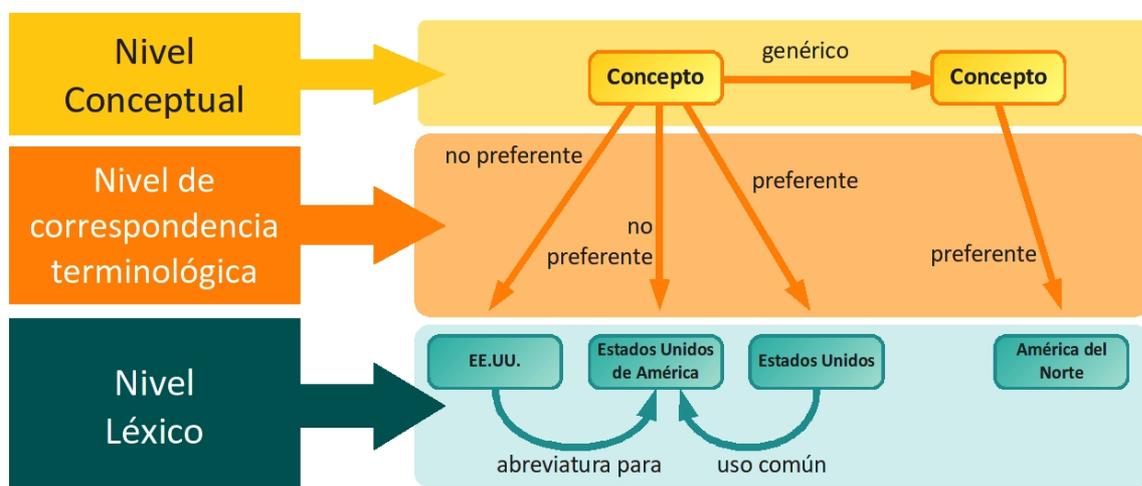


Ilustración 19: Niveles de organización de un tesoro basado en conceptos

Los tesauros basados en términos posibilitan la representación de esquemas más compactos y cercanos al modelo tradicional, pero los tesauros basados en conceptos permiten realizar descripciones más precisas de las estructuras de información, aunque en un principio su comprensión no resulte tan intuitiva como la de los tesauros basados en términos.

El cambio de paradigma en la modelización de tesauros supone un gran esfuerzo aunque dicho cambio se ve compensado con creces al optimizarse los procesos de elaboración, mantenimiento y productos derivados de la aplicación del tesoro en la indización de documentos. Los tesauros basados en términos conllevan problemas de aplicación en procesos de indización cuando se realizan operaciones de cambio de

términos preferentes y términos no preferentes. Así, cuando un descriptor con el que se ha indizado un documento pasa a ser un no descriptor hay que realizar una reorganización de los índices que asocian dicho descriptor con los documentos.

Esto no sucede cuando con los tesauros conceptuales ya que los procesos de indización son independientes de las alteraciones terminológicas del tesauro. En este caso los documentos se asocian a conceptos y no a términos; las alteraciones con respecto a los términos preferentes y no preferentes no influyen en la indización.

3.5.2. Estructuras de agrupación

Es ocasiones resulta de interés crear grupos o colecciones de conceptos, que tienen como objetivo crear familias de conceptos para aportar un mayor nivel de contenido semántico al tesauro o definir criterios complementarios para organizarlos.

La agrupación de conceptos en campos semánticos posibilita su acotación a partir de áreas temáticas que supongan un punto de acceso para la consulta del tesauro. Los grupos de conceptos pueden ser etiquetados con expresiones que denotan el significado de estas estructuras.

Otra posibilidad es la de agrupar los conceptos en facetas, lo cual permitiría caracterizarlos según múltiples criterios y al mismo tiempo mantener estructuras jerárquicas y asociativas independientes.

En cualquier caso, es necesario tener en cuenta que estas agrupaciones no constituyen elementos con la suficiente entidad individual como para establecer relaciones semánticas con conceptos o con otras estructuras de agrupación. Esto se debe a la diferencia de granularidad entre ambos elementos por lo que el establecimiento de relaciones semánticas entre los mismos podría introducir distorsiones en la estructura del tesauro.

3.5.3. Relaciones léxicas entre etiquetas

Las etiquetas y las relaciones léxicas entre las mismas son los elementos que conforman el nivel léxico de un tesauro. Aunque la visión clásica de tesauro no contempla las relaciones léxicas se trata de un elemento de gran interés debido a que aporta información de valor añadido.

El establecimiento de relaciones léxicas permitirá llevar a cabo una descripción más precisa del conocimiento que se desea representar sobre el dominio de tesauro. Esta precisión permitirá realizar ciertas operaciones de forma automática, como por ejemplo la desambiguación de términos introducidos por los usuarios en las consultas durante los procesos de búsqueda.

3.5.4. Relaciones como arcos o como nodos

Las relaciones semánticas o léxicas pueden representarse en forma de arcos o de nodos. La solución basada en arcos es mucho más compacta e intuitiva que su alternativa basada en nodos. Por contra, las relaciones basadas en nodos conllevan una dificultad añadida durante el desarrollo y el mantenimiento de este tipo de tesauros ya que la tareas de creación e interpretación de los grafos basados en esta aproximación suelen ser complejas y confusas.

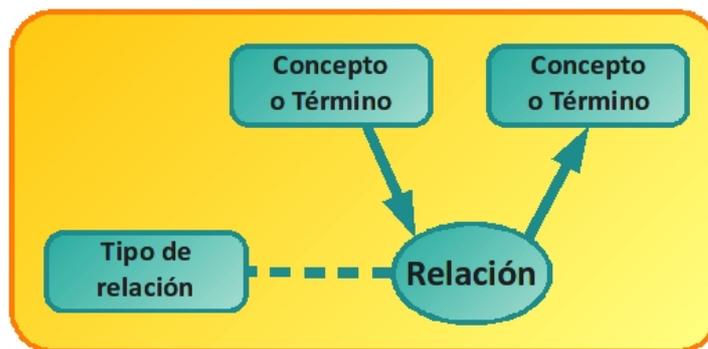


Ilustración 20: Relación expresada en forma de nodo

No obstante, las relaciones expresadas en arcos tienen la ventaja de que pueden referenciarse mediante URIs. Este tipo de relaciones pueden aplicarse tanto a las que se establecen a nivel conceptual o a nivel léxico. Algunos vocabularios RDF para la representación de tesauros aplican relaciones como nodos, tal es el caso de ETB.

3.5.5. Estructuras multilingües

Teniendo en cuenta el ámbito de aplicación de los tesauros expresados con RDF (la Web Semántica) resulta imprescindible contemplar mecanismos que permitan expresar la realidad multilingüe de los mismos. Por ello es perfectamente factible

asociar a un mismo concepto diferentes etiquetas preferentes para cada idioma.

Otra ventaja de los tesauros basados en conceptos es la posibilidad de establecer relaciones léxicas adaptadas a la realidad terminológica de cada idioma.

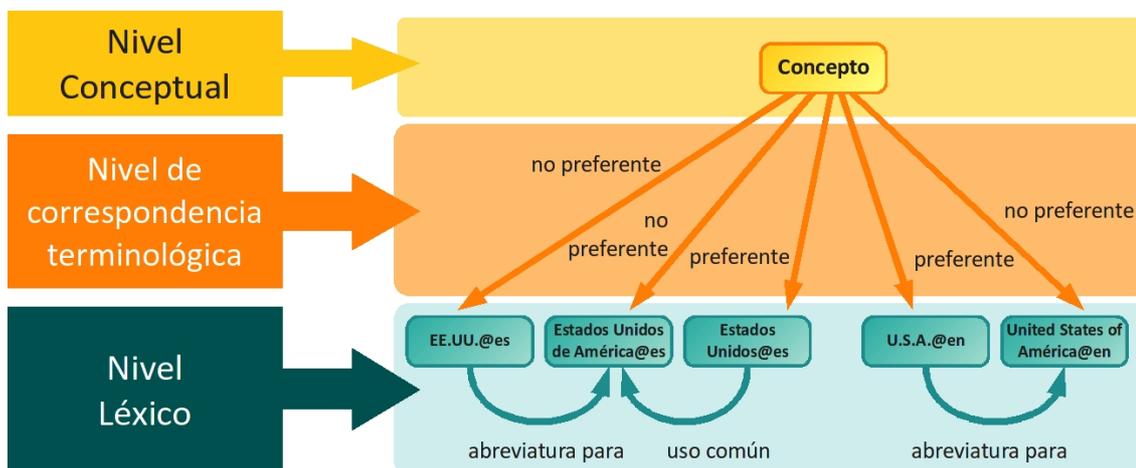


Ilustración 21: Independencia del nivel conceptual y adaptación del nivel léxico en estructuras multilingües

En la figura anterior podemos ver cómo el nivel conceptual no interviene en las estructuras multilingües, dejando esta función para el nivel léxico. El idioma de cada término se expresa mediante “@es” (Español) o “@en” (Inglés). Las relaciones léxicas son distintas para cada idioma; en el caso del Español se considera como preferente el término de uso común mientras que en el caso del Inglés el término abreviado se considera preferente.

3.5.6. Restricciones semánticas

Las restricciones semánticas controlan aspectos asociados a la coherencia estructural del tesoro. Algunas de estas restricciones tienen carácter opcional y otras en cambio son imprescindibles para asegurar un desarrollo correcto del proceso de construcción de un tesoro. Las principales restricciones semánticas son las siguientes:

- Control de estructuras polijerárquicas: Esta restricción es opcional y tiene como finalidad establecer si es posible que un concepto pueda tener dos o más conceptos genéricos.

- Integridad del nivel correspondencia terminológica: Controla que un concepto solo tenga asociado un término o etiqueta preferente.
- Inferencia de las propiedades simétrica e inversa: Esta restricción semántica establece que es necesario completar todas las relaciones con sus correspondientes relaciones simétricas o inversas. Es decir, cuando se establece una relación genérica desde un concepto A a otro concepto B es necesario establecer la relación específica en sentido inverso. Por su parte, las relaciones asociativas tienen carácter simétrico por lo que al establecer una relación de este tipo entre dos conceptos hay que indicarlo en ambos sentidos.
- Control de conceptos cabecera: No será posible definir un concepto como cabecera de un tesauro si se ha definido sobre dicho concepto algún tipo de relación jerárquica genérica.
- Control de ciclos en estructuras jerárquicas: Durante el establecimiento de relaciones genéricas o específicas es necesario evitar la creación de ciclos que puedan dificultar la consulta del tesauro. Un ciclo en una estructura jerárquica se produce al crear una relación específica desde un concepto A a otro concepto B al tiempo que el concepto A se encuentra en la misma estructura jerárquica en un nivel superior.
- Control de la propiedad disyuntiva entre relaciones asociativas y jerárquicas: Aunque algunos tesauros no hacen uso de esta restricción, pero resulta más coherente impedir el establecimiento de relaciones asociativas entre conceptos que se encuentran dentro de una misma estructura jerárquica.
- Control de las relaciones de correspondencia: Las relaciones de correspondencia podrán aplicarse únicamente a conceptos que pertenezca a diferentes esquemas conceptuales.
- Control de la propiedad reflexiva entre conceptos: Esta restricción semántica controla que un concepto no pueda establecer consigo mismo relaciones semánticas de ningún tipo.

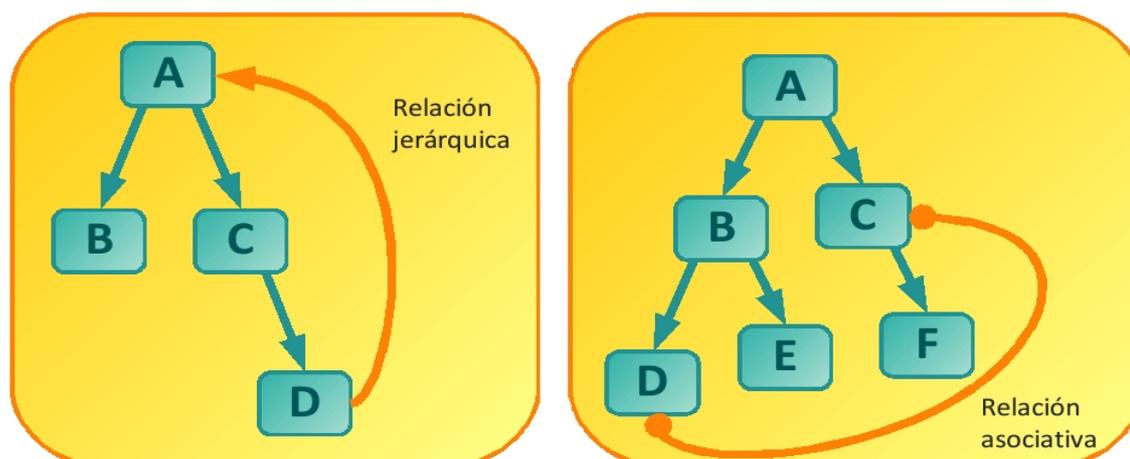


Ilustración 22: Ejemplos de relaciones semánticas incoherentes⁴²

3.5.7. Ampliación de la tipología de relaciones

Una de las ventajas de aplicar RDF en la representación de tesauros es la posibilidad de ampliar la tipología de relaciones con OWL o simplemente con RDF/RDFS. De esta forma sería posible crear una nueva relación definiendo algunas de sus propiedades como la transitividad, simetría, relación inversa o reflexividad declarando con rigurosidad las restricciones semánticas necesarias.

Esta incrementaría las posibilidades de éxito de la aplicación de los tesauros para la recuperación de información en la Web. Además supone una importante sinergia en los aspectos asociados a la evolución y adaptación del concepto de tesoro. Las relaciones jerárquicas y asociativa pueden verse ampliadas o sus características modificadas en un futuro, o bien personalizarse para proyectos concretos.

3.5.8. Equivalencia entre tesauros

Resulta de gran utilidad la posibilidad de establecer equivalencias entre conceptos pertenecientes a distintos tesauros. Es necesario definir varios tipos de equivalencia ya que la correspondencia entre conceptos no siempre es exacta, sino que pueden implicar conceptos más específicos o genéricos. También es posible establecer

⁴² . En el primer grafo se produce un ciclo dentro de la estructura jerárquica; en el segundo se establece una relación asociativa entre conceptos de una misma estructura jerárquica (la flecha indica el sentido de la relación jerárquica).

equivalencias para asociar ciertos aspectos de dos conceptos sin que suponga necesariamente que un concepto sea más genérico o específico que otro. Estas relaciones serían similares a las que se establecen entre conceptos pertenecientes a un mismo tesoro.

Este mapeado permite la reutilización o integración de tesauros externos dentro de Sistemas de Información. La correspondencia entre conceptos permite transformar consultas de un sistema a otro o complementarlas, ya que sería posible utilizar consultar repositorios de información indizados con un esquema conceptual diferente al utilizado en primera instancia por el usuario.

3.5.9. Definiciones, explicaciones, notas de alcance y metadatos

Las clásicas notas de alcance resultan muy limitadas para expresar información de valor añadido de gran utilidad para el uso de tesauros en la Web Semántica. Algunos de estos contenidos pueden ser definiciones, notas de diverso tipo, información sobre los cambios en la estructura del tesoro o en el significado de términos, ejemplos de uso, etc. Esta información, generalmente no incluida en tesauros editados en papel, podría ser consultada de un modo sencillo y ágil en un sistema de información desarrollado en entorno Web.

Junto con los contenidos indicados anteriormente se pueden añadir metadatos usando Dublin Core para incluir datos sobre la autoría y edición asociados a conceptos, etiquetas y relaciones.

3.6. SKOS

3.6.1. Introducción

SKOS (siglas de *Simple Knowledge Organization System*)⁴³ es una iniciativa del W3C en forma de aplicación de RDF que proporciona un modelo para representar la estructura básica y el contenido de esquemas conceptuales como listas encabezamientos de materia, taxonomías, esquemas de clasificación, tesauros y cualquier tipo de vocabulario controlado. El origen del proyecto se remonta a la elaboración de un tesoro de actividades dentro del proyecto SWAD-Europe⁴⁴. La primera versión de SKOS Core se presentó en el año 2003.

En SKOS los conceptos se identifican con referencias URI. Estos conceptos pueden etiquetarse en cadenas de texto en uno o varios idiomas, documentarse y estructurarse a través de relaciones semánticas de diversa tipología. Este modelo permite mapear conceptos de diferentes esquemas, así como definir colecciones ordenadas y agrupaciones de conceptos. También permite establecer relaciones entre las etiquetas asociadas a los conceptos.

El uso de RDF en el desarrollo de SKOS permite obtener documentos en un formato que permita su lectura por parte de aplicaciones informáticas, así como su intercambio y su publicación en la Web. SKOS se ha diseñado para crear nuevos sistemas de organización o migrar los ya existentes adaptándolos a su uso en la Web Semántica de forma fácil y rápida. Proporciona un vocabulario muy sencillo y un modelo intuitivo que puede ser utilizado conjuntamente con OWL o de forma independiente. Por todo ello, SKOS se considera como un paso intermedio, un puente entre el caos resultante del bajo nivel de estructuración de la Web actual y el riguroso formalismo descriptivo de las ontologías definidas con OWL.

SKOS está aún en desarrollo y los documentos disponibles aún son borradores de trabajo. Aunque en los años 2007 y 2008 se han elaborado algunos borradores de

43 Puede encontrarse más información sobre SKOS en <http://www.w3.org/2004/02/skos/>

44 SWAD-Europe: *Semantic Web Advanced Development for Europe*. Se trata de un proyecto ya finalizado que fue patrocinado por el W3C y orientado a la creación de tecnologías aplicadas a la Web Semántica mediante actividades de investigación con la consiguiente publicación de resultados. Este proyecto fue dirigido por cinco socios entre los que destacamos la Universidad de Bristol o los Laboratorios Hewlett-Packard.

trabajo que vienen a sustituir a otros publicados en el año 2005, el W3C ha considerado de gran valor seguir conservando todos los documentos anteriores.

Esta decisión es lógica si pensamos que el desarrollo de SKOS se encuentra en una fase inicial y resulta de gran utilidad mantener publicados estos documentos mientras que no se vaya completando los borradores de trabajo del 2007/2008 y sus contenidos puedan reemplazar totalmente los documentos previos. Los documentos de trabajo actuales que ofrece el W3C sobre SKOS son:

- Manual de Introducción a SKOS (*SKOS Primer*). Se trata de una guía básica de implementación. Realiza una descripción de los principales elementos de SKOS y algunos aspectos de ciertas posibilidades avanzadas de SKOS (W3C, 2008a2).
- Guía de referencia de SKOS (*SKOS Reference*). Es una guía completa de referencia del vocabulario de SKOS. Está dirigida a personas con experiencia en el uso de RDF y OWL (W3C, 2008a1).
- Casos de uso y peticiones de SKOS (*SKOS Use Cases and Requirements*). Se trata de una lista de casos de uso y recopilación de peticiones. Este documento se ha elaborado a partir de la cumplimentación de un cuestionario contestado por parte de personas e instituciones que han aplicado SKOS (W3C, 2007).

Si bien SKOS está en una fase inicial de desarrollo, el núcleo fundamental de trabajo ya ha sido establecido, así como su vocabulario básico.

3.6.2. Modelo SKOS

El modelo de datos SKOS es en realidad una ontología definida con OWL Full. Obviamente, al estar basado en RDF, SKOS estructura los datos en forma de tripletas que pueden ser codificadas en cualquier sintaxis válida para RDF. SKOS puede ser utilizado conjuntamente con OWL para expresar formalmente estructuras de conocimiento sobre un dominio concreto ya que SKOS no puede realizar esta función al no tratarse de un lenguaje para la representación de conocimiento formal.

El conocimiento descrito de manera explícita como una ontología formal se expresa como un conjunto de axiomas y hechos. Pero un tesoro o cualquier tipo de esquema de clasificación no incluye este tipo de afirmaciones, sino que identifica y describe (con

el lenguaje natural o expresiones no formales) ideas o significados a los que nos referimos como conceptos. Estos conceptos pueden organizarse en estructuras que carecen de una semántica formal y que no pueden considerarse como axiomas o hechos. Es decir, un tesauro únicamente proporciona un mapa intuitivo de como están organizados los temas dentro de procesos de clasificación y búsqueda de objetos (generalmente documentos) relevantes a un dominio específico.

Para convertir un tesauro o esquema de clasificación en conocimiento formal, debe transformarse en una ontología, un proceso que resulta muy costoso. En efecto, transformar la estructura de un tesauro en una ontología OWL conlleva un gran esfuerzo ya que una ontología no proporciona un modelo de datos que se pueda aplicar fácilmente. Esto sucede porque los tesauros se han desarrollado sin una semántica formal, fundamentalmente como herramientas que ayudan en la navegación o en la recuperación de información.



Ilustración 23: Grafo correspondiente a la codificación del ejemplo anterior mediante SKOS XL y su correspondiente grafo en el que no se establecen relaciones entre entidades léxicas.

No obstante, resulta factible aplicar OWL para construir un modelo de datos (en este caso concreto SKOS) que sea apropiado al nivel de formalización exigido por un tesoro. De esta forma, los conceptos de un tesoro se modelan como entidades en el modelo de datos SKOS y las relaciones entre conceptos como hechos sobre dichas entidades.

Los elementos del modelo SKOS son esencialmente clases y propiedades. La estructura e integridad del modelo de datos están definidas por las características lógicas y por las relaciones entre dichas clases y propiedades. Para SKOS, un sistema de organización del conocimiento se expresa en términos de conceptos que se estructuran en relaciones para conformar esquemas de conceptos. Tanto los conceptos como los esquemas de conceptos se identifican mediante URIs.

Los conceptos pueden ser etiquetados en cualquier idioma. Un concepto puede tener asociadas múltiples etiquetas, pero sólo una de ellas por cada idioma puede asociarse como etiqueta preferente. El resto de etiquetas asociadas al concepto se denominan etiquetas alternativas. También pueden definirse etiquetas ocultas con la finalidad de asignar a un concepto etiquetas que solo serían aplicables en los procesos de búsqueda e indización sin que sean visibles para los usuarios.

Es posible asignar a los conceptos códigos de clasificación o de identificación dentro de un esquema conceptual determinado. Estas notaciones no están expresadas en lenguaje natural sino en forma de códigos nemotécnicos o similares. Los conceptos también pueden ser documentados con notas de diferente naturaleza como definiciones, notas de alcance o notas de edición entre otras.

El modelo SKOS contempla el establecimiento de enlaces entre conceptos denominados relaciones semánticas. Estas relaciones pueden ser jerárquicas o asociativas, contemplándose la posibilidad de ampliar la tipología de relaciones. Los conceptos también pueden agruparse en colecciones que a su vez pueden etiquetarse y ordenarse. SKOS se complementa con la posibilidad de que conceptos de diferentes esquemas se pueden mapear entre sí empleando relaciones jerárquicas, asociativas o de equivalencia exacta.

3.6.3. Conceptos y esquemas de conceptos

Un concepto representa una idea, una noción o una unidad de pensamiento. Es necesario dotar de cierta flexibilidad a SKOS ya que tiene como finalidad la representación de un sistema semiformal de organización de conocimiento. Por tanto, hacemos notar que se trata de una entidad abstracta independiente del término o términos que se puedan utilizar para etiquetarlo. Los conceptos se asocian a una URI o a un identificador RDF para su reutilización y referencia. Los conceptos pueden asociarse a esquemas conceptuales. Un esquema conceptual es la agregación de uno o más conceptos SKOS y generalmente se usan para representar e identificar a los tesauros o esquemas de clasificación. Un esquema de conceptos puede tener uno o varios conceptos cabecera (*Top Concept*) que son aquellos conceptos que encabezan las estructuras jerárquicas dentro del propio esquema de conceptos. Para los usuarios suelen constituir los puntos de inicio de las tareas de búsqueda y navegación.

Elemento	Vocabulario	Definiciones de clases y propiedades
Concepto	skos:Concept	■ Instancia de owl:Class
Esquema de conceptos	skos:ConceptScheme	■ Instancia de owl:Class ■ Es disyunto de skos:Concept
Asociación a un esquema de conceptos	skos:inScheme	■ Instancia de owl:ObjectProperty ■ Dominio skos:Concept ■ Rango de valores skos:ConceptScheme
Concepto cabecera	skos:hasTopConcept	■ Instancia de owl:ObjectProperty ■ Dominio skos:ConceptScheme ■ Rango de valores skos:Concept

Tabla: 7: Clases y propiedades para conceptos, esquemas, pertenencia a esquemas y conceptos cabecera.

Según el vocabulario de SKOS los conceptos se definen con `skos:Concept` y los esquemas con `skos:ConceptScheme`. Para asociar un concepto a un esquema se utiliza la propiedad `skos:inScheme`. La propiedad `skos:hasTopConcept` se usa con los esquemas de conceptos para indicar los conceptos cabecera. A continuación se incluyen dos ejemplos. En el primero se muestra una definición básica de varios conceptos y su asociación a un esquema de conceptos. El segundo contiene la definición de un esquema de conceptos y sus conceptos cabecera.

3. Representación de tesauros con SKOS

```
1. <rdf:RDF xmlns:skos="http://www.w3.org/2004/02/skos/core">
2.   <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#america">
3.     <skos:inScheme rdf:resource="http://www.ejemplo.org/esquema"/>
4.   </skos:Concept>
5.   <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#asia">
6.     <skos:inScheme rdf:resource="http://www.ejemplo.org/esquema"/>
7.   </skos:Concept>
8.   <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#americanorte">
9.     <skos:inScheme rdf:resource="http://www.ejemplo.org/esquema"/>
10.  </skos:Concept>
11. </rdf:RDF>
```

Ejemplo 6: Definición de conceptos y asociaciones a un esquema de conceptos

```
1. <rdf:RDF xmlns:skos="http://www.w3.org/2004/02/skos/core">
2.   <skos:ConceptScheme rdf:about="http://www.ejemplo.org/esquema">
3.     <skos:hasTopConcept rdf:resource="http://www.ejemplo.org/conceptos#america"/>
4.     <skos:hasTopConcept rdf:resource="http://www.ejemplo.org/conceptos#asia"/>
5.   </skos:ConceptScheme>
6. </rdf:RDF>
```

Ejemplo 7: Definición de un esquema de conceptos y de conceptos cabecera

3.6.4. Etiquetas léxicas

Por etiqueta léxica se entiende a una cadena de caracteres que representa una expresión en lenguaje natural. SKOS ofrece etiquetas de distinto tipo: preferentes, alternativas y ocultas. Las etiquetas preferentes pueden asociarse a conceptos para representar los términos usados como descriptores en los sistemas de indización, tesauros o clasificaciones. Solo puede asociarse una etiqueta preferente en cada idioma a un concepto. Es recomendable que no existan etiquetas léxicas idénticas para representar conceptos diferentes.

Las etiquetas alternativas permiten asignar múltiples expresiones no preferentes a un concepto. Esto es útil para representar términos equivalentes, no descriptores, sinónimos, acrónimos, etc. Estas expresiones enriquecen el vocabulario disponible en el sistema ofreciendo a los usuarios un mayor número de posibles accesos a un mismo concepto y por lo tanto incrementar las posibilidades de éxito en los procesos de indización y búsqueda. Las etiquetas ocultas suelen asignarse a conceptos, no para que sean accesibles a los usuarios, sino para que sean procesadas por aplicaciones informáticas. Un ejemplo podrían ser las variantes de errores ortográficos de otras

etiquetas, bien sean preferentes o alternativas. En SKOS es posible asignar una etiqueta preferente de forma individualizada para cada idioma usando caracteres UNICODE que permitirían utilizar diferentes sistemas de escritura. La etiqueta `skos:prefLabel` indica el término preferente de un concepto, mientras que los términos alternativos (no preferentes) se denotan con `skos:altLabel`. Para las etiquetas ocultas se utiliza `skos:hiddenLabel`.

Elemento	Vocabulario	Definiciones de clases y propiedades
Etiqueta preferente	<code>skos:prefLabel</code>	<ul style="list-style-type: none"> ■ Instancia de <code>owl:DatatypeProperty</code> ■ Rango de valores: <code>rdf:PlainLiteral</code> ■ Dominio: Cualquier recurso ■ Un recurso no puede tener asociada más de una <code>skos:prefLabel</code> en cada idioma ■ Propiedad disyunta de <code>skos:altLabel</code> y <code>skos:hiddenLabel</code>
Etiqueta alternativa	<code>skos:altLabel</code>	<ul style="list-style-type: none"> ■ Instancia de <code>owl:DatatypeProperty</code> ■ Rango de valores: <code>rdf:PlainLiteral</code> ■ Dominio: Cualquier recurso ■ Propiedad disyunta de <code>skos:prefLabel</code> y <code>skos:hiddenLabel</code>
Etiqueta oculta	<code>skos:hiddenLabel</code>	<ul style="list-style-type: none"> ■ Instancia de <code>owl:DatatypeProperty</code> ■ Rango de valores: <code>rdf:PlainLiteral</code> ■ Dominio: Cualquier recurso ■ Propiedad disyunta de <code>skos:prefLabel</code> y <code>skos:altLabel</code>

Tabla 8: Vocabulario y definiciones de clases y propiedades para etiquetas léxicas.

A continuación se muestra un sencillo ejemplo del uso de los diferentes tipos de etiquetado para representar sus correspondientes unidades léxicas.

```

1. <rdf:RDF xmlns:skos="http://www.w3.org/2004/02/skos/core">
2.   <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#americanorte">
3.     <skos:prefLabel xml:lang="es">América del Norte</skos:prefLabel>
4.     <skos:altLabel xml:lang="es">Norte América</skos:altLabel>
5.     <skos:hiddenLabel xml:lang="es">América Norte</skos:hiddenLabel>
6.     <skos:prefLabel xml:lang="en">North America</skos:prefLabel>
7.   </skos:Concept>
8. </rdf:RDF>

```

Ejemplo 8: Definición de concepto y etiquetas léxicas asociadas

Es posible definir tipologías de relaciones entre etiquetas mediante una extensión denominada SKOS-XL⁴⁵ (W3C, 2008a3). Para establecer una relación léxica con esta extensión es necesario definir las etiquetas como recursos, igual que los conceptos, esquemas y colecciones. De esta forma se definen un tipo especial de entidad léxica al que se le asigna una cadena literal que puede repetirse para distintas unidades.

Elemento	Vocabulario	Definiciones de clases y propiedades
Etiqueta XL	skosxl:Label	<ul style="list-style-type: none"> ■ Instancia de owl:Class ■ Cardinalidad exacta de 1 con skosxl:literalForm ■ Clase disyunta de: skos:Collection, skos:Concept y skos:ConceptScheme
Forma literal	skosxl:literalForm	<ul style="list-style-type: none"> ■ Instancia de owl:DatatypeProperty ■ Dominio: skosxl:Label ■ Rango de valores cualquier cadena literal
Etiqueta preferente XL	skosxl:prefLabel	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Dominio skos:Concept ■ Rango de valores skosxl:Label ■ El conjunto (skosxl:prefLabel+skosxl:literalForm) es una subpropiedad de skos:prefLabel
Etiqueta alternativa XL	skosxl:altLabel	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Dominio skos:Concept ■ Rango de valores skosxl:Label ■ El conjunto (skosxl:altLabel+skosxl:literalForm) es una subpropiedad de skos:prefLabel
Etiqueta oculta XL	skosxl:hiddenLabel	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Dominio skos:Concept ■ Rango de valores skosxl:Label ■ El conjunto (skosxl:prefLabel+skosxl:literalForm) es una subpropiedad de skos:prefLabel
Relaciones entre etiquetas	skosxl:labelRelation	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty y owl:SymmetricProperty ■ Dominio y Rango skosxl:Label

Tabla: 9: Clases y propiedades para conceptos, esquemas

45 SKOS-XL: *Simple Knowledge Organization System eXtension for Labels*. Vocabulario RDF cuya finalidad es la aplicación del vocabulario básico de SKOS para el establecimiento de relaciones entre entidades léxicas.

Las unidades léxicas se definen como recursos de la clase `skosxl:Label` y se asocian con los conceptos con las correspondientes propiedades de `skosxl:prefLabel`, `skosxl:altLabel` y `skosxl:hiddenLabel`. Como los recursos definidos como `skosxl:Label` siempre tienen asociados una forma literal con `skosxl:literalForm`, se define la combinación de esta propiedad con cualquiera de los tipos de unidad léxica de `skosxl` como una subpropiedad de su correspondiente tipo de unidad léxica de `skos`. Por ejemplo, si un concepto está relacionado con un recurso de la clase `skosxl:Label` y este a su vez con su forma literal a través de `skosxl:prefLabel` se puede inferir la existencia de la propiedad `skos:prefLabel` asociado a dicho literal. Las relaciones entre entidades léxicas se realiza con la propiedad `skosxl:labelRelation`. Esta propiedad puede utilizarse directamente pero lo más eficaz es definir un tipo de relación personalizado como subpropiedad de `skosxl:labelRelation`. En el siguiente ejemplo puede observarse el mecanismo de SKOS XL para expresar este tipo de relaciones.

```

1. <!DOCTYPE rdf:RDF [
2. <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
3. <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
4. <!ENTITY ex "http://www.cine.org/">
5. <!ENTITY skos "http://www.w3.org/2004/02/skos/core#">]>
6. <!ENTITY skosxl "http://www.w3.org/2008/05/skos-xl">]>
7. <rdf:RDF xmlns:skos="&skos;" xmlns:rdf="&rdf;" xmlns:rdfs="&rdfs;"
      xmlns:ex="&ex;" xmlns:skosxl="&skosxl;" xml:base="&ex;">
8.   <rdf:Property rdf:ID="fullForm"/>
9.     <rdfs:subPropertyOf rdf:resource="&skosxl;labelRelation"/>
10.    <owl:inverseOf rdf:resource="#acronymForm"/>
11.  </rdf:Property>
12.  <rdf:Property rdf:ID="acronymForm"/>
13.    <rdfs:SubPropertyOf rdf:resource="&skos;LabelRelated"/>
14.  </rdf:Property>
15.  <skosxl:Label rdf:ID="ONUFull">
16.    <skosxl:literalForm xml:lang="es">Organización de las
      Naciones Unidas</skos:literalForm />
17.    <ex:acronymForm rdf:ID="ONUAbrev" />
18.  </skosxl:Label>
19.  <skosxl:Label rdf:ID="ONUAbrev">
20.    <skosxl:literalForm xml:lang="es">ONU</skos:literalForm />
21.    <ex:fullForm rdf:ID="ONUFull" />
22.  </skosxl:Label>
23.  <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#onu">
24.    <skosxl:prefLabel rdf:ID="ONUAbrev" />
25.    <skosxl:altLabel rdf:ID="ONUFull" />
26.  </skos:Concept>
27. </rdf:RDF>

```

Ejemplo 9: Definición de una relación entre etiquetas léxicas

Se definen dos propiedades `ex:fullForm` y `ex:acronymForm` con la finalidad de establecer posteriormente una relación para indicar que una unidad léxica es un acrónimo de otra. Adicionalmente se declara con OWL que son propiedades inversas. Seguidamente se crean los recursos `skosxl:Label` correspondientes a la forma completa y el acrónimo de un concepto, definiéndose sus propiedades `skosxl:literalForm`. Finalmente se establecen las relaciones entre ambos recursos con las etiquetas `ex:fullForm` y `ex:acronymForm`, así como las relaciones `skosxl:prefLabel` y `skosxl:altLabel` entre el concepto y las unidades léxicas.

El grafo resultante es ligeramente más complejo que el elaborado a partir de la situación en la que no es necesario realizar ningún tipo de relación entre unidades léxicas, tal y como se muestra en la siguiente figura.

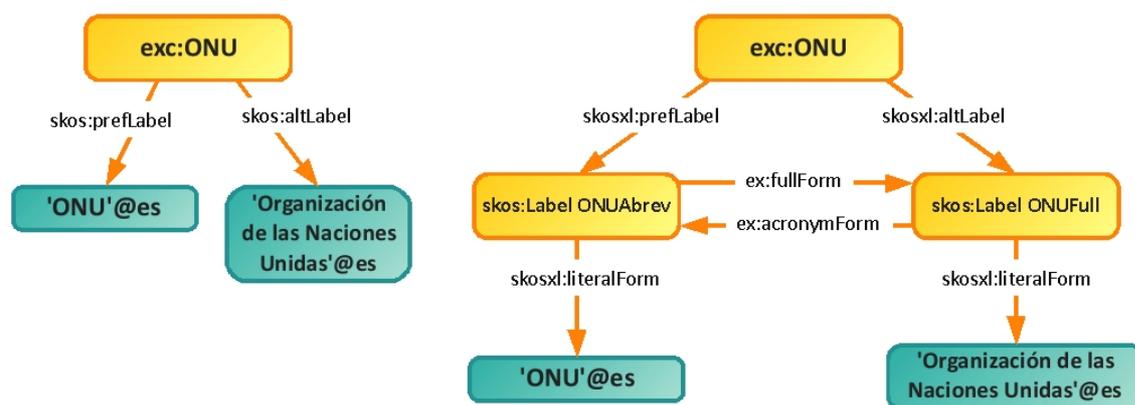


Ilustración 24: Grafo correspondiente a la codificación del ejemplo 9 mediante SKOS XL y su correspondiente grafo en el que no se establecen relaciones entre entidades léxicas.

3.6.5. Relaciones semánticas

Las relaciones semánticas de SKOS son enlaces entre conceptos. Estos enlaces son inherentes al significado de los conceptos relacionados. SKOS distingue dos tipos básicos de relaciones semánticas: jerárquicas y asociativas. Una relación jerárquica entre dos conceptos indica que uno de ellos es más general que otro. Por su parte, una relación asociativa indica que ambos conceptos están relacionados de alguna forma sin que ninguno de ellos sea más genérico o específico que el otro. Puede observarse que la tipología de relaciones es casi idéntica a la que se aplica en construcción de tesauros. En SKOS las relaciones semánticas son esenciales para la definición de

conceptos más allá de las etiquetas léxicas que tengan asociadas. En efecto, el significado de un concepto también se complementa con las relaciones semánticas que se establecen con otros conceptos.

Las relaciones jerárquicas básicas se definen sin la propiedad transitiva, únicamente para crear declaraciones de estructuras de conceptos. La propiedad transitiva en una estructura jerárquica implica que si un concepto A es más genérico que B, y B es más genérico que C se podrá deducir que A es más genérico que C.

También se contemplan las relaciones jerárquicas transitivas con la finalidad de poder realizar inferencias e implementar algoritmos de consultas expandidas en aplicaciones de búsqueda. Éste es el motivo por el que se definen clases superiores de relaciones jerárquicas que tienen asociadas la propiedad transitiva siendo las relaciones de uso básico, que no contemplan dicha propiedad, una subclase de aquellas.

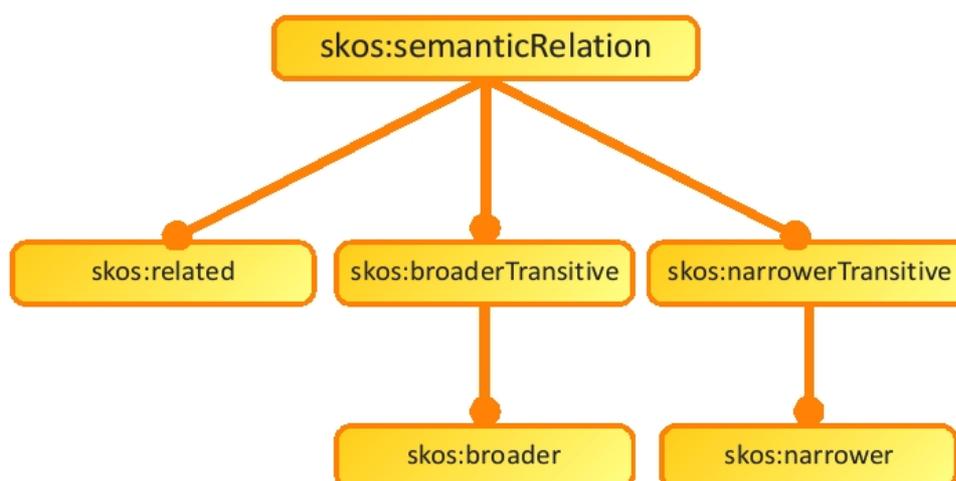


Ilustración 25: Diagrama organizativo de las relaciones semánticas en SKOS

Por su parte, la relación asociativa se define con la propiedad simétrica. Es decir, si un concepto A tiene una relación asociativa con B, se podrá deducir que B tiene una relación asociativa con A. Del mismo modo, es necesario indicar que la relación asociativa no dispone de la propiedad transitiva de forma que si A tiene una relación asociativa con B y B a su vez con C no se puede deducir que pueda establecerse una relación asociativa de A con C.

SKOS incorpora la propiedad `skos:semanticRelation` de la que se derivan `skos:broaderTransitive`, `skos:narrowerTransitive` (relaciones jerárquicas genérica y específica respectivamente) y `skos:related` (para las relaciones asociativas).

Elemento	Vocabulario	Definiciones de clases y propiedades
Relación semántica	<code>skos:semanticRelation</code>	<ul style="list-style-type: none"> ■ Instancia de <code>owl:ObjectProperty</code> ■ Dominio <code>skos:Concept</code> ■ Rango de valores: <code>skos:Concept</code>
Relación genérica transitiva	<code>skos:broaderTransitive</code>	<ul style="list-style-type: none"> ■ Instancia de <code>owl:ObjectProperty</code> ■ Instancia de <code>owl:TransitiveProperty</code> ■ Subpropiedad de <code>skos:semanticRelation</code>
Relación específica transitiva	<code>skos:narrowerTransitive</code>	<ul style="list-style-type: none"> ■ Instancia de <code>owl:ObjectProperty</code> ■ Instancia de <code>owl:TransitiveProperty</code> ■ Se define como <code>owl:inverseOf</code> de <code>skos:broaderTransitive</code>⁴⁶ ■ Subpropiedad de <code>skos:semanticRelation</code>
Relación genérica	<code>skos:broader</code>	<ul style="list-style-type: none"> ■ Instancia de <code>owl:ObjectProperty</code> ■ No es instancia de <code>owl:TransitiveProperty</code> ■ Subpropiedad de <code>skos:broaderTransitive</code>
Relación específica	<code>skos:narrower</code>	<ul style="list-style-type: none"> ■ Instancia de <code>owl:ObjectProperty</code> ■ No es instancia de <code>owl:TransitiveProperty</code> ■ Se define como <code>owl:inverseOf</code> de <code>skos:broader</code>⁴⁷ ■ Subpropiedad de <code>skos:narrowerTransitive</code>
Relación asociativa	<code>skos:related</code>	<ul style="list-style-type: none"> ■ Instancia de <code>owl:ObjectProperty</code> ■ Instancia de <code>owl:SymmetricProperty</code> ■ No es instancia de <code>owl:TransitiveProperty</code> ■ Subpropiedad de <code>skos:semanticRelation</code> ■ Disyunto con <code>skos:broaderTransitive</code>

Tabla 10: Vocabulario y definiciones de clases y propiedades para elementos de relaciones semánticas

⁴⁶ En SKOS no se define `skos:broaderTransitive` como `owl:inverseOf` de `skos:narrowerTransitive`, ya que en OWL solo es necesario hacer la definición en un único sentido, en este caso desde el más específico al más genérico.

⁴⁷ Idéntica situación que en `skos:narrowerTransitive`.

Las relaciones jerárquicas no transitivas se definen con las subpropiedades `skos:broader` y `skos:narrower`.

```
1. <rdf:RDF xmlns:skos="http://www.w3.org/2004/02/skos/core">
2.   <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#america">
3.     <skos:prefLabel xml:lang="es">América</skos:prefLabel>
4.     <skos:narrower rdf:resource="http://www.ejemplo.org/conceptos#americasur"/>
5.   </skos:Concept>
6.   <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#americasur">
7.     <skos:prefLabel xml:lang="es">América del Sur</skos:prefLabel>
8.     <skos:broader rdf:resource="http://www.ejemplo.org/conceptos#america"/>
9.     <skos:related rdf:resource="http://www.ejemplo.org/conceptos#cultivocafe"/>
10.  </skos:Concept>
11.  <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#cultivocafe">
12.    <skos:prefLabel xml:lang="es">Cultivo de Café</skos:prefLabel>
13.    <skos:related rdf:resource="http://www.ejemplo.org/conceptos#americasur"/>
14.  </skos:Concept>
15. </rdf:RDF>
```

Ejemplo 10: Definición de conceptos y relaciones semánticas asociadas

Veamos a continuación un ejemplo a partir de la entrada que para el descriptor “Tesauro” ofrece el Tesauro de la UNESCO⁴⁸.

Tesauro

English term: Thesauri
Terme français: Thésaurus
Русский термин : Тезаурусы
NA Lenguaje documental controlado y dinámico que contiene érminos relacionados semántica y genéricamente que abarcan de manera exhaustiva una esfera concreta del conocimiento.
MT 5.05 Ciencias de la información
UP Descriptores
UP Tesauro monolingüe
UP Tesauro multilingüe
UP Tesauro
TG Lenguaje de indexación [32]
TE Compilación de tesauro [70]
TR Control terminológico [19]
TR Encabezamiento por materia [185]
TR Material de referencia [123]
TR Terminología [404]

Ejemplo 11: Entrada del descriptor “Tesauro” en el Tesauro de la UNESCO.

48 El Tesauro de la UNESCO dispone de una versión electrónica en la siguiente dirección:
<http://databases.unesco.org/thessp/>

3. Representación de tesauros con SKOS

Seguidamente detallamos la codificación que se realizaría con SKOS para dicho descriptor, tanto del propio concepto, como de las relaciones y conceptos asociados a las mismas.

```
1. <rdf:RDF xmlns:skos="http://www.w3.org/2004/02/skos/core">
2.   <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#tesauro">
3.     <skos:prefLabel xml:lang="es">Tesauro</skos:prefLabel>
4.     <skos:prefLabel xml:lang="en">Thesauri</skos:prefLabel>
5.     <skos:prefLabel xml:lang="fr">Thésaurus</skos:prefLabel>
6.     <skos:prefLabel xml:lang="ru">Тезауры</skos:prefLabel>
7.     <skos:scopeNote xml:lang="es">Lenguaje documental controlado y dinámico
      que contiene términos relacionados semántica y genéricamente que
      abarcan de manera exhaustiva una esfera concreta del conocimiento.
9.   </skos:scopeNote>
10.  <skos:altLabel xml:lang="es">Descriptores</skos:prefLabel>
11.  <skos:altLabel xml:lang="es">Tesauro monolingüe</skos:prefLabel>
12.  <skos:altLabel xml:lang="es">Tesauro multilingüe</skos:prefLabel>
13.  <skos:altLabel xml:lang="es">Thesauri</skos:prefLabel>
14.  <skos:broader rdf:resource="http://www.ejemplo.org/conceptos#lengindex"/>
16.  <skos:narrower rdf:resource="http://www.ejemplo.org/conceptos#comptes"/>
17.  <skos:related rdf:resource="http://www.ejemplo.org/conceptos#controlterm"/>
18.  <skos:related rdf:resource="http://www.ejemplo.org/conceptos#encmateria"/>
19.  <skos:related rdf:resource="http://www.ejemplo.org/conceptos#matref"/>
20.  <skos:related rdf:resource="http://www.ejemplo.org/conceptos#terminologia"/>
21. </skos:Concept>
22. <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#lengindex">
23.   <skos:prefLabel xml:lang="es">Lenguaje de indización</skos:prefLabel>
24. </skos:Concept>
22. <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#comptes">
23.   <skos:prefLabel xml:lang="es">Compilación de tesauro</skos:prefLabel>
24. </skos:Concept>
22. <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#controlterm">
23.   <skos:prefLabel xml:lang="es">Control terminológico</skos:prefLabel>
24. </skos:Concept>
22. <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#encmateria">
23.   <skos:prefLabel xml:lang="es">Encabezamiento de materia</skos:prefLabel>
24. </skos:Concept>
22. <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#matref">
23.   <skos:prefLabel xml:lang="es">Material de referencia</skos:prefLabel>
24. </skos:Concept>
22. <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#terminologia">
23.   <skos:prefLabel xml:lang="es">Terminología</skos:prefLabel>
24. </skos:Concept>
25. </rdf:RDF>
```

Ejemplo 12: Codificación con SKOS del descriptor “Tesauro” según la entrada correspondiente en el Tesauro de la UNESCO

El ejemplo anterior puede representarse en forma de grafo de la siguiente forma:

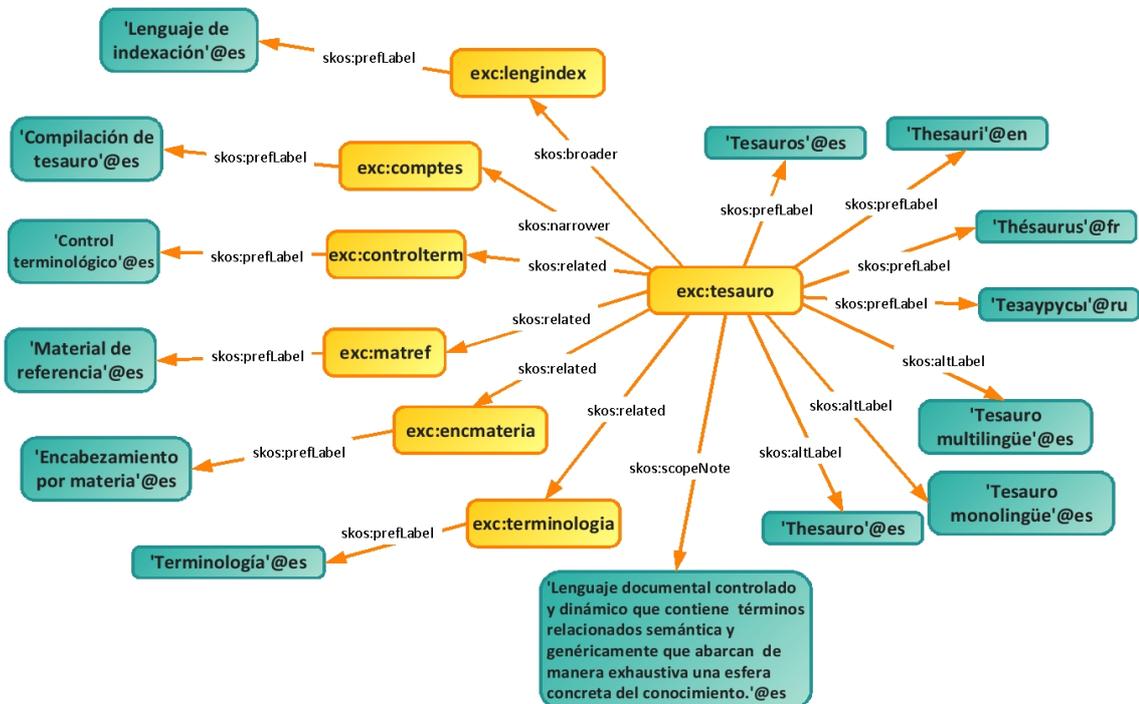


Ilustración 26: Grafo correspondiente a la codificación con SKOS del descriptor “Tesauro” según la entrada correspondiente en el Tesauro de la UNESCO.

Una de las características de SKOS es su flexibilidad, tal vez por ello no se han definido, de forma explícita, restricciones semánticas que impidan la creación de ciclos en las estructuras jerárquicas o el establecimiento de relaciones asociativas entre conceptos, pertenecientes a la misma jerarquía.

Pese a todo, el modelo de SKOS sí contempla la restricción semántica que supone la disyunción entre `skos:related` y `skos:broaderTransitive` (y por tanto con `skos:narrowerTransitive`), lo cual impide el establecimiento de relaciones asociativas entre conceptos estructurados en la misma línea jerárquica.

En gran medida, las propias aplicaciones serán las encargadas de controlar que estas y otras posibles restricciones no mencionadas hasta ahora (como la reflexividad de las relaciones semánticas) se aplican correctamente, siempre según los requisitos del esquema conceptual definidos por los usuarios del sistema.

3.6.6. Notaciones

Con SKOS es posible asociar una notación a un concepto. Esta característica permite asociar un concepto a su correspondiente entrada dentro de un tesoro, clasificación decimal o cualquier otro sistema de organización en el que se identifiquen los elementos con signaturas decimales, códigos de identificación o similares. Así se asocia un concepto al ámbito específico de un esquema conceptual. Los valores asociados a las notaciones suelen definirse con un literal tipificado que se construye generalmente a partir de tipos de datos definidos con un esquema XML. Por lo tanto, el valor que puede asociarse a una notación es una combinación de un literal tipificado y de una URI donde se define el formato de dicho literal. Un mismo concepto pueda tener asignadas varias notaciones, aunque una notación solo debería ser asignada a un único concepto. Ésto es una convención al no definirse formalmente en el modelo de datos de SKOS por lo que no hay condiciones de integridad para este elemento.

Elemento	Vocabulario	Definiciones de clases y propiedades
Notación	skos:notation	<ul style="list-style-type: none"> ■ Instancia de owl:DatatypeProperty ■ Dominio skos:Concept ■ Rango de valores: Literal tipado

Tabla 11: Vocabulario y definiciones de clases y propiedades para notaciones.

Un sencillo ejemplo de notación asociada a un concepto podría ser el siguiente:

```

1. <!DOCTYPE rdf:RDF [<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2. <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#"
3. <!ENTITY exdt "http://www.cine.org/datatypes#"
4. <!ENTITY skos "http://www.w3.org/2004/02/skos/core#">]>
5. <rdf:RDF xmlns:skos="&skos;" xmlns:rdf="&rdf;" xmlns:rdfs="&rdfs;"
   xmlns:exdt="&exdt;"
6.   <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#america">
7.     <skos:prefLabel>America</skos:prefLabel>
8.     <skos:notation rdf:datatype="&exdt;signatura">D.GEO.3.0</skos:notation>
9.   </skos:Concept>
10. </rdf:RDF>

```

Ejemplo 13: Definición de una notación para un concepto

3.6.7. Documentación

Los elementos de documentación permiten incluir información no formal acerca del significado de relaciones, conceptos y etiquetas, así como su evolución a lo largo del tiempo, notas de publicación, ejemplos, notas de ámbito de uso y otro tipo de información dirigida fundamentalmente hacia los usuarios humanos. En definitiva, estos elementos de SKOS tienen como objetivo documentar ciertos aspectos de los elementos del esquema conceptual. SKOS ofrece una tipología que permite distinguir el tipo de documentación realizada, permitiendo formalizar, hasta cierto punto, estos contenidos. La tipología de elementos de documentación es la siguiente:

- Nota: Elemento general del que derivan el resto de elementos de documentación.
- Nota de alcance: Ofrece información acerca del significado de un concepto en un determinado ámbito de uso como por ejemplo la forma de aplicar un determinado concepto en procesos de indización.
- Nota de historial: El objetivo de este elemento es el de describir cambios importantes en el significado de un concepto.
- Nota de cambios: Permite documentar cambios realizados sobre un concepto en los procesos de gestión y mantenimiento de un esquema conceptual. Este elemento resulta muy útil para documentar los cambios en la ubicación de un concepto dentro de la estructura de relaciones.
- Definición: Ofrece la posibilidad de describir una completa explicación acerca del significado de un concepto.
- Nota de edición: Información de tareas administrativas de edición y publicación.
- Ejemplo: Este elemento suministra ejemplos de uso de un concepto.

3. Representación de tesauros con SKOS

Elemento	Vocabulario	Definiciones de clases y propiedades
Nota	skos:note	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Dominio rdfs:Resource
Nota de alcance	skos:scopeNote	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Subpropiedad de skos:note ■ Dominio rdfs:Resource
Nota de historial	skos:historyNote	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Subpropiedad de skos:note ■ Dominio rdfs:Resource
Nota de cambios	skos:changeNote	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Subpropiedad de skos:note ■ Dominio rdfs:Resource
Definición	skos:definition	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Subpropiedad de skos:note ■ Dominio rdfs:Resource
Nota de edición	skos:editorialNote	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Subpropiedad de skos:note ■ Dominio rdfs:Resource
Ejemplo	skos:example	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Subpropiedad de skos:note ■ Dominio rdfs:Resource

Tabla 12: Vocabulario y definiciones de clases y propiedades para elementos de documentación.

```

1. <rdf:RDF xmlns:skos="http://www.w3.org/2004/02/skos/core"
   xmlns:dc="http://purl.org/dc/elements/1.1/">
2.   <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#americanorte">
3.     <skos:prefLabel xml:lang="es">América del Norte</skos:prefLabel>
4.     <skos:definition>Subcontinente que forma parte del Continente Americano
       cuya extensión abarca el área geográfica de Canadá, Estados Unidos y
       México</skos:definition>
5.     <dc:source>Diccionario Universal de Internet, junio 2008</dc:source>
6.   </skos:Concept>
7. </rdf:RDF>

```

Ejemplo 14: Definición de concepto y elementos de documentación asociados

Por supuesto, SKOS permite aprovechar elementos de otros vocabularios como Dublin Core. En el ejemplo anterior se incluye un ejemplo de ello en el que se hace uso del elemento `dc:source`. Los elementos de documentación de SKOS puede utilizarse en

documentos cuyo contenido principal utilice otro vocabulario.

3.6.8. Colecciones de conceptos

Las colecciones de conceptos en SKOS permiten definir agrupaciones para enriquecer la estructura sin llegar a establecer relaciones semánticas explícitas que distorsionen las estructuras jerárquicas o asociativas del esquema conceptual. Los conceptos y las colecciones de conceptos son disyuntos en SKOS por lo que no es posible establecer entre ambos relaciones semánticas de cualquier tipo.

Las colecciones pueden estar asociadas a recursos RDF identificados a través de una URI aunque lo más común es que se declaren como nodos vacíos. Las colecciones pueden tener asociadas etiquetas léxicas y al igual que las colecciones de RDF pueden estar ordenadas utilizando listas de elementos.

Es posible inferir los elementos de colección a partir de los elementos de una colección ordenada. También está permitida la inclusión de colecciones dentro de otras colecciones como si de un elemento más se tratara.

Elemento	Vocabulario	Definiciones de clases y propiedades
Colección	skos:Collection	<ul style="list-style-type: none"> ■ Instancia de owl:Class ■ Disyunto de skos:Concept y skos:ConceptScheme
Colección ordenada	skos:OrderedCollection	<ul style="list-style-type: none"> ■ Instancia de owl:Class ■ Subclase de skos:Collection ■ Solo puede asignado un único elemento skos:memberList
Miembro de colección	skos:member	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Dominio skos:Collection
Miembro de una lista	skos:memberList	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Instancia de owl:FunctionalProperty ■ Dominio skos:OrderedCollection ■ Rango de valores rdf:List ■ Permite tener asignados varios valores de skos:member

Tabla 13: Vocabulario y definiciones de clases y propiedades para elementos de colecciones

```
1. <rdf:RDF xmlns:skos="http://www.w3.org/2004/02/skos/core"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
2.   <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#leche">
3.     <skos:prefLabel xml:lang="es">Leche</skos:prefLabel>
4.     <skos:narrower rdf:resource="http://www.ejemplo.org/conceptos#lechevaca"/>
5.     <skos:narrower rdf:resource="http://www.ejemplo.org/conceptos#lecheoveja"/>
6.     <skos:narrower rdf:resource="http://www.ejemplo.org/conceptos#lechecabra"/>
7.   </skos:Concept>
8.   <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#lechevaca">
9.     <skos:prefLabel xml:lang="es">Leche de vaca</skos:prefLabel>
10.    <skos:broader rdf:resource="http://www.ejemplo.org/conceptos#leche"/>
11.  </skos:Concept>
12.  <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#lecheoveja">
13.    <skos:prefLabel xml:lang="es">Leche de oveja</skos:prefLabel>
14.    <skos:broader rdf:resource="http://www.ejemplo.org/conceptos#leche"/>
15.  </skos:Concept>
16.  <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#lechecabra">
17.    <skos:prefLabel xml:lang="es">Leche de oveja</skos:prefLabel>
18.    <skos:broader rdf:resource="http://www.ejemplo.org/conceptos#leche"/>
19.  </skos:Concept>
24.  <skos:Collection>
25.    <skos:prefLabel>Tipos de leche según su origen animal</skos:prefLabel>
26.    <skos:member rdf:resource="http://www.ejemplo.org/conceptos#lecheoveja"/>
27.    <skos:member rdf:resource="http://www.ejemplo.org/conceptos#lechecabra"/>
28.    <skos:member rdf:resource="http://www.ejemplo.org/conceptos#lechevaca"/>
29.  </skos:Collection>
30.  <skos:OrderedCollection>
31.    <skos:prefLabel>Tipos de leche ordenados según su volumen de
      producción</skos:prefLabel>
32.    <skos:memberList>
33.      <skos:member rdf:resource="http://www.ejemplo.org/conceptos#lechevaca"/>
34.      <skos:member rdf:resource="http://www.ejemplo.org/conceptos#lecheoveja"/>
35.      <skos:member rdf:resource="http://www.ejemplo.org/conceptos#lechecabra"/>
36.    </skos:memberList>
36.  </skos:OrderedCollection>
37. </rdf:RDF>
```

Ejemplo 15: Conceptos y agrupación de los mismos utilizando dos colecciones (una de ellas ordenada)

3.6.9. Propiedades de mapeado

Con SKOS es posible incluir conceptos en varios esquemas conceptuales permitiendo así su reutilización. De forma adicional, SKOS ofrece la funcionalidad de realizar un mapeado entre conceptos incluidos en esquemas conceptuales distintos, estableciendo de esta forma una correspondencia entre los mismos. Puede suceder que la correspondencia entre ambos conceptos no sea exacta y por tanto SKOS distingue diferentes tipos de mapeado. Es posible declarar que entre dos conceptos existe una correspondencia exacta, que uno de ellos es más genérico o específico que

otro o establecer una correspondencia de asociación. No debe infravalorarse la inclusión de estas propiedades en SKOS, ya que proporcionan un mecanismo formal para realizar un mapeado entre conceptos, muy útil en el momento de utilizar varios tesauros en el proceso de recuperación de información. Las propiedades de correspondencia son subclases de sus correspondientes relaciones semánticas, tal y como puede observarse en la figura siguiente.

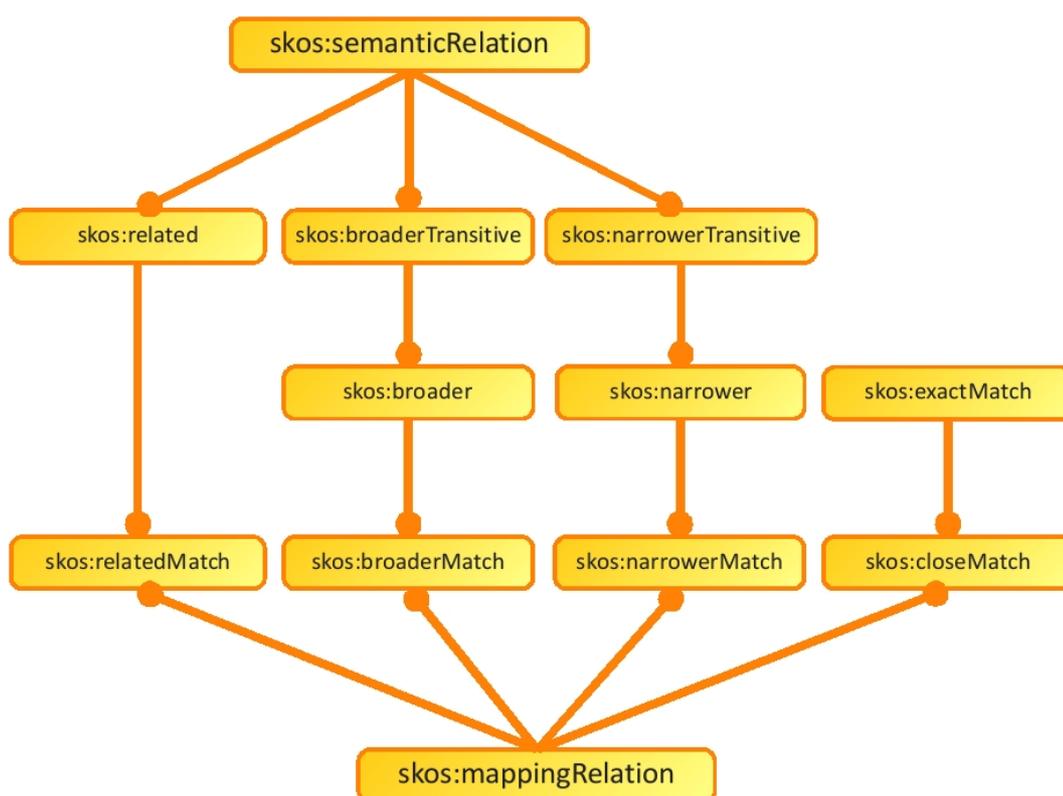


Ilustración 27: Diagrama organizativo de las relaciones semánticas y de correspondencia

Estas propiedades no tienen la misma función que una relación semántica. Por tanto relacionar dos conceptos mediante `skos:broaderMatch` no implica la creación de ningún tipo de jerarquía, simplemente indica en qué medida se corresponde dos conceptos pertenecientes a esquemas conceptuales diferentes. La inclusión de la propiedad `skos:closeMatch` ofrece una forma de ampliar relaciones de mapeado que puedan sugerirse en un futuro o que se desarrollen en una aplicación concreta. La propiedad `skos:exactMatch` se define como subpropiedad de `skos:closeMatch` e indica una correspondencia completa y exacta entre dos conceptos.

En la siguiente tabla se describen los elementos, vocabulario asociado y las definiciones de clases y propiedades del mismo.

Elemento	Vocabulario	Definiciones de clases y propiedades
Relación de mapeado	skos:mappingRelation	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Dominio skos:Concept ■ Rango de valores skos:Concept
Correspondencia cercana	skos:closeMatch	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Instancia de owl:SymmetricProperty ■ Subpropiedad de skos:mappingRelation
Correspondencia exacta	skos:exactMatch	<ul style="list-style-type: none"> ■ Subpropiedad de skos:closeMatch
Correspondencia genérica	skos:broadMatch	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Subpropiedad de skos:mappingRelation ■ Subpropiedad de skos:broader
Correspondencia específica	skos:narrowerMatch	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Subpropiedad de skos:mappingRelation ■ Subpropiedad de skos:narrower ■ Se define como owl:inverseOf de skos:broader⁴⁹
Correspondencia asociativa	skos:relatedMatch	<ul style="list-style-type: none"> ■ Instancia de owl:ObjectProperty ■ Instancia de owl:SymmetricProperty ■ Subpropiedad de skos:mappingRelation ■ Subpropiedad de skos:related

Tabla 14: Vocabulario y definiciones de clases y propiedades para elementos de colecciones

A continuación se incluye un ejemplo, que muestra las relaciones de correspondencia entre conceptos de diferentes esquemas.

```

1. <rdf:RDF xmlns:skos="http://www.w3.org/2004/02/skos/core">
2.   <skos:ConceptScheme rdf:about="http://www.ejemplo.org/esquema1">
3.     <skos:hasTopConcept
4.       rdf:resource="http://www.ejemplo.org/conceptos#america"/>
5.   </skos:ConceptScheme>
6.   <skos:ConceptScheme rdf:about="http://www.ejemplo.org/esquema2">
7.     <skos:hasTopConcept
8.       rdf:resource="http://www.ejemplo.org/conceptos#continenteamericano"/>
9.   </skos:ConceptScheme>

```

49 Idéntica situación que para skos:narrowerTransitive y skos:narrower.

```

8. <skos:Concept rdf:about="http://www.ejemplo.org/conceptos#america">
9.   <skos:prefLabel xml:lang="es">América</skos:prefLabel>
10.  <skos:inScheme rdf:resource="http://www.ejemplo.org/esquema1"/>
11.  <skos:exactMatch
      rdf:resource="http://www.ejemplo.org/conceptos#continenteamericano"/>
12. </skos:Concept>
13. <skos:Concept
      rdf:about="http://www.ejemplo.org/conceptos#continenteamericano">
14.  <skos:prefLabel xml:lang="es">Continente Americano</skos:prefLabel>
15.  <skos:inScheme rdf:resource="http://www.ejemplo.org/esquema2"/>
16.  <skos:exactMatch rdf:resource="http://www.ejemplo.org/conceptos#america"/>
17. </skos:Concept>
18. </rdf:RDF>

```

Ejemplo 16: Definición de relaciones de correspondencia

3.7. Ventajas de la aplicación de SKOS

La primera ventaja obvia de SKOS es la representación de tesauros conceptuales en una forma que se amolda perfectamente a los requisitos de la Web Semántica. A pesar de encontrarse en fase de desarrollo, muchas aplicaciones⁵⁰ de gestión de tesauros o indexación de documentos, ya hacen uso de este modelo. Esto se debe a que los elementos y propiedades esenciales se han definido plenamente y conforman un núcleo sólido de desarrollo.

Posiblemente SKOS evolucione en un futuro, con nuevos tipos de relaciones, adaptándose con RDF y OWL. Aun así, el modelo de estructuración de conceptos, etiquetas y relaciones se mantendrá a largo plazo por lo que se ha comenzado a emplear incluso antes de que el W3C haya redactado una primera recomendación definitiva.

Los tesauros en particular o los esquemas conceptuales en general se han empleado, desde hace tiempo, en tareas de organización y recuperación de información. Pese a ello se han diseñado soluciones parciales o con un ámbito de actuación limitado. En la web la mayoría de estas iniciativas se han creado mediante vocabularios XML e incluso RDF/XML para la implementación de sistemas funcionales.

⁵⁰ Una recopilación de estas aplicaciones pueden encontrarse en una página web mantenida por el W3C: <http://esw.w3.org/topic/SkosDev/ToolShed>

Sin embargo, hasta la aparición de SKOS estos desarrollos no estaban coordinados entre sí, ni tenían una visión global de aplicación válida para cualquier tipo de esquema conceptual. Tampoco compartían un modelo común, por lo que el intercambio de datos entre sistemas debía hacerse tras aplicar mecanismos de mapeado, que en ocasiones no podían realizarse de forma completa.

Otra de las ventajas de SKOS es que resulta mucho más sencillo de utilizar que el desarrollo completo de una ontología. De hecho SKOS se trata de una ontología y por lo tanto podría ampliarse o adaptarse a través de RDF y OWL. En cierta medida esto asegura la viabilidad de SKOS y su evolución a lo largo del tiempo. Por tanto, SKOS no se trata de un desarrollo cerrado, sino que en su propia naturaleza están disponibles aquellos mecanismos necesarios para su adaptación.

La interoperabilidad de sistemas es otro de los aspectos en los que participa SKOS ya que los metadatos constituyen el núcleo fundamental de la Web Semántica. Resulta mucho más ventajoso definir reglas y restricciones, sobre la información de un sistema, fuera del código de programación y representarlas en documentos elaborados con un lenguaje estándar.

Esto es lo que ofrecen los metadatos y las ontologías a través de RDF y OWL y por lo tanto algo que también aporta SKOS. El intercambio de esquemas conceptuales, representados con un modelo y vocabulario estándar como SKOS, ofrece una serie de ventajas derivadas de la incorporación y consiguiente reutilización de recursos externos.

Una posible aplicación sería la incorporación de tesauros externos, como un elemento integrado en sistemas de información, para los procesos de organización y búsqueda. Es una técnica similar a la sindicación de contenidos web. Dentro de esta integración ya hemos mencionado anteriormente la ampliación de las consultas de búsqueda a otros tesauros con los que el tesoro del sistema haya establecido relaciones de correspondencia entre conceptos.

La adopción del modelo de SKOS y la separación entre el nivel conceptual y léxico del tesoro permite una mayor eficacia en los procesos de indización. De esta forma los cambios en el nivel léxico no tendrían repercusión en la asignación de conceptos a los recursos de información.

La utilidad de los esquemas conceptuales no se limita a los procesos de indización y búsqueda de información sino que también podría utilizarse en los CMS⁵¹. La principal dificultad a la que se enfrentan estos sistemas es la creación de una herramienta de organización de información, integrada con los procesos de generación automática de los correspondientes sistemas de navegación del sitio web.

Con un tesauro podría diseñarse el esquema de organización y con una ontología se definirían las reglas de correspondencia para crear dinámicamente el sistema de navegación. SKOS podría utilizarse para la descripción de la organización de contenidos de un sitio web y participaría, empleando una ontología, para el diseño de sistemas de navegación de un sitio web.

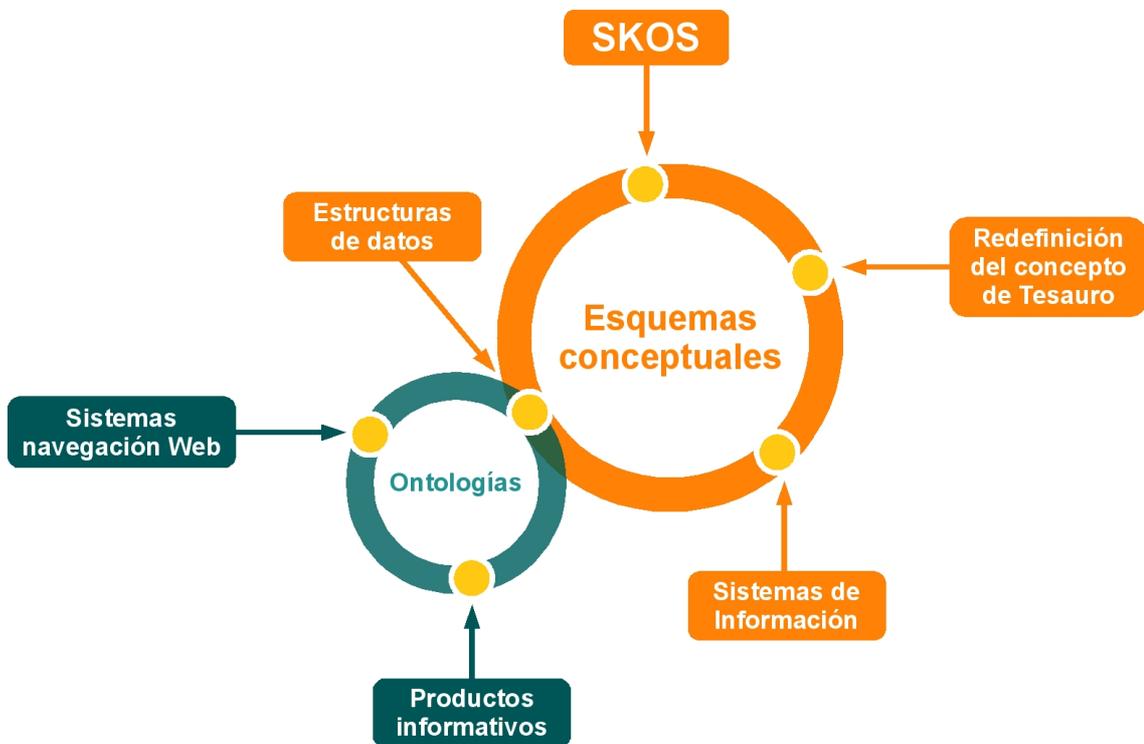


Ilustración 28: El papel de SKOS y los esquemas conceptuales en los Sistemas de Información, complementado con las ontologías

⁵¹ CMS: *Content Management System*. Se tratan de aplicaciones especializadas en la gestión de contenidos de sitios web. La principal característica de estos sistemas es que se utilizan a través de un navegador web.

Por todo lo anterior, pensamos que SKOS, además de poder emplearse a nivel general en la Web, también podría ser utilizado en un primer momento dentro de sistemas de información corporativos, asociados a su explotación en Intranets. El uso conjunto de SKOS y OWL podría permitir la conceptualización de un sistema de información como una esfera de datos estructurados, de la cual podrían definirse productos y servicios en forma de vistas específicas de los contenidos gestionados por el sistema.

4

*Propuesta de modelo
conceptual de aplicación*

4. Propuesta de modelo conceptual de aplicación

"Los ordenadores son inútiles. Sólo pueden darte respuestas." (Pablo Picasso)

Resumen: En el presente capítulo se procederá a definir las principales características de una aplicación para la gestión y uso de tesauros basados en el modelo SKOS. Para ello realizaremos un análisis del entorno tecnológico de desarrollo y ejecución, fundamentalmente, sistemas de bases de datos y lenguajes de programación para la Web. A continuación se identificarán las características funcionales de la aplicación como paso previo a su modelado conceptual con UML.

4.1. Entorno tecnológico de la aplicación.

La aparición de Internet en general y de la Web en particular ofrece posibilidades que van más allá de la clásica aplicación de escritorio. La información ya no se localiza únicamente en servidores corporativos, sino que puede estar ubicada para su mantenimiento y explotación en cualquier ubicación de Internet. De esta forma, la Web se ha convertido en un marco de creación de aplicaciones convirtiéndose en un entorno cliente/servidor Universal.

La situación descrita anteriormente ha sido propiciada por la creación de nuevos lenguajes y entornos de programación orientados al desarrollo de aplicaciones web. Es muy frecuente que estas herramientas estén integradas con sistemas de gestión de bases de datos, lo que permite diseñar aplicaciones avanzadas y en ocasiones con estructuras de almacenamiento de datos muy complejas.

Nuestro trabajo se orienta al planteamiento de un modelo de aplicación para la gestión y uso de tesauros en Internet. Por lo tanto resulta razonable que dicha propuesta fije su horizonte en el desarrollo, en un futuro, de una aplicación web. A efectos de realizar tal propuesta es necesario realizar un análisis, cuanto menos introductorio, del entorno de desarrollo de aplicaciones Web para comprender cómo funcionan y se relacionan las tecnologías utilizadas en este campo.

4.1.1. La Web como cliente/servidor universal.

A principios de los años 70, las organizaciones con sistemas informáticos centralizados empleaban grandes ordenadores centrales, encargados de las tareas asociadas al almacenamiento y procesamiento de datos. Estos ordenadores se conectaban a través de redes informáticas con terminales donde los usuarios introducían datos y consultaban la información que solicitaban al ordenador central. El ordenador central disponía de procesadores de gran potencia ya que soportaba toda la carga de trabajo del sistema. Se trataban de máquinas muy costosas que necesitaban una infraestructura de red que hiciera posible la transmisión de aquellos datos asociados a la interfaz de representación de datos, que eran visualizados por los usuarios desde terminales sin ninguna capacidad de procesamiento.

El desarrollo de los ordenadores personales y el consiguiente abaratamiento del equipamiento informático hizo posible otro modelo donde el terminal del usuario realiza gran parte del trabajo del sistema. Los ordenadores personales disponen de unidades centrales de proceso y discos duros de almacenamiento, que permiten la ejecución de programas encargados del procesamiento de datos y todos aquellos aspectos de la interfaz de usuario. En estos sistemas los ordenadores centrales son sustituidos por servidores de ficheros, que en realidad se tratan de ordenadores personales especializados en la función de almacenamiento y atención de solicitudes de datos. Los puestos de trabajo de los usuarios disponen de software de base de datos encargados de procesar las respuestas de las peticiones que han sido atendidas por el servidor de ficheros.

Tras un breve análisis es fácil concluir que en realidad la evolución de los sistemas distribuidos ha oscilado desde un extremo a otro, desde sistemas donde la carga de trabajo la realiza un ordenador central hacia otros donde un servidor de ficheros no realiza ningún proceso, con la salvedad de atender peticiones de datos. Los sistemas centralizados aseguran la integridad de datos y aplicaciones pero sobrecargan al ordenador central con los procesos asociados a la interfaz de usuario, limitándose consiguientemente el número de accesos que pueden atender. Los sistemas basados en servidores de ficheros descargan al ordenador que almacena los datos de cualquier tarea ajena a esta función. Por contra, la integridad de las aplicaciones se ve dificultada ya que cada puesto de trabajo requiere un mantenimiento complejo de las aplicaciones que procesan los datos y, en ocasiones, las aplicaciones necesitan ficheros completos de bases de datos, por lo que el ancho de banda necesario es considerable

(Leloup, 1998).

La arquitectura cliente/servidor encuentra un equilibrio entre ambos modelos de trabajo, fundamentándose en la existencia de un programa, denominado servidor, que atiende las peticiones de otra aplicación denominada cliente, generalmente a través de una red informática. En este modelo, el servidor desempeña tareas de almacenamiento de datos, atención de solicitudes y procesado de los mismos (back-end), mientras que el cliente realiza las peticiones y procesa las respuestas para adecuarlas a una interfaz de usuario (front-end).

Las principales bazas de esta arquitectura son la integridad de datos y de las aplicaciones. Las modificaciones de los programas que gestionan el sistema se realizan en el servidor. Los usuarios únicamente han de tener instalada en su máquina la aplicación cliente que, generalmente, se actualiza con menor frecuencia que las aplicaciones del servidor. Otra de las ventajas de este modelo es que permite el desarrollo de sistemas abiertos heterogéneos, donde se interconectan cualquier tipo de ordenador, alcanzando un funcionamiento eficaz en cualquier entorno de trabajo del usuario final (Corbin, 1991).

La difusión de las tecnologías asociadas a Internet, más concretamente de la Web, ha propiciado el desarrollo de lo que Lefebvre (1997) denominó cliente/servidor universal. Esta expresión denota la aplicación de la Web para el desarrollo de todo tipo de aplicaciones, incluso aquellas en las que la interacción entre el usuario y el sistema fuera una constante.

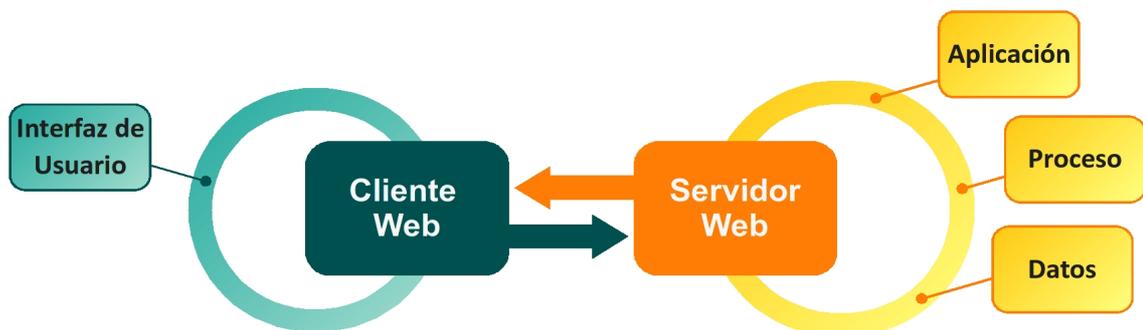


Ilustración 29: Esquema básico de funcionamiento de la Web como Cliente/Servidor Universal

Los usuarios utilizan los navegadores como interfaz para acceder a cualquier aplicación

web, ya sea de forma anónima o mediante un mecanismo de identificación. Las aplicaciones de gestión de bases de datos se encuentran en un servidor de aplicaciones y el servidor web hace de intermediario entre éstas y el cliente, recibiendo las peticiones, trasladándolas a las aplicaciones de gestión y enviando los resultados al cliente.

Los costes de desarrollo de la aplicación cliente son prácticamente inexistentes ya que se tratan de los propios navegadores web, lo cual aumenta la interoperabilidad de sistemas. Las aplicaciones web comparten la misma interfaz, lo cual permite reducir el tiempo de aprendizaje, centrándose únicamente en las funcionalidades de la aplicación. Resulta de enorme interés el efecto que está teniendo esta tecnología ya que las aplicaciones de escritorio para la gestión de bases de datos corporativas están siendo sustituidas por aplicaciones web que permiten reducir los costes de desarrollo, distribución y mantenimiento.

Asimismo, este tipo de desarrollos permite integrar información de diverso tipo (texto, sonidos, gráficos, vídeos, documentos ofimáticos) de un modo intuitivo, sencillo y natural. Tal vez en un futuro no muy lejano la mayoría de las aplicaciones se desarrollen utilizando esta tecnología, ofreciendo entornos de trabajo que dispongan de todo lo necesario (procesadores de texto, hojas de cálculo, agendas, bases de datos) para que el usuario pueda realizar todas sus tareas, utilizando para ello su navegador web.

4.1.2. Lenguajes de programación para la Web

La causa de la evolución de la Web como marco para el desarrollo de aplicaciones, viene motivada por la aparición de lenguajes de programación, dirigidos precisamente a este fin. La Web no sólo aporta páginas estáticas, sino que existen contenidos dinámicos que interaccionan con los usuarios, siendo los lenguajes de programación los que permiten realizar este tipo de creaciones. Existen dos visiones complementarias de la programación en entorno web: los lenguajes de programación del lado del cliente y los del lado del servidor.

4.1.2.1. Programación web del lado del cliente

La primera opción la constituyen los lenguajes de programación del lado del cliente. Se trata de lenguajes que son interpretados por los navegadores, que tienen como objetivo mejorar la interacción entre la página web y el usuario para incrementar la Usabilidad mediante elementos dinámicos, validación de datos introducidos por el usuario, etc. Sin embargo, las diferencias entre navegadores y ciertos aspectos inherentes a la modificación de elementos de una página de forma dinámica hacen que en la práctica el uso de estos lenguajes suponga un problema de accesibilidad. Además, los lenguajes interpretados por el cliente no pueden comunicarse directamente con las aplicaciones del servidor de un modo sencillo. Javascript⁵² es el más conocido de los lenguajes interpretados por los navegadores web (Zakas, 2006).

El W3C se vio obligado a desarrollar un Modelo de Objeto de Documento (Document Object Model, DOM) para estructurar los objetos de una página web según un modelo jerárquico. DOM no es ningún lenguaje de programación sino la forma en la que se estructuran los diferentes tipos de objetos para que posteriormente sean utilizados por un lenguaje de programación, como por ejemplo JavaScript.

Aunque el desarrollo inicial de este modelo coincidió con la creación de JavaScript, la competencia entre distintas compañías hizo que DOM no fuera idéntico en todos los navegadores por lo que el W3C decidió desarrollar una versión unificada para facilitar la interoperabilidad entre navegadores, facilitando con ello la compatibilidad de los desarrollos.

En definitiva, DOM es una Interfaz de Programación de Aplicaciones (API) para documentos HTML y XML. De esta manera, los programadores pueden crear objetos, recorrer la estructura de árbol en la que se organizan, añadir o eliminar contenido y otra serie de tareas. Existen tres niveles DOM para los que se han desarrollado sus respectivas especificaciones aunque la especificación del nivel 3 aún está en desarrollo y está disponible parcialmente. Cada nivel supone un mayor grado de abstracción ya que especifica cómo realizar operaciones más complejas. Mientras que el nivel 1 se centra en aspectos del manejo básico de los documentos XML y HTML el nivel 2 se

⁵² Javascript fué desarrollado en 1995 por Brendan Eich para Netscape Communications. Se incorporó por primera vez en el navegador Netscape Navigator 2.0 y fue adoptado por European Computer Manufacturers' Association (ECMA) bajo el nombre de ECMAScript. Microsoft realizó una versión denominada Jscript para incorporarla a su navegador Internet Explorer.

centra en la gestión total de los diferentes objetos del documento. El nivel 3 está asociado a operaciones de cargar y guardar documentos, validación y aspectos avanzados de XML.

Según Lanker (2005) uno de los aspectos más destacables de la explotación de las posibilidades de DOM con Javascript es DHTML⁵³. No se trata de un lenguaje de programación propiamente dicho, en realidad es una técnica en la que se utiliza el DOM junto con un lenguaje del navegador web (como JavaScript) y las hojas de estilo CSS. El objetivo de esta técnica es modificar el documento HTML/XHTML una vez que ya ha sido cargado por el navegador, aportando al usuario de la página una experiencia de interactividad con diferentes objetos de la página.

Es importante no confundir DHTML con el concepto de *páginas web dinámicas* referido a páginas cuyo contenido está sincronizado con el de recursos de información externos a la propia página (como bases de datos, sindicación RSS, fuentes XML, etc) y que por regla general tiene que ver con aplicaciones que se ejecutan en el servidor. DHTML se suele utilizar para menús desplegados, formularios dinámicos, interacción con objetos (arrastrar, mover) y otras funciones que responden a las acciones del usuario. Otra técnica muy utilizada recientemente en el desarrollo de páginas Web es AJAX⁵⁴. En realidad AJAX combina cuatro tecnologías, tal y como nos indica Barbin (2007):

- XHTML/HTML + CSS para la visualización de la información.
- DOM + JavaScript (o cualquier otro lenguaje de script del cliente)
- El objeto XMLHttpRequest (o en ocasiones el objeto iFrame) para el intercambio de datos de forma asíncrona entre el servidor y el cliente web.
- XML como formato de envío de datos al cliente por parte del servidor (también se puede utilizar otros formatos como XHTML/HTML, texto plano, JSON, etc).

La mayoría de los navegadores utilizados actualmente disponen del objeto

53 DHTML: *Dynamic HTML*. Conjunto de técnicas combinadas y estructuradas para la creación de páginas web y que utiliza HTML, un lenguaje del lado del cliente, hojas de estilo CSS y la jerarquía de objetos DOM.

54 AJAX: *Asynchronous JavaScript And XML*. Es una técnica para crear aplicaciones web que permite crear una comunicación asíncrona con un servidor para actualizar objetos de una página web.

XMLHttpRequest/iFrame. Sin embargo, los navegadores basados en texto, las versiones antiguas de los navegadores gráficos y posiblemente navegadores para personas con necesidades especiales de accesibilidad no disponen de este objeto. Por lo tanto, al igual que sucede con JavaScript hay que dimensionar adecuadamente el alcance de un proyecto de aplicación web y evaluar si es posible, necesario o recomendable la aplicación de AJAX. En cualquier caso, una aplicación web debe desarrollarse de manera que ofrezca alternativas cuando el usuario no pueda utilizar esta tecnología.

AJAX permite actualizar los contenidos de objetos concretos de la página web a partir de los datos solicitados al servidor. El tratamiento de estos datos que se reciben por parte de un lenguaje de script del cliente y el uso conjunto del DOM para modificar los objetos hace innecesaria la recarga de toda la página. Esto hace que una página web sea más eficiente a la hora de validar datos o actualizar ciertos elementos dinámicos y pueda incrementar la usabilidad de un sitio web si se aplica AJAX de forma adecuada. Veamos a continuación un esquema acerca del funcionamiento de AJAX.

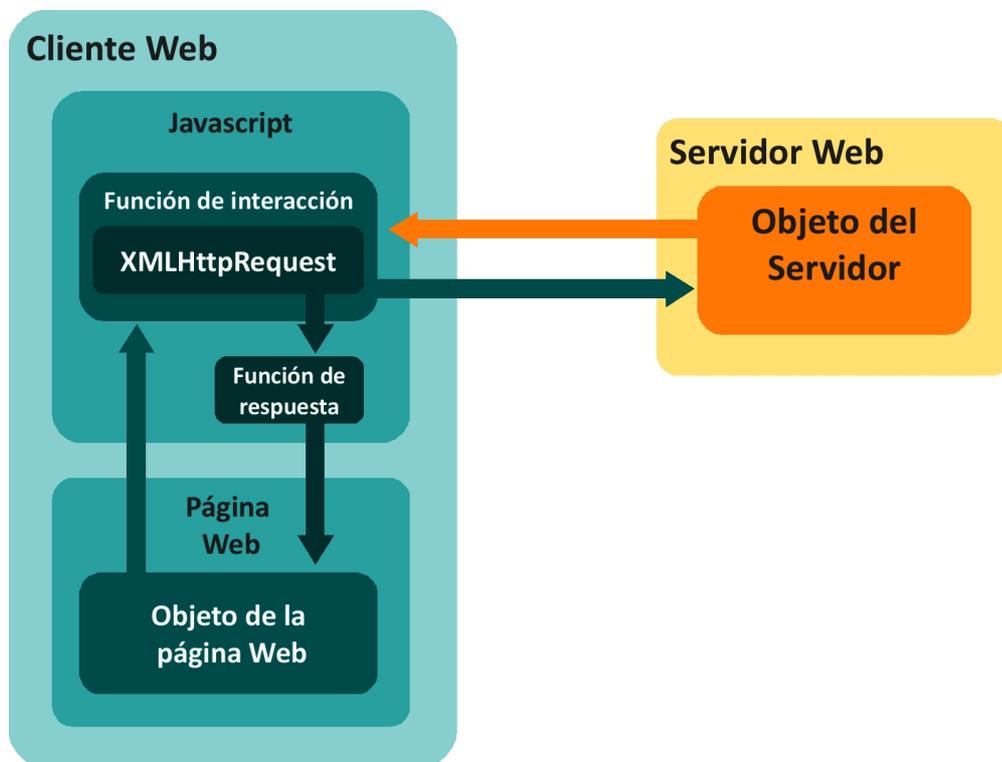


Ilustración 30: Esquema básico de funcionamiento de AJAX

En la página web se referencia a un objeto con un identificador, por ejemplo un campo de formulario. Mediante un lenguaje como JavaScript se crea un objeto XMLHttpRequest utilizado para realizar una llamada a un objeto del servidor a través de una función de interacción que realiza la petición enviando parámetros o datos de un formulario. El servidor devuelve los datos en un formato determinado (XML, XHTML o JSON⁵⁵) y esta respuesta es procesada por una función de respuesta que se encarga de actualizar el objeto referenciado en la página web. En consecuencia, las ventajas que aportan al desarrollo de aplicaciones web los lenguajes de programación del cliente son numerosas. Pese a ello, es necesario tener en cuenta algunas consideraciones:

- La funcionalidad de la aplicación web debe diseñarse sin pensar en el uso de un lenguaje del lado del cliente utilizando para ello HTML/XHTML junto con CSS y las posibilidades que nos ofrezca el lenguaje del servidor utilizado.
- En un segundo paso se pueden aplicar estos lenguajes para dotar de una mayor usabilidad a la página web.
- Hay que proporcionar alternativas para aquellas personas que no tengan disponible la ejecución de lenguajes del lado del cliente, por lo que deberían existir dos versiones con la misma funcionalidad. Hay que pensar que ciertos usuarios no disponen de soporte JavaScript y que con el paso del tiempo se incrementa el número de usuarios que utilizan dispositivos alternativos como videoconsolas, teléfonos móviles o dispositivos con conexión inalámbrica de todo tipo para acceder a la Web.
- Es necesario establecer si en el ámbito de uso de la aplicación que estemos desarrollando existen personas que no puedan utilizar esta funcionalidad. Si se trata de una aplicación para un grupo de trabajo podríamos tomar la decisión de utilizar un lenguaje, como por ejemplo JavaScript, sin ofrecer alternativas si dicho grupo está en disposición de utilizarlo sin problemas. Sin embargo también hay que tener en cuenta la movilidad de las personas o la variación del perfil de acceso de dicho grupo por lo que tomar una decisión ha de hacerse con mucho cuidado.

⁵⁵ JSON: *JavaScript Object Notation*. Se trata de un formato para el intercambio de datos mediante una notación literal de objetos de JavaScript para su ejecución mediante el procedimiento eval(), haciendo innecesario el uso de XML para este propósito.

4.1.2.2. Programación web del lado del servidor

En el apartado anterior se han analizado diferentes tecnologías cuya característica común es su interpretación y ejecución por parte del navegador web. Sin embargo, este modelo de trabajo supone una fuerte dependencia del software cliente y por lo tanto existen ciertas diferencias que hacen que algunos elementos no se comporten del mismo modo en todos los navegadores web.

Utilizar únicamente aplicaciones que se ejecuten en el cliente supondría la imposibilidad de realizar aplicaciones web que utilizaran de forma eficiente recursos ubicados en el servidor en forma de bases de datos. Por ello es necesario disponer de una tecnología que permita acceder a fuentes de datos de forma centralizada y que además garantice un comportamiento idéntico con independencia de las características del software cliente que utilice el usuario.

Una de las principales ventajas que conlleva el uso de lenguajes del lado del servidor es la inmediata integración con sistemas de gestión de bases de datos. Esto conlleva una serie de beneficios obvios derivados de la propia arquitectura cliente-servidor:

- El servidor web es un software cuyo mantenimiento se realiza de forma centralizada.
- El uso de un lenguaje de programación que se ejecute en el servidor permite que el mantenimiento de las aplicaciones web también se realice de forma centralizada. Cualquier actualización del software o aplicación de gestión se realiza en el servidor siendo transparente para el usuario final y su software cliente.
- Los recursos de información también se mantienen de forma centralizada a través de bases de datos gestionadas por SGBD⁵⁶ principalmente relacionales.

El uso de las bases de datos en el entorno web ha seguido una evolución en consonancia a las tecnologías disponibles para el desarrollo de aplicaciones en la Web:

- En una primera fase se mantenían, a través de software de SGBD, bases de datos informativas que posteriormente eran utilizadas para la creación y mantenimiento de páginas web estáticas a través de scripts automáticos que se

⁵⁶ SGBD: Sistemas de Gestión de Bases de Datos.

encargaban de realizar las consultas a las bases de datos y generar las páginas web en un proceso de sincronización que se ejecutaba según una frecuencia temporal establecida.

- En una segunda fase los lenguajes de servidor permitían recoger las necesidades de información de los usuarios a través de formulario y lanzar consultas al SGBD que devolvía una salida que era tratada y devuelta al usuario. También comenzaron a desarrollarse páginas dinámicas cuyo contenido variaba según el contenido de bases de datos concretas.
- En una tercera fase se ha comenzado a utilizar la propia web como interfaz no solo para la consulta de información, sino para el mantenimiento de las bases de datos. Es cuando surgen los CMS y otras aplicaciones web que mantienen el contenido de las bases de datos con las que están integradas.

La aplicación de bases de datos en la Web conlleva una serie de ventajas muy interesantes:

- Permite separar las aplicaciones de los datos con los que trabajan al tiempo que hacen posible la elaboración de especificaciones de elementos mediante estructuras de datos eficientes (siempre y cuando las estructuras de tablas se encuentren normalizadas) con la consiguiente eficacia en el mantenimiento tanto de las aplicaciones como de los datos.
- La gestión de los datos puede realizarse por personas con una formación ofimática básica y que no precisen de conocimientos de programación o diseño de páginas web.
- La información gestionada en bases de datos puede ser reutilizada y presentarse de diversos modos. El uso conjunto de estos datos con XHTML/HTML y CSS permite crear una estructura en donde la información se muestra mediante varias capas: estructura, datos, formato, presentación y aplicación.
- Las bases de datos permiten a los usuarios realizar búsquedas según diferentes criterios cuyos contenidos y especificaciones pueden variar a lo largo del tiempo. Por ejemplo, el usuario no solamente puede cambiar el texto de sus

búsquedas sino que a nivel de aplicación pueden añadirse nuevos campos que sean objeto de la búsqueda.

- Desde el punto de vista de la Web Semántica constituyen un punto de partida, ya que los datos se encuentra en un primer nivel de formalización y por consiguiente pueden reutilizarse para su representación según especificaciones RDF, OWL o de otro tipo.

Existen muchas soluciones de aplicación de bases de datos en entorno web a través de diferentes SGBD. La elección de un software u otro dependerá de la evaluación de diferentes factores como recursos económicos disponibles, número de estructuras de datos con las que se va a operar simultáneamente, requisitos de control de integridad, control automático de eventos en cascada o grado de rapidez de respuesta de la aplicación entre otras.

Algunos de los sistemas de este tipo más utilizados actualmente son:

- **Oracle:** Se trata de un SGBD potente y escalable, aunque muy exigente con los requisitos del servidor donde se ejecute y poco ágil para aplicaciones que necesiten pequeñas bases de datos o tablas informativas. Tiene un buen servicio de soporte y su coste es alto por lo que solo está al alcance de diseñadores encargados del desarrollo de aplicaciones para la gestión de bases de datos corporativas utilizadas en procesos de gestión complejos.
- **MySQL:** Se trata de un software desarrollado por MySQL AB (subsidiaria de SUN Microsystems desde febrero de 2008) pero con un sistema de licencia dual: propietaria y GPL⁵⁷. Puede instalarse en prácticamente todos los sistemas operativos y suele estar incluido en las distribuciones más populares de software servidor web, junto con Apache⁵⁸ y PHP⁵⁹.

57 GPL: *General Public Licence*. Es una licencia creada por la Free Software Foundation, dentro del proyecto GNU, cuyo objetivo es la protección del software de libre distribución, modificación y uso frente a posibles intentos de apropiación. Actualmente está disponible la tercera versión (GPLv3) publicada en 2007.

58 Apache: Servidor web de código abierto y que implementa el protocolo HTTP/1.1 desarrollado inicialmente en 1995. Su última versión (2.2.9) data de junio de 2008.

59 PHP: *PHP Hypertext Pre-Processor*. Es un lenguaje de programación, orientado principalmente al desarrollo de aplicaciones web. Se trata de un lenguaje interpretado que se ejecuta en el servidor, devolviendo los resultados al cliente web.

- **Microsoft SQL Server:** Es un SGBD disponible para sistemas operativos Microsoft Windows. La fuerte dependencia de su sistema operativo limita sus posibilidades de aplicación. Aunque la ejecución de aplicaciones basadas en este software es algo lenta, su potencia basada en la capacidad de gestión de vistas y su integración con los entornos de desarrollo de ASP y ASP.Net⁶⁰ hacen de esta solución una opción óptima para desarrollos bajo sistemas operativos Windows.
- **PostgreSQL:** Es una gran solución, potente y escalable cuya ejecución está soportada en todos los sistemas operativos. Es algo lenta con estructuras de bases de datos sencillas pero su curva de funcionamiento supera a la de MySQL con bases de datos muy grandes o complejas.

Si la aplicación desarrollada no requiere de un sistema de bases de datos muy complejo con un gran número de tablas y fuertemente integrado con PHP y tampoco se necesitan operaciones complejas con índices y vistas, la solución ideal es MySQL. Si los requisitos de la aplicación suponen estructuras complejas de datos y con necesidades avanzadas de control automático de integridad con un bajo coste y que no requieran tiempos de proceso muy rápidos se podría optar por el uso de PostgreSQL.

Aquellos desarrolladores que antepongan como requisito fundamental el soporte del software, que necesiten gran potencia y que también dispongan de una gran infraestructura de servidores pueden escoger Microsoft SQL Server u Oracle como las soluciones óptimas (Burbano Proaño, 2006).

En cualquier caso, una opción óptima suele ser el uso de pequeñas bases de datos informativas con soporte de software libre y ligadas al desarrollo de aplicaciones propias, junto con la integración con el contenido de bases de datos corporativas implementadas con algún tipo de SGBD comercial. Esta integración se realiza a través del uso de XML, como formato para generar la salida de datos en forma de servicios web⁶¹.

60 ASP y ASP.Net: *Active Server Pages*. Lenguaje de programación desarrollado por Microsoft. Al igual que PHP está destinado al desarrollo de aplicaciones web. La versión 3.0 de ASP dio paso a ASP.Net, con una mayor carga de los fundamentos de la programación orientada a objetos, ampliando el marco de desarrollo de aplicaciones más allá de la Web.

61 En este contexto, por *servicio web* entendemos el conjunto de estándares y protocolos, utilizados por una aplicación web para el intercambio de datos en formato XML. Generalmente el intercambio

ODMG⁶² publicó la especificación ODMG 3.0 para el desarrollo de bases de datos bajo el paradigma de la programación orientada a objetos (ODMG, 2000). Se están comenzando a desarrollar SGBD orientados a objetos que superan los problemas asociados a la persistencia de objetos y la interoperabilidad de datos. Algunos sistemas están totalmente basados en este paradigma como Orient. Otros han adoptado alternativas híbridas como Virtuoso Universal Server, o la extensión que incorpora Oracle. ODMG se disolvió en 2001 y los trabajos fueron retomados por OMG⁶³ en 2004 formando dos años más tarde un grupo de trabajo para el estudio y desarrollo de las tecnologías de bases de datos orientadas a objetos. Por tanto, se trata de una tecnología aún en desarrollo. La realidad actual muestra que los SGBD más utilizados siguen siendo los que están basados en el modelo relacional de bases de datos.

Al igual que ocurre con los SGBD existen varias alternativas en lo referente a lenguajes de programación para la Web. Nos centraremos en las dos soluciones más utilizadas actualmente (Jabba Molinares, 2004):

- PHP es el paradigma actual de programación en la Web. Se trata de un lenguaje creado en 1994 por el programador de origen danés Rasmus Lerdorf, aunque las bases actuales de este lenguaje fueron establecidas por los programadores israelíes Zeev Suraski y Andi Gutsman en 1995 con PHP 3 (Sklar, 2005). Este lenguaje, en un principio era de propósito general pero que se ha extendido ampliamente en el desarrollo de aplicaciones web.
- JSP (Java Server Pages) es una tecnología basada en el lenguaje de programación Java⁶⁴ y desarrollado por Sun Microsystems, entre 1999 y 2001, año en el que se liberó la versión 1.2 de JSP. Hoy día está disponible la versión 2.1). Se trata de un lenguaje compilado en tiempo de ejecución.

de datos sigue un esquema de petición/respuesta.

62 ODMG: *Object Database Management Group*. Es un grupo de trabajo, organizado en 1991 por Rick Cattell, ingeniero jefe del grupo de tecnologías de bases de datos de Sun Microsystems. Se disolvió en el año 2001 tras la publicación de la especificación ODMG 3.0 y para concentrarse en la especificación de datos de Java.

63 OMG: *Object Management Group*. Consorcio de empresas fundado en 1989, por Apple, Hewlett-Packard, IBM o Sun Microsystems entre otras y con sede en Needham, Massachusetts. Su objetivo es el desarrollo de estándares relativos a tecnologías orientadas a objetos.

64 Java: Lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems en los años 90, cuyo código se compila en un formato apto para su ejecución en una máquina virtual.

Hay que resaltar el enorme auge del que goza PHP al amparo del desarrollo de las aplicaciones de software libre. Se trata de un lenguaje que destaca por su integración con MySQL y su facilidad de aprendizaje. Utiliza una sintaxis muy parecida a la del lenguaje C con instrucciones orientadas principalmente a la programación estructurada, uso de funciones, manejo de librerías externas e incluso funcionalidades de programación orientada a objetos que han sido totalmente reescritas en PHP 5. Es ideal para la creación de páginas web con contenido dinámico y para pequeñas aplicaciones web. Es una solución muy utilizada en entornos que integran Apache, PHP y MySQL. Puede utilizarse en la mayoría de sistemas operativos por lo que se considera un lenguaje multiplataforma.

Una de las desventajas de PHP es la ausencia de capas de abstracción para ciertos aspectos avanzados del desarrollo de aplicaciones web, principalmente el manejo de sesiones, las conexiones a bases de datos y la autenticación y gestión de sesiones de forma integrada. Hasta la versión PHP 5 se echaba en falta el manejo de datos XML de un modo simplificado, aunque esta situación ha mejorado considerablemente.

La sencillez y flexibilidad del uso de PHP ha conllevado su aplicación en el diseño de aplicaciones mal estructuradas y con un nivel de modularización y estructuración nulo. Por ello es recomendable realizar un análisis de la estructura de la aplicación y crear las librerías necesarias de un modo bien estructurado, en el que no se mezclen aspectos de la interfaz con aspectos funcionales de la aplicación (Welling, 2005). Algunos entornos de desarrollo como ECLIPSE⁶⁵ permiten incorporar módulos para trabajar de forma ordenada y estructurada con PHP.

JSP es una tecnología basada en el lenguaje de programación Java cuya orientación es en cierto sentido similar a la de PHP. Es decir, JSP permite crear páginas web con contenidos dinámicos mediante la programación de scripts y posee cierto grado de abstracción ya que ofrece una serie de etiquetas predefinidas para realizar ciertas acciones muy comunes en el funcionamiento de aplicaciones web. El grado de exigencia en el desarrollo de aplicaciones Java es bastante alto. Sin embargo esta exigencia se ve compensada por la eficiencia, reusabilidad y facilidad de mantenimiento de las aplicaciones basadas en Java. Para la ejecución de programas

⁶⁵ Eclipse: Aplicación de software libre, cuyo objetivo es el de proporcionar un entorno de desarrollo integrado, que ofrece una serie de herramientas para aumentar la productividad del programador y facilitar el mantenimiento de aplicaciones. Puede adaptarse para su uso en diferentes entornos de programación. Su sitio web es: <http://eclipse.org>

Java es necesario disponer de JRE⁶⁶ para instanciar lo que se denomina “*máquina virtual Java*” ya que los programas Java se compilan para su posterior ejecución en dicha máquina virtual⁶⁷.

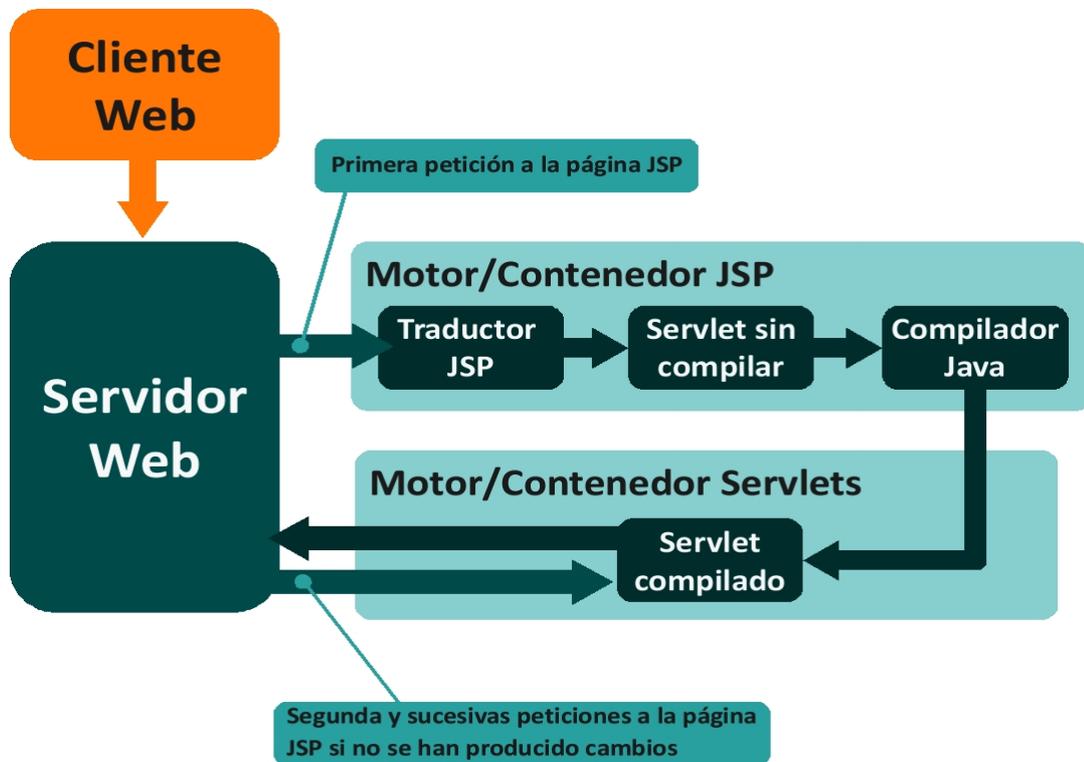


Ilustración 31: Esquema básico de funcionamiento de la tecnología JSP

De esta manera se consiguen aplicaciones eficientes que funcionan en múltiples sistemas operativos, basta con tener instalado la correspondiente versión de JRE adaptada al sistema operativo. Por tanto, Java se trata de un lenguaje multiplataforma, una característica que para Eckel (2007) es, tal vez, la clave del éxito de JSP y de otros lenguajes como PHP.

Para ejecutar aplicaciones basadas en JSP, además de un servidor Web y de JRE hay

66 JRE: *Java Runtime Environment*. Entorno que incluye todo un elenco de aplicaciones destinadas a la ejecución de programas Java. Una máquina virtual Java es en realidad una instancia de JRE.

67 Una máquina virtual Java es una aplicación para ejecutar aplicaciones compiladas en código binario Java (denominado Java bytecode) generado por un compilador Java. Cada sistema operativo dispone de su propia instalación de máquina virtual Java, pero el código de los programas Java puede ejecutarse en cualquiera de estas máquinas sin ningún tipo de modificación.

que disponer de un servidor/contenedor de servlets⁶⁸. Hay que aclarar que los scripts JSP y los servlets son lo mismo ya que para ejecutar un scripts JSP se realiza una compilación del script para obtener un servlet que es el que se ejecuta sobre la máquina virtual Java.

La tecnología Java es infinita, incluso existen plataformas como Java EE (Enterprise Edition) integradas con software de gestión de bases de datos y otras aplicaciones que permiten desarrollar y ejecutar aplicaciones Java de todo tipo incluidas las que están orientadas a su explotación en la Web. JSP y la tecnología Java en general, son un mundo fascinante: potente, modular, escalable, aplicaciones compactas y sólidas, fáciles de mantener y que obligan a un desarrollo estructurado y planificado. Sin embargo se trata de una tecnología compleja en cuanto a su aprendizaje y muy exigente en cuanto a la infraestructura y organización del servidor. Esta exigencia también puede contemplarse como una garantía de desarrollo ordenado y metódico de las aplicaciones web.

4.1.3. Servicios web

El concepto de servicio web ofrece una solución para el intercambio de información entre distintos sistemas, independientemente de la plataforma en la que se ejecuten, o la naturaleza de su desarrollo. Un servicio web es un sistema de software cuya finalidad es la de ofrecer soporte a la interoperabilidad entre máquinas a través de red (W3C, 2004d). Generalmente estos servicios disponen de una interfaz que utiliza un formato determinado para la descripción de los datos.

La base de la tecnología de los servicios web son XML y HTTP. El lenguaje XML proporciona la base sintáctica para la descripción de los datos transmitidos en el proceso de intercambio. HTTP ofrece un protocolo para la recepción y envío de datos, que encuentra menos restricciones en el ámbito de las infraestructuras de seguridad y cortafuegos, ya que se suele bloquear la mayoría de los puertos de comunicaciones, dejando libres, por regla general, los puertos destinados a las transmisiones web.

⁶⁸ Servlets: Programas escritos en Java, compilados y preparados para su ejecución en una máquina virtual. Los servlets son gestionados por una aplicación denominada contenedor de servlets cuyo exponente más representativo es Tomcat.

Tres son los estándares principales utilizados por los servicios web: WSDL⁶⁹, SOAP⁷⁰ y UDDI⁷¹. La integración de estos estándares para la configuración de servicios web se realiza en las siguientes fases (García Sánchez, 2007):

- Registro del servicio mediante UDDI para que los servicios web puedan ser localizados y utilizados por usuarios y máquinas.
- Búsqueda del servicio web más adecuado a las necesidades del cliente, para lo que se utiliza el registro UDDI. En este proceso se especifican las funcionalidades deseadas y se obtiene una relación de las descripciones de los servicios web (WSDL).
- Una vez que el cliente dispone de todos los ficheros WSDL de los servicios a los que necesita acceder se determina el más apropiado según las preferencias del cliente
- Petición de ejecución del servicio al proveedor del mismo a través de un mensaje SOAP.
- Ejecución del servicio según los parámetros indicados en el mensaje SOAP del cliente y devolución de los resultados en forma de otro mensaje SOAP.

Esta arquitectura de trabajo se denomina SOA⁷² y permite el uso de recursos de red sin que sea necesario conocer cómo están organizados a nivel interno y sin que influya la plataforma sobre la que opera el cliente.

Por contra, presenta el gran inconveniente del bajo rendimiento de ejecución, ya que al utilizar XML, debe ser debidamente procesado por la aplicación, lo que requiere una carga de cómputo mucho mayor que la que ofrecen otras alternativas de programación distribuida.

69 WSDL: *Web Service Description Languages*. Especificación XML utilizada para la descripción de servicios web.

70 SOAP: *Simple Object Access Protocol*. Protocolo que describe la forma en la que dos objetos distribuidos pueden comunicarse usando XML.

71 UDDI: *Universal Description Discovery and Integration*. Registro para la publicación de forma normalizada de documentos WSDL con las descripciones de los servicios web, con la finalidad de poder realizar búsquedas de los mismos.

72 SOA: *Service Oriented Architecture*. Metodología y marco de trabajo para el desarrollo y utilización de servicios web.

Para solventar estos problemas, el W3C está trabajando en el intercambio eficiente de XML obteniendo una codificación más compacta de este lenguaje⁷³. Además la Web Semántica ofrece un campo muy fértil para el desarrollo de los servicios web, automatizando aspectos tales como el descubrimiento, selección, composición e invocación de los mismos mediante ontologías y metadatos. Por todo lo anterior pensamos que el uso de servicios web en nuestro modelo de aplicación resulta factible especialmente para el diseño de las funciones de intercambio de información.

4.1.4. Los sistemas de gestión de contenidos

El concepto de la Web como cliente/servidor universal para el mantenimiento de contenidos informativos resulta esencial para el desarrollo de la Web Ubicua. La Web podría describirse como un cristal que ofrece una doble visión, de un lado la perspectiva del usuario, para quien un sitio web debe ofrecer contenidos dinámicos, estructurados e integrados mediante una interfaz homogénea con sistemas de navegación coherentes, de otro la que poseen los administradores y gestores del sitio, agrupados en torno a una tipología de perfiles de usuarios y con funciones de gestión de información delimitadas y estructuradas. También es de aplicación en estos sistemas ciertas características relacionada a los sistemas que incorporan funcionalidades de trabajo en grupo y flujo de trabajo.

Para Pérez-Montoro (2006), los Sistemas de Gestión de Contenidos (CMS⁷⁴) son un conjunto de aplicaciones web integradas en una interfaz de usuario, transmitiendo al usuario la idea de que se encuentra ante una única aplicación, y proporcionando todo tipo de funciones para la estructuración, gestión y comunicación de información a través de una serie de procesos y herramientas.

En definitiva, los CMS permiten la gestión de información y su publicación por parte de personas que no precisan de conocimientos de HTML o XHTML, hojas de estilo CSS o cualquier otra tecnología utilizada en la Web. En ocasiones los usuarios que alimentan el sistema ni tan siquiera deben ocuparse de crear la estructura del sitio web sino categorizar o etiquetar la información, delegando en el sistema las tareas de

73 Se han creado dos grupos de trabajo en el seno del W3C: *XML Binary Characterization Working Group* (<http://www.w3.org/XML/Binary/>) y *Efficient XML Interchange Working Group* (<http://www.w3.org/XML/EXI/>).

74 CMS: *Content Management Systems*. Utilizaremos en este apartado las siglas en inglés para referirnos a este concepto.

construcción de los correspondientes sistemas de navegación y de recuperación de información (Ambite, 2006).

Una visión más concreta, y en cierto modo pragmática, es la ofrecida por Cuerda y Minguillón (2004) para quienes los CMS tienen como funciones principales la creación, gestión, publicación y presentación de contenidos. Esto es totalmente compatible con la estructura de subsistemas de un CMS descrita por Boiko (2002):

- Subsistema de colección. Permite la creación, estructuración, adquisición, conversión, agregación y sindicación de recursos de información.
- Subsistema de gestión. Desarrolla las funciones asociadas a los ficheros y bases de datos, control de flujo de trabajo y configuración del sistema.
- Subsistema de publicación. Encargado de los aspectos relacionados con la presentación, reutilización y definición de servicios de sindicación de contenidos.

A nuestro parecer resulta destacable la automatización en los procesos de reutilización de contenidos. La sindicación de contenidos, fundamentalmente a través de RSS, acerca estos sistemas a la Web Semántica, junto con la filosofía de separación entre contenido, estructura y presentación implícita en los CMS.

Según la tipología de sistemas de gestión de contenidos propuesta por Tramullas y Garrido (2006), nuestra propuesta se enmarcaría dentro de los CMS orientados a la creación de entornos de colaboración. Por tanto, el modelo de aplicación ha de incorporar aspectos inherentes a los sistemas de flujo de trabajo y trabajo en grupo. Esto permitiría la coordinación de una comunidad de usuarios estructurada en grupos de trabajo a los que se asociarían determinados permisos de gestión. Se trataría de un CMS especializado en la gestión de esquemas conceptuales y la indización de recursos de información organizados en repositorios.

Finalmente es necesario recordar lo indicado en el apartado 3.7, por el que nuestra propuesta se integraría, a través de procesos de sindicación o explotación de servicios web, dentro de los CMS de tipo general, como herramienta para la elaboración de esquema conceptuales orientados a la categorización de contenidos o el diseño de sistemas de navegación.

4.1.5. Accesibilidad, Usabilidad y Arquitectura de la información

Para la consecución de aspectos relacionados con la ubicuidad de cualquier sitio o aplicación web, es necesario considerar ciertos criterios de Accesibilidad, Usabilidad y Arquitectura de la información para la Web que abordamos en el presente apartado.

En el ámbito del diseño de sitios web, la Accesibilidad se refiere al grado con el que puede utilizarse un sitio web con independencia de sus capacidades físicas y técnicas. Por lo tanto no hay que limitar el ámbito de aplicación de la accesibilidad al de aquellas personas con discapacidades (visuales, auditivas, motrices o cognitivas). La accesibilidad va mucho más allá, implicando la interoperabilidad de la propia información. La accesibilidad tiene su aplicación en el desarrollo de sitios web de forma que los usuarios no se vean en la imposibilidad de acceder a la información debido al software de navegación o los dispositivos de entrada/salida empleados.

La accesibilidad ha sido relegada, generalmente, a un segundo plano en los procesos de desarrollo de aplicaciones web. En muchas ocasiones estas herramientas solo eran utilizadas por un grupo de usuarios, cuyos miembros eran conocidos por el equipo de desarrollo, por lo que conocían sus limitaciones y su entorno de trabajo. El modelo de gestión de contenidos ha cambiado esta realidad. Las aplicaciones web se diseñan para un usuario cuyo perfil, habilidades y capacidades se desconocen. La Accesibilidad ofrece unas líneas de trabajo para obtener un producto cuyo uso será independiente de las circunstancias del usuario final.

Las pautas de accesibilidad promovidas por el WAI⁷⁵ se refieren al contenido Web (WCAG, *Web Content Accessibility Guidelines*) a las herramientas de autor (ATAG, *Authoring Tool Accessibility Guidelines*) y a los agentes de usuario (UAAG, *User Agent Accessibility Guidelines*). Para los desarrolladores de sitios web las más importantes a tener en cuenta son las referidas al diseño de contenido. Puede resultar de interés la consulta del resto de recomendaciones (ATAG y UAAG), sin embargo para la finalidad del presente trabajo nos ceñiremos a la guía WCAG 2.0 (W3C, 2008b1) que supone la línea de trabajo más avanzada del WAI. Esta guía agrupa las recomendaciones de accesibilidad en cuatro niveles, o capas, de orientación, relativas a que el contenido web sea:

⁷⁵ WAI: *Web Accessibility Initiative*. Es un grupo de trabajo tutelado por el W3C que centra su ámbito de actuación en el desarrollo de trabajos relativos a la accesibilidad web.

- **Perceptible:** La información ha de presentarse a los usuarios en un modo que pueda visualizarla. En esta capa se agrupan aspectos tales como que las imágenes tengan texto alternativo, proveer al usuario de información textual alternativa a los elementos multimedia, que los contenidos puedan presentarse de formas distintas sin pérdida de información o estructura o que se distingan fácilmente los elementos informativos.
- **Operable:** Los componentes y la navegación de la interfaz de usuario deben funcionar durante la consulta del contenido. Además del ratón, el teclado del usuario debe estar operativo y poder realizar todas las funciones de navegación y consulta desde el mismo. La consulta del contenido no debe contener elementos que puedan suponer un riesgo potencial para la salud del lector (como por ejemplo, flashes intermitentes). También deben incluirse ayudas a la navegación, que ayuden a encontrar la información y aporten contexto.
- **Comprensible:** La información y el funcionamiento de la interfaz deben ser comprensibles para el usuario. En este nivel se aborda la legibilidad e inteligibilidad del contenido textual, que las páginas se comporten según espera el usuario y ayuden al usuario a evitar o corregir errores.
- **Robusto:** Las sintaxis del código utilizado para escribir las páginas (XHTML, HTML, CSS, Javascript, etc) debe ser correcta, lo que asegurará la compatibilidad con diferentes navegadores (agentes de usuario) y garantizará la adaptación de contenidos a tecnologías futuras.

Las diferentes pautas de cada nivel se organizan a su vez en una serie de niveles de conformidad:

- Nivel de conformidad "A": La página debe cumplir una serie de requisitos sin los cuales ciertos grupos de usuarios **no podrían** acceder a la información del sitio Web.
- Nivel de conformidad "doble A": La página debe cumplir una serie de requisitos sin los cuales sería **muy difícil** acceder a la información para ciertos grupos de usuarios.
- Nivel de conformidad "triple A": La página debe cumplir una serie de requisitos

sin los cuales algunos usuarios experimentarían **ciertas dificultades** para acceder a la información.

La accesibilidad es uno de los motivos por los que XHTML es tan importante en el desarrollo de la Web. XHTML exige un nivel de corrección sintáctico mucho mayor que HTML. Es la herramienta ideal para crear documentos cuya visualización sea idéntica en cualquier navegador, siempre y cuando el cliente interprete correctamente las especificaciones XHTML y CSS.

El coste inicial de plantear cualquier desarrollo web accesible se ve compensado por una serie de beneficios:

- Incremento del número de potenciales usuarios de un sitio web: cuando una página web es accesible no presenta barreras que dificulten su acceso, independientemente de las condiciones del usuario. Una página web que cumple los estándares es más probable que se visualice correctamente en cualquier dispositivo con cualquier navegador.
- Optimización de los costes de desarrollo y mantenimiento: una página web accesible respecta los estándares XHTML/HTML y CSS, por lo que es menos propensa a contener errores y por tanto más sencilla de actualizar.
- Reducción del tiempo de carga de las páginas web: la separación del contenido informativo y los estilos visuales permite desarrollar código XHTML más eficaz y optimizado. Esto también permite al cliente web una interpretación más eficaz de las páginas que recupera.
- Permite la ubicuidad del usuario: las aplicaciones web, desarrolladas bajo criterios de usabilidad, pueden ser utilizadas por los usuarios con independencia del navegador, plataforma o sistema operativo con el que opere el dispositivo de consulta. Esto facilita la movilidad de trabajadores y personas en general, desde el punto de vista del acceso a sistemas y procesos.

Otro aspecto a tener en cuenta en el desarrollo de sitios web es la Usabilidad. Generalmente este concepto se asocia a la interacción entre personas y máquinas, estudiando esta realidad desde diversos puntos de vista: social, psicológico, ergonómicos, comunicativos, técnicos, etc.

Existen normas que tratan de formalizar estos aspectos. La norma ISO/IEC 9126-1 contempla la usabilidad como uno de los componentes del modelo de calidad para el desarrollo de software (ISO/IEC, 2001), Las normas ISO 9241-11 e ISO 9241-12 (ISO, 1998a y 1998b) abordan la Usabilidad asociada a la ergonomía para la visualización de información. La norma ISO/TR 16982 (ISO, 2002a), para el diseño de sistemas interactivos, incorpora la Usabilidad, a partir de un diseño centrado en el usuario. De gran interés son las normas sobre la ergonomía de interfaces ISO 14915 (ISO, 2002b, 2003 y 2002c) que abordan aspectos de diseño, navegación, control e integración de elementos y componentes multimedia. El diseño de procesos según criterios de Usabilidad también han sido objeto de normalización en ISO/13407 e ISO/TR 18529 (ISO, 1999 y 2000).

Precisamente la norma ISO 9241 ofrece una definición de Usabilidad bastante clara, según la cual se trata del *“rango en el que un producto puede usarse por un grupo específico de usuarios, para alcanzar ciertos objetivos definidos, con efectividad, eficiencia y satisfacción en un contexto de uso concreto”*.

Para Cobo Romaní (2005), la Usabilidad aplicada a la Web plantearía el análisis y mejora de la calidad de los procesos que intervienen en la interacción del usuario con la Web. Implicaría un compromiso de desarrollo en la calidad de la interacción entre usuario y sistema, en un modelo de mejora continua. Esto se puede llevar a cabo con técnicas de evaluación, lo que conllevaría una eficiencia creciente del sistema y la consiguiente reducción de errores y de tiempo de aprendizaje en el uso del sistema, culminando en una mejora en la productividad y aceptación del mismo.

Para Granollers (2004), la aplicación de criterios de usabilidad en el diseño de páginas web se justifica por los beneficios obtenidos tanto para la organización, que podrá realizar un mejor marketing, como para los usuarios, que ven como se reduce el esfuerzo requerido para interactuar con la interfaz del sistema.

Existen múltiples iniciativas para la puesta en marcha de procesos, metodologías y técnicas de implantación de la usabilidad en el núcleo de desarrollo de contenidos web. Podría decirse que quien difundió esta idea fue Nielsen, planteando la Ingeniería de la Usabilidad (Nielsen, 1993) y conformando las bases de esta disciplina en una serie de áreas de actuación que deben realizarse durante el diseño, desarrollo y mantenimiento de un sitio web (Nielsen, 2002a y 2002b). Para este mismo autor el nivel de Usabilidad puede definirse en cinco atributos esenciales (Nielsen, 1993):

Facilidad de aprendizaje, Eficiencia, Retención sobre el tiempo, Tasas de error por parte de los usuarios y Satisfacción subjetiva.

Otros trabajos fortalecen el modelado de la Usabilidad, a través de propuestas con una fuerte carga metodológica (Granollers, 2004):

- Método DUTCH⁷⁶. Creado por el grupo de investigadores de la sección de Gestión de la Información y Ingeniería del Software de la Universidad de Vrije (Amsterdam). Considera tres puntos de vista para modelar el entorno funcional de una aplicación: Las personas (usuarios y grupos), el trabajo y la situación. Las actividades de estos puntos de vista se estructuran en dos modelos: El Modelo de Tareas 1 (situación actual) y el Modelo de Tareas 2 (cambios y situación futura). Es un modelo basado en la iteratividad y una retroalimentación constante (Chisalita, Puerta y Veer, 2003).
- Ciclo de vida de la Ingeniería de la Usabilidad. Desarrollado por un grupo de trabajo dirigido por Mayhew (1999). Es una aproximación al diseño de sistemas usables, que estructura el trabajo en tres fases: Análisis de requisitos, Diseño-prueba-desarrollo e Instalación. Esta propuesta plantea algunas ideas muy interesantes, como la integración de los procesos de Ingeniería del Software con la Usabilidad a medida del proyecto, el reconocimiento de la amplitud de técnicas aplicables, la estructuración del trabajo en subconjuntos de funcionalidades o la compensación del esfuerzo con el producto final.
- Desarrollo basado en escenarios. Este método creado por Rosson y Carroll (2002) se estructura en un marco de trabajo de tres fases: Análisis (donde se estudian los escenarios del problema), Diseño (donde intervienen los escenarios de la actividad, la información y la interacción) y Prototipado/Evaluación. El flujo de trabajo es flexible y se puede pasar de una fase de forma intercalada.
- Modelo de Proceso de Usabilidad Pervasiva. Se trata de un modelo de usabilidad (Brink, Gergle y Wood, 2002) orientado exclusivamente a la Web, mientras que las anteriores propuestas pueden aplicarse a cualquier tipo de

⁷⁶ DUTCH: Siglas en inglés para *Designig for Users and Tasks from Concepts to Handles*. Curiosamente la palabra “dutch” significa “holandés”, que es la nacionalidad del grupo de trabajo que creó este método.

desarrollo. Organiza el trabajo en cinco fases, en las que la evaluación está totalmente integrada: Análisis de requisitos, Diseño conceptual, Maquetas y prototipos, Producción y Lanzamiento.

Un tema recurrente en los últimos años es el desarrollo y aplicación del concepto de la Usabilidad a partir de diseños basados (o centrados) en el usuario. La norma ISO 13407 enfoca este concepto dentro del análisis de las necesidades concretas de los usuarios con el objetivo de construir o diseñar sistemas adaptados a las mismas. La comprensión del usuario final, su realidad funcional, técnica y, en ocasiones, cognitiva rigen los procesos de diseño centrados en el usuario.

El análisis del entorno y motivaciones del usuario, así como la aplicación continua de técnicas de retroalimentación y evaluación suponen las principales herramientas de trabajo en este tipo de desarrollos (Albert, Goes y Gupta, 2004). Por tanto, la experiencia de funcionamiento del sistema y la definición de canales de comunicación que hagan que los propios usuarios participen en su mejora son fundamentales. Pensamos que realmente este tipo de diseños subyacen a los propios principios de la Usabilidad, tratándose más de formalizar la participación del usuario final, que de una nueva orientación de esta disciplina.

De todo lo anterior es posible inferir una serie de reflexiones acerca de la Usabilidad en el desarrollo de sitios y aplicaciones web:

- Considerar la web como un medio de comunicación hipertextual situando a los elementos multimedia como un complemento a la información textual, debiendo utilizarse de forma adecuada y pertinente de manera que aporte significado de valor añadido a la información textual.
- Aumentar al máximo el espacio visual destinado al contenido informativo, aplicando diseños que faciliten al usuario la separación visual entre contenido informativo y sistemas de navegación.
- Ser exigentes en la aplicación de los criterios de accesibilidad, algo esencial teniendo en cuenta la diversidad de dispositivos, resoluciones, navegadores y dificultades de accesibilidad por parte de los usuarios.
- Redacción adecuada y estructurada de los documentos web que además deben

ser claros, concisos y breves. Es necesario utilizar adecuadamente los elementos de marcado que nos permiten estructurar adecuadamente los contenidos: encabezados, párrafos, listas numeradas y no numeradas, listas de definición, etc.

- Los sistemas de navegación han de ser claros en su presentación y ámbito de aplicación al tiempo que deben ofrecer una visión global de la organización de contenidos del sitio web. Han de ser intuitivos y basarse en un esquema de organización descendente y fácil de recordar en sucesivas visitas al sitio web.
- La organización jerárquica de un sitio web debe guardar un equilibrio entre profundidad y amplitud de la estructura informativa, contemplando tanto los diferentes perfiles de usuarios como una clasificación temática clara e intuitiva.
- Utilizar preferentemente documentos web para la presentación de la información, utilizar formatos compactos (como PDF) únicamente en documentos que precisen de una impresión muy precisa o documentos de gran tamaño cuya lectura en la Web puede presentar dificultades.
- Ofrecer información sobre la ubicación del usuario dentro de la estructura informativa mediante el uso adecuado de logotipos y cabeceras.
- Adaptar los mecanismos de acceso, contenidos y el lenguaje de la información según el perfil del usuario.
- El usuario debe tener la sensación de que tiene control total sobre los procesos de consulta y navegación sin que, por ejemplo, se realicen acciones no solicitadas de forma explícita.
- El aspecto y funcionalidad de elementos similares que se repitan en diferentes páginas del sitio deben ser iguales.
- Los errores cometidos durante los procesos de navegación, búsqueda o cumplimentación de formularios deben ser reversibles.
- No incluir elementos innecesarios que no aporten ningún tipo de función informativa, de navegación, de localización o de mejora de experiencia del usuario.

- Sistema de ayuda eficaz que permitan la consulta de instrucciones de uso del sitio web y de asistencia al usuario.

Es muy interesante el comentario efectuado por Granollers (2004) observando que algunos autores *“confunden la Evaluación de la Usabilidad de un sistema o aplicación software con el desarrollar la misma siguiendo alguno de los métodos de la Ingeniería de la Usabilidad”*. Coincidimos plenamente con las tesis de autor al aclarar que *“la Ingeniería de la Usabilidad es una aproximación metodológica que permite desarrollar aplicaciones interactivas con el parámetro de la facilidad de uso o usabilidad como objetivo preferente”* y que la evaluación de la usabilidad *“constituye sólo una parte de la Ingeniería de la Usabilidad —que incluye un rango de métodos que una vez aplicados permiten conocer si dicho sistema, interfaz o prototipo, es o no usable— que no conlleva ni mucho menos haber seguido un proceso sistemático para garantizar la usabilidad del mismo”*.

El tercer aspecto analizado en este apartado es el de la Arquitectura de la Información. Se trata de una evolución de las técnicas metodológicas para la gestión de información, debido a la sobreabundancia, desorganización, heterogeneidad y complejidad de la misma. Es una disciplina surgida antes de la aparición de la Web, que para Bryson (1997) se desarrolla al amparo de organismos y empresas con necesidades concretas de optimización de los procesos y sistemas de información, alineados con las políticas generales de la organización.

Según Vitruvio⁷⁷ los tres pilares de la Arquitectura son *utilitas, firmitas y venustas* (utilidad, firmeza y belleza). La analogía puede trasladarse al ámbito de la Web. Al igual que en la Arquitectura, donde el objetivo es la distribución de los espacios de la manera más armónica posible para aprovechamiento de éstos y comodidad del usuario, la Arquitectura de la Información para la Web intenta lo mismo; crear espacios llenos de contenidos de acceso sencillo e interrelacionados entre sí, de manera que el usuario encuentre rápidamente lo que busque, le sea de utilidad y lo encuentre agradable y atractivo de consultar (Tosete, 2007).

Para Rosenfeld y Morville (2002) la Arquitectura de la Información se define en una

⁷⁷ Marco Vitruvio Polión fue un arquitecto, ingeniero y tratadista romano del siglo I a.C. Trabajó como ingeniero para los ejércitos de Julio César y como arquitecto civil para Augusto. Es el autor del tratado sobre Arquitectura más antiguo que se conserva y el único de la Antigüedad clásica: *De Architectura*.

triple vertiente que contempla su estructura, función y objeto: *“1.- La combinación de organización, etiquetado y esquemas de navegación dentro de un sistema de navegación. 2.- El diseño estructural de un espacio de información que facilite la finalización de tareas y el acceso intuitivo a los contenidos. 3.- El arte y la ciencia de estructurar y clasificar los sitios web y las Intranets para ayudar a las personas a encontrar y gestionar información”*.

La anterior definición nos mueve a la reflexión, ya que con la aparición de la Web, la Arquitectura de la Información cobra una mayor relevancia. El modelo de acceso a la información de la Web, basado en el hipertexto, complica los procesos de organización. El diseño de las estructuras organizativas de la información ha de aprovechar los esquemas de acceso, para lo cual hay que diseñar cuidadosamente sistemas de navegación que permitan a los usuarios localizar la información de una forma rápida y sencilla. Tampoco hay que olvidar el entorno en el que se desarrollan las actividades de la Arquitectura de Información, relacionando usuarios y contenidos dentro de un determinado ámbito o contexto.

La primera propuesta de metodología formal de aplicación de Arquitectura de la Información en la Web fue realizada también por Rosenfeld y Morville (2002), según la cual los principales componentes de un sitio son:

- Organización de la información. Creación de estructuras de organización de información (alfabéticas, cronológicas, geográficas, funcional, por perfiles, metafóricas, temáticas, híbridas) y definición de las estructuras de acceso que se van a utilizar (jerárquicas, bases de datos, hipertexto).
- Diseño de los sistemas de navegación. A partir de la organización de la información se crean los sistemas de navegación (globales o locales) que orientarán y permitirán al usuario desplazarse entre los contenidos del sitio o aplicación web., creando distintos tipos de sistemas de navegación de diverso tipo.
- Elaboración de los sistemas de rotulación, utilizados para representar la información a través de un lenguaje adecuado para la Web.
- Sistemas de búsqueda. Son elementos que complementan los sistemas de navegación, permitiendo la localización en sitios web de gran tamaño.

Esta metodología se aplica desde el desarrollo de un proceso planificado, en donde la evaluación, el análisis organizativo y de usuarios, así como un minucioso diseño conceptual, integran los componente enumerados anteriormente.

Al igual que con la Usabilidad, existen iniciativas que plantean las actividades de la Arquitectura de la Información centradas en el usuario. Garret (2002) plantea cinco niveles, dentro de un marco general que integra por un lado la Arquitectura de la Información y por otro la evaluación de la Usabilidad: Estrategia, Alcance, Estructura, Esqueleto y Superficie. Todo ello considerando el doble papel de la Web como interfaz de software y sistema de hipertexto.

También resulta de gran interés la observación que realiza Gilchrist (2003), para quien resulta esencial la naturaleza terminológica de la Arquitectura de la Información. Este hecho se constata a través de la aplicación de taxonomías u otras herramientas terminológicas en la organización de contenidos, en el diseño de sistemas de navegación, e incluso para relacionar procesos, personas y herramientas.

A nuestro parecer es necesario hacer un enfoque común que tenga en cuenta Accesibilidad, Usabilidad y Arquitectura de la Información. Algunos autores, como Tramullas (2004), ya apuntan en este sentido al afirmar que *“la finalidad de la Arquitectura de la Información es crear espacios y productos de información digital que satisfagan las necesidades de información del usuario. La de la Usabilidad es la de asegurar la facilidad de uso de los sistemas y de las interfaces de usuario; en consecuencia, la convergencia ideal de los métodos y técnicas de diseño y evaluación adquiere todo su significado en el marco de un enfoque más amplio, integrador, como el que propone el diseño centrado en el usuario”*. Por lo tanto la Usabilidad es un componente más dentro de las actividades de diseño y desarrollo de sistemas de información basados en web, y no el elemento central.

Para Garret (2002) el desarrollo de un sitio web es un proceso iterativo, que va desde la conceptualización (asociada a procesos abstractos) hacia la culminación en un producto concreto. Esto es compatible con la visión de Rosenfeld y Morville (2002), donde la Interfaz es lo concreto, lo visible, lo cercano al usuario, mientras que prácticamente la totalidad del proceso de conceptualización, diseño y desarrollo utilizan técnicas que en mayor o menor medida son transparentes al usuario. Por ello planteamos el siguiente esquema, en donde se aúnan ambas visiones y se complementan con la ubicación de los conceptos de Accesibilidad, Usabilidad y

Arquitectura de la Información.

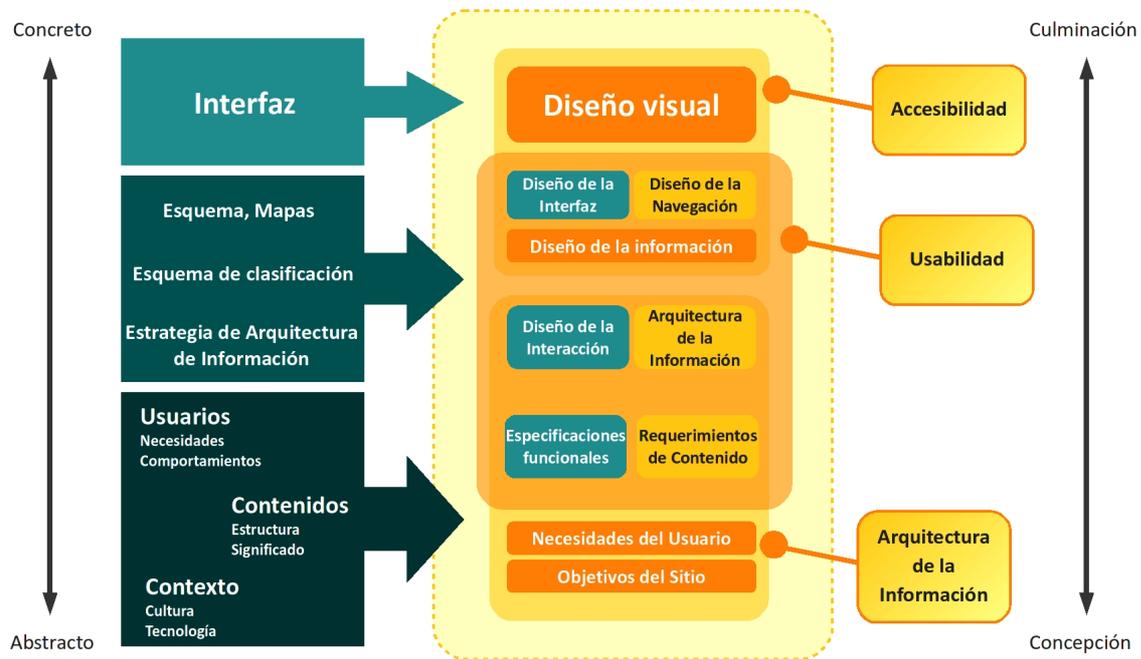


Ilustración 32: Integración de la Accesibilidad, Usabilidad y Arquitectura de la Información en el desarrollo de un sitio web

En la columna de la izquierda se representa la visión de Rosenfeld y Morville, en la del centro la de Garret y a la derecha los momentos en los que intervienen las diferentes técnicas de Accesibilidad, Usabilidad y Arquitectura de la Información. Se observa como no hay un límite claro entre Accesibilidad y Usabilidad durante los procesos de desarrollo de un sitio web, como tampoco lo hay entre Usabilidad y Accesibilidad durante la creación de la interfaz y el sistema de navegación. El proceso de planificación y diseño mediante la Arquitectura de la Información también es permeable a criterios de usabilidad. Hay que contemplar el proyecto de creación de un sitio o aplicación web como algo abierto, iterativo, en constante evolución (debido a los cambios que se producen en el entorno) y por tanto objeto de un constante rediseño y mantenimiento a través de mecanismos de evaluación y retroalimentación entre usuarios y diseñadores. Creemos que no merece la pena discutir acerca de si la Usabilidad está dentro de la Accesibilidad, o a la inversa, o si ambas se incluyen dentro de la Arquitectura de la Información. Cada una de estas disciplinas tiene su momento de aplicación, dentro de una constante sinergia a través de procesos y herramientas que ayudan a la evaluación y mejora de los resultados.

4.2. Análisis funcional de una aplicación web para la gestión y uso de tesauros.

En el bloque anterior hemos analizado el entorno tecnológico y metodológico en el que se crea y evoluciona cualquier aplicación web. El conocimiento de las distintas posibilidades de desarrollo, tanto en el lado del cliente como del servidor, proporciona una visión general de las posibilidades de diseño y un breve estudio del concepto de servicio web permite analizar la potencia del intercambio de datos entre aplicaciones a través de XML. Disponer de una perspectiva integrada de las pautas de accesibilidad, de las recomendaciones de usabilidad y de diseño según criterios de Arquitectura de la Información permite esbozar las principales características de la interfaz de usuario, así como la organización de las diversas funciones y contenidos de un sitio o aplicación web.

Es preciso delimitar las características funcionales específicas que ha de ofrecer una aplicación web para la gestión y uso de tesauros basados en SKOS. Nuestra herramienta tiene como propósito la construcción de tesauros por parte de una comunidad de usuarios que se comunican y trabajan a través de Internet, principalmente con el Correo Electrónico y la Web. Por tanto, debe incorporar ciertas funciones de grupos de trabajo y flujo de trabajo.

Éste es un rasgo incorporado por Pastor y Martínez (2003) y que aplicaremos de forma extendida a nuestra propuesta. Obviamente, también hay que contemplar las operaciones propias que realiza una aplicación de este tipo (la creación y mantenimiento de tesauros). Para ello se parte del trabajo de Moya y Gil (2001), recogiendo las principales conclusiones del mismo y añadiendo nuevas funcionalidades que nos sugiere SKOS para obtener una aplicación completa lo más fiel posible a este modelo.

Igualmente consideramos imprescindible ofrecer una herramienta que no se limite a las tareas propias de creación y mantenimiento de un tesoro. El modelo de aplicación que también asumirá las funciones de descripción de recursos, agrupados en repositorios y su indización mediante los conceptos de los tesauros gestionados. Por su parte, la descripción de los recursos se plantea a través de la aplicación del vocabulario de Dublin Core. El proceso de indización que planteamos se realiza de forma manual o asistido, a partir de los metadatos descriptivos del recurso.

Se profundizará en ofrecer un modelo de interfaz de aplicación. Esto resulta de especial relevancia al ser necesario estructurar las funciones que describiremos a continuación, en una interfaz integrada. Abordaremos los aspectos relacionados con la interacción del usuario con tesauros, repositorios, interrelacionando los procesos de consulta, mantenimiento e indización y dando cabida a las funciones de trabajo en grupo y flujo de trabajo. La finalidad de este análisis es desarrollar un esquema organizativo que permita una comprensión sencilla de la dinámica del sistema para un posible desarrollo de una aplicación, siempre a partir de los procesos de navegación, búsqueda y consulta.

Otro de los aspectos que caracterizarán nuestro modelo será la integración con aplicaciones externas a través de servicios web. De este modo, aplicaciones especializadas en la gestión de contenidos o en el mantenimiento de repositorios de metadatos, podrán hacer uso de estos servicios web para ampliar sus funcionalidades, a través de la incorporación de un tesauro como herramienta para la indización de recursos. Asimismo, estos servicios web pueden utilizarse para el desarrollo de marcos de explotación tanto de los tesauros, como de los repositorios, mantenidos por una aplicación basada en el modelo propuesto.

4.2.1. Indicaciones sobre la aplicación de SKOS en el diseño funcional

Una de las características de SKOS es la posibilidad de que un mismo concepto pueda pertenecer a uno o varios esquemas conceptuales, e incluso no pertenecer a ninguno. Esto puede suponer algunos problemas en la gestión de tesauros cuando se realiza de un modo compartido por varios usuarios. Para modelar nuestra propuesta es necesario delimitar o establecer ciertas restricciones en este aspecto con el fin de asegurar la integridad de los datos del sistema.

Sin embargo ello no debe implicar un aumento en la complejidad del modelo o un incremento en el número de operaciones que ha de realizar el usuario. A estos efectos definiremos algunas consideraciones orientadas a definir un modelo más preciso y que facilite el desarrollo futuro de la aplicación, resolviendo algunas cuestiones que puede suponer la implementación de esta característica.

Por tanto se establecen los siguientes considerandos:

- Cada concepto está asociado a un único tesauro. Por tanto la creación de un concepto conllevaría el establecimiento de la propiedad `skos:inScheme` entre el concepto creado y el tesauro que se encuentre gestionando el usuario en ese momento. De esta forma se conservará la información acerca del tesauro en el que se ha creado el concepto, y se evita la gestión manual de esta propiedad, impidiendo su borrado.
- Para que un concepto pueda ser compartido en la construcción de otros tesauros deberá indicarse de forma explícita. Por defecto todo concepto nuevo se creará como de uso exclusivo dentro del tesauro gestionado a menos que el usuario indique lo contrario. Un concepto compartido únicamente podrá ser utilizado en un tesauro diferente desde el que se creó para el establecimiento de relaciones de correspondencia.
- Se prescinde de la propiedad `skos:hasTopConcept`, ya que es implícita a la propia estructura jerárquica del tesauro y a nuestro parecer resulta redundante una definición manual de la misma, con la posibilidad de producirse incoherencias. Los conceptos cabecera de un esquema serán aquellos que tengan definida una propiedad `skos:inScheme` con el esquema en cuestión, ubicado en una estructura jerárquica y que no tenga ninguna relación `skos:broader` o `skos:broaderTransitive` asociada. De esta forma, la identificación de los conceptos cabecera será inferida por la propia aplicación.
- Un concepto no podrá tener asociadas dos etiquetas del mismo idioma con idéntica cadena literal con independencia de su carácter preferente o alternativo.
- No se podrá establecer relaciones entre conceptos si alguno de ellos no tiene asociada una etiqueta preferente en cualquier idioma. Asimismo, no podrá eliminarse una etiqueta preferente si al hacerlo se deja a un concepto que intervenga en algún tipo de relación sin ninguna etiqueta de este tipo.

Otro aspecto a considerar para permitir la coherencia de los procesos de gestión se refiere a la pertenencia explícita de las colecciones de conceptos a un único tesauro. Aunque en el modelo propuesto por SKOS no se define ningún tipo de restricción o

indicación acerca de la posibilidad de asociar el elemento `skos:inScheme` a una colección. En la práctica el modelo estaría definido para realizar una función similar.

Por supuesto durante la creación de relaciones la aplicación realizará las acciones y comprobaciones necesarias para asegurar el cumplimiento de las restricciones semánticas indicadas en el apartado 3.5.6.

4.2.2. Funciones de trabajo en grupo y flujo de trabajo

La coordinación de diferentes personas organizadas en grupos de trabajo para la colaboración en el desarrollo de una actividad concreta (en nuestro caso la creación y mantenimiento de un tesoro) precisa de elementos de comunicación. Estas herramientas, como el Correo Electrónico, pueden proporcionar la infraestructura necesaria, para los flujos de intercambio de información entre los miembros del equipo de trabajo.

Tal como indica González Lorca (2006), la Coordinación, Colaboración y Comunicación, constituyen el núcleo organizativo de los procesos de trabajo en grupo y flujo de trabajo. De este modo, estas propiedades deben ser contemplarse y ser incorporadas en nuestro modelo.

Se sugieren una serie de roles, o grupos de permisos, asociados a las diversas operaciones que podrán realizar los usuarios sobre los tesauros o los repositorios de recursos de información. La estructura de permisos y roles se incluye en la siguiente tabla.

↓ Privilegios	Rol →	Administrador	Gerente	Editor	Autor	Lector	Usuario anónimo
Configuración general de la aplicación		SI	NO	NO	NO	NO	NO
Gestión de usuarios y grupos de trabajo		SI	SI ⁷⁸	NO	NO	NO	NO
Gestión de tesauros/repositorios		SI	NO	NO	NO	NO	NO
Configuración general de tesauros/repositorios		SI	SI	NO	NO	NO	NO
Propuesta de elementos del tesoro/repositorio		SI	SI	SI	SI	NO	NO
Revisión de elementos del tesoro/repositorio		SI	SI	SI	NO	NO	NO
Gestión de elementos del tesoro/repositorio		SI	SI	SI	NO	NO	NO
Consulta del tesoro/repositorio		SI	SI	SI	SI	SI	NO
Consulta tesauros/repositorios públicos		SI	SI	SI	SI	SI	SI

Tabla 15: Roles con sus respectivos permisos

Describamos a continuación las funciones asociadas a cada privilegio con la finalidad de conocer el contexto al tratar aspectos relacionados con uno u otro rol:

- **Configuración general de la aplicación:** Esta tarea, realizada por el administrador, se asocia a aspectos tales como indicación de la base de datos utilizada en el servidor, incorporación de vocabularios generales (disponibles para todos los tesauros y repositorios) de metadatos, plantillas de visualización generales, indicación de la URI base del sistema, gestión de módulos de ampliación o tipo de registro de usuarios en el sistema.
- **Gestión de usuarios y grupos de trabajo:** El administrador tendrá un control total sobre todos los usuarios, grupos de trabajo y asignación de privilegios. Dependiendo del tipo de registro de usuarios, configurado a nivel general de

⁷⁸ Se trata de un permiso parcial, restringido a los tesauros y recursos que tenga asignados como gerente. El control total de usuarios, grupos y asignación de roles es función del administrador.

aplicación, los usuarios serán registrados por el administrador o podrán registrarse directamente ellos mismos (con validación posterior por parte del administrador). Los roles se asignan a grupos de trabajo compuestos por usuarios. Un usuario podrá estar en más de un grupo de trabajo. Resulta indispensable especificar el grupo, tipo de rol y tesoro o repositorio al que se refieren los privilegios durante el proceso de asignación de los mismos. Los gerentes solo podrán asignar privilegios referidos únicamente a los tesauros o repositorios que supervisen.

- **Configuración general de tesauros/repositorios:** En esta función se indica si un tesoro o repositorio tiene carácter público (consultado por cualquier usuario, incluso aquellos ajenos a la aplicación), plantilla de visualización, posibles módulos de ampliación activados, política de flujo de trabajo y de revisiones o si se envían notificaciones de correo electrónico para informar de propuestas y revisiones de elementos. Desde esta función se puede establecer el idioma principal del tesoro o repositorio, es decir, aquel que por defecto se asigna a las etiquetas asociadas a conceptos o colecciones.
- **Propuesta de elementos del tesoro/repositorio:** Esta tarea es realizada normalmente por los autores, aunque también entra dentro de los privilegios de editores y gerentes. Realizar una propuesta de un término, etiqueta o relación, ya sea semántica o léxica, no implica su incorporación inmediata al tesoro. Previamente ha de ser revisada por el editor. La propuesta de elementos que puedan producir incoherencias en el tesoro, tales como infracciones de la consistencia semántica o etiquetas preferentes duplicadas, podrán realizarse previa advertencia de ello al usuario.
- **Revisión de elementos del tesoro/repositorio:** Los grupos asignados al rol de editor son quienes realizarán, principalmente, esta función. Las propuestas son revisadas por el editor que puede validarlas, anularlas o dejarlas pendientes de validación. Es posible que un autor haya propuesto elementos que infrinjan la coherencia del tesoro, en estos casos el editor que efectúe la revisión deberá solventar las incoherencias antes de validar las propuestas.
- **Gestión de elementos del tesoro/repositorio:** Se trata de otra tarea propia de los editores y en ocasiones de los gerentes. Los usuarios con este privilegio podrán crear directamente los diferentes elementos del tesoro o repositorio

sin necesidad de validación. Obviamente, los elementos deberán respetar la coherencia del tesoro. Se podrán importar tesauros en su totalidad o parcialmente a partir de URIs o ficheros de datos en los que se encuentren tesauros representados en SKOS o en los formatos propios de CERES o LIMBERT

- **Consulta del tesoro/repositorio:** El rol de lector solo tiene disponible los permisos requeridos para consultar los tesauros o repositorios. No hay que confundir este rol con los usuarios anónimos que pueden consultar los recursos de carácter público. Los usuarios que tengan asociado el rol de lector podrán consultar los elementos de tesauros o repositorios, que hayan sido revisados (o creados en su caso directamente por los editores), con independencia de que esté permitida o no su consulta pública.
- **Consulta de tesauros/repositorios públicos:** Resulta razonable que durante el proceso de elaboración de un tesoro o la carga inicial de un repositorio, estos recursos no estén disponibles para su consulta. Los gerentes decidirán el momento de hacer público un recurso, para lo que realizarán los cambios necesarios en la configuración general del tesoro o repositorio en cuestión.

Los usuarios podrían configurar cierta información personal de su cuenta de usuario, como clave de acceso, fotografía, datos personales, datos de contacto o correo electrónico entre otros datos.

La dinámica de flujo de trabajo no reviste complejidad. El administrador o el gerente pueden crear los grupos de trabajo necesarios asignándoles los roles correspondientes. Los usuarios se asocian a uno o varios grupos. Los autores realizan las propuestas, y de forma opcional, el sistema puede enviar un mensaje de correo electrónico a los editores informando de tal hecho. Los editores revisan las propuestas que pueden validar o rechazar. El resultado de la revisión puede comunicarse a los autores, también de forma opcional, por correo electrónico.

El sistema contemplará diversas opciones para configurar la política de flujo de trabajo para la distribución de las tareas de revisión. El gerente decidirá si las propuestas se quedan en una lista compartida por todos los editores para su posterior auto-asignación, si se realiza un reparto automático según la carga de trabajo de los editores o si se realiza el proceso de forma ponderada por varios editores. Para esta última opción, el gerente o gerentes fijan un número de editores que deben validar una

propuesta para que sea aceptada, al que denominaremos umbral de validación. El último editor con el que se supere este umbral deberá solventar las posibles incoherencias de las propuestas. El gerente podrá asignar manualmente las propuestas. Igualmente, los editores podrán redirigir tareas de revisión a otro editor de su grupo o a otro grupo de editores del tesoro o repositorio.

En la configuración también se podrá asignar a un grupo de autores, un grupo específico de editores para la revisión. De esta forma se podrán crear áreas temáticas que mejoren la eficacia y especialización de las tareas de creación y mantenimiento de un tesoro o repositorio.

Los editores podrán acceder a un listado completo de las propuestas asignadas pendientes de revisión o bien a una relación completa de propuestas no asignadas para la auto-asignación por su parte. El sistema guardará una relación completa de las propuestas y del flujo que han seguido.

Una alternativa que en un principio resultaría interesante es la posibilidad de crear jerarquías de grupos de trabajo, aunque pensamos que podría añadir una complejidad excesiva en el sistema de permisos, por lo que en nuestra propuesta no se contempla dicha posibilidad.

4.2.3. Funciones de construcción y mantenimiento de tesauros⁷⁹

El conjunto de funciones que, obviamente, debe realizar esta aplicación está formado por la creación, la configuración y la eliminación de esquemas conceptuales (Tesauros). Esto incluye la creación de conceptos, etiquetas y relaciones. Igualmente, los procesos secundarios, fruto de la creación, mantenimiento y borrado de conceptos, relaciones semánticas y etiquetas deberán efectuarse automáticamente por la aplicación, lo que garantizará la integridad del tesoro, al evitar que el usuario deba realizar estas labores manualmente. Por ejemplo, si un usuario desea borrar un concepto, el sistema deberá advertirle que también se eliminarán las relaciones semánticas y etiquetas que pudiera tener asociadas, y tras confirmarse el borrado, la aplicación realizará esta tarea por sí misma.

Durante el proceso de navegación por la estructura del tesoro se podrán añadir o

⁷⁹ La medida en la que un usuario podrá construir y mantener el tesoro vendrá determinada por el perfil de gestión asignado. Este aspecto se aborda en el apartado 4.2.1

eliminar conceptos en ciertos conjuntos de conceptos, a los que denominaremos grupos de marcadores. En realidad, son listas de conceptos en las que el usuario podrá añadir o quitar referencias a conceptos al tiempo que realiza los procesos de navegación, consulta y mantenimiento del tesauro. La finalidad de estas listas es la realización de diversas operaciones, en las que se encuentren implicados varios conceptos simultáneamente. Identificamos los siguientes grupos de marcadores, cada uno de los cuales se orienta a una funcionalidad concreta:

- **Mantenimiento del tesauro:** Permite crear varias relaciones de un mismo tiempo entre un concepto y aquellos que se han almacenado en este grupo de marcadores. También se puede aplicar para añadir elementos a una colección de conceptos.
- **Proceso de indización:** Los conceptos de este grupo se asignan a uno o varios recursos de un repositorio.
- **Definición de búsquedas:** Podrán utilizarse los conceptos de este grupo para definir o refinar búsquedas sobre los repositorios.

Estos grupos de marcadores serán individuales para cada usuario y se mantendrán entre sesiones. Podrán incluir conceptos de diferentes tesauros, aunque sólo los conceptos del grupo de marcadores pertenecientes al tesauro actual podrán ser empleados para la creación de relaciones semánticas. Aquellos conceptos no pertenecientes a otro tesauro podrán ser utilizados para las relaciones de correspondencia.

Todo concepto tendrá asociado un nombre legible por máquina, suministrado por el usuario en el momento de su creación. Si no se facilita dicho nombre, el sistema le asignará automáticamente un identificador numérico. En la creación del concepto se especificará si el mismo está restringido al tesauro gestionado o tendrá carácter compartido. Podrán asociarse tantas etiquetas léxicas como se desee, aunque la aplicación asegurará que para un mismo idioma no se asocie más de una etiqueta preferente ni se permitan dos etiquetas con la misma morfología. También podrán asociarse varias firmas (notaciones) a cada concepto.

Las relaciones semánticas y de correspondencia se crearán a partir del concepto visualizado por el usuario. El concepto destino de la relación se seleccionará utilizando

un listado completo de conceptos pertenecientes a cualquier tesoro del sistema a los que el usuario tenga acceso. Para obtener una rápida localización se ofrecerá una herramienta para realizar filtros a partir de cadenas de texto existentes en las etiquetas. El filtro se aplicará automáticamente entre los conceptos del mismo tesoro para las relaciones semánticas, o sobre los conceptos compartidos de distintos tesoros para las relaciones de correspondencia. En cualquier caso el usuario solo tendrá acceso a aquellos conceptos para los que tenga permisos al menos de lectura. Para ello los conceptos deberán haber sido validados y pertenecer a un tesoro público o que el usuario tenga al menos permisos de lector para el mismo.

También será posible utilizar los conceptos almacenados en el grupo de marcadores de mantenimiento, pudiendo crear en una sola operación varias relaciones del mismo tipo. Las relaciones léxicas se establecerán seleccionando el tipo de relación y dos etiquetas de un concepto. Otra tarea que debe realizar la aplicación es el establecimiento automático de las relaciones inversas (entre conceptos o unidades léxicas), así como el control de la integridad de las mismas, tal y como indicamos en el capítulo anterior.

La tipología de relaciones semánticas, léxicas y de correspondencia debe estar abierta a posibles ampliaciones y configuraciones personalizadas. En el caso de las relaciones semánticas y de correspondencia, deberán especificarse los siguientes datos:

- Identificador de la relación semántica
- Si se trata de una relación con propiedades de transitividad.
- Si se trata de una relación simétrica y en caso contrario cual es la relación inversa.
- Si deriva de alguna relación básica ya existente (genérico, específico, relacionado o alguna de las relaciones de correspondencia)
- El ámbito de aplicación de la relación: dentro del mismo tesoro (relación semántica) entre conceptos de tesoros diferentes (relación de correspondencia).

El proceso de ampliación es similar para las relaciones léxicas. El sistema realizará las operaciones necesarias para mantener la integridad del tesoro cuando se elimina

alguna de estas definiciones. Las relaciones semánticas y de correspondencia básicas no podrán ser modificadas por el usuario. Únicamente los usuarios con permisos de administrados o gerente podrán realizar estas operaciones.

Otro aspecto a considerar es el tratamiento distinto que da SKOS a las unidades léxicas en el caso de que se establezcan relaciones entre las mismas o no. Este hecho ha de ser totalmente transparente al usuario sin que sea consciente de que una unidad léxica es tratada como un recurso o como una propiedad. En el caso de que sea tratada como recurso se asignará como identificador del mismo el texto de la etiqueta previa transformación para su normalización a un formato legible por máquina.

La aplicación permitirá la creación de colecciones para agrupar conceptos. De forma opcional podrá definirse el orden de los conceptos dentro de las colecciones. Las colecciones podrán tener asociadas etiquetas siguiendo los mismos considerandos que los conceptos. La asignación de conceptos a colecciones puede hacerse de forma individual, concepto a concepto, o utilizando el grupo de marcadores de mantenimiento. De igual forma se tendrá en cuenta la creación de elementos de documentación de diversa índole para los conceptos, atendiendo a la tipología propuesta por SKOS: notas simples, notas de historial, notas de alcance, ejemplos, definiciones, notas de edición y notas de cambios.

Los usuarios con nivel de permisos de gerente o superior podrán realizar una incorporación automática de elementos a través de la importación, previa validación, de ficheros o URIs que utilicen SKOS para la codificación de tesauros. Será posible indicar la extensión del proceso de importación, restringiendo o ampliando el mismo a conceptos, relaciones, etiquetas, colecciones o metadatos.

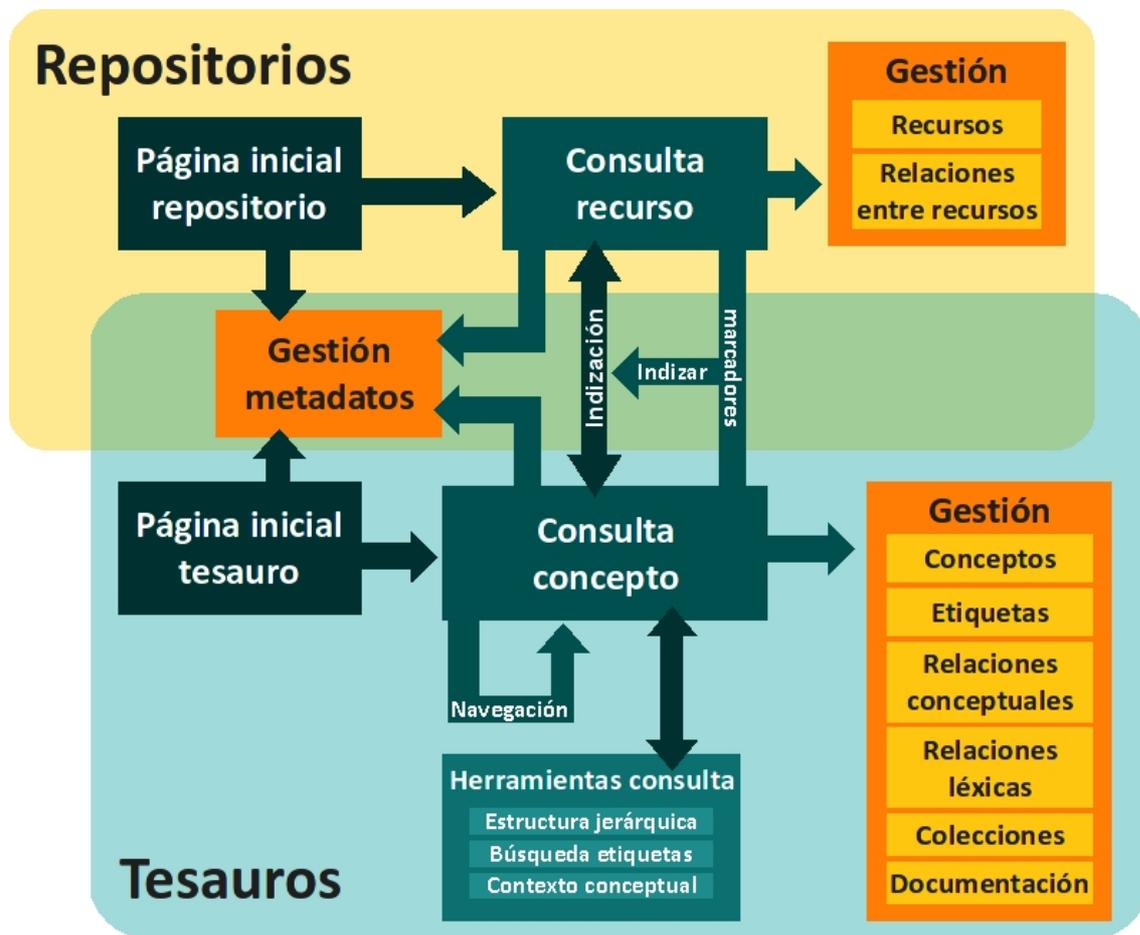


Ilustración 33: Dinámica integral de las funciones de consulta y mantenimiento

Parte de la potencia de SKOS tiene origen en su naturaleza como vocabulario RDF/XML, en el que juega un papel fundamental Dublin Core. Encontramos de gran utilidad la anotación mediante metadatos de esquemas conceptuales, conceptos y colecciones. De esta forma podrán anotarse los recursos de SKOS (esquemas, conceptos y colecciones) con metadatos Dublin Core. Ha de disponerse de un vocabulario general aplicable a todos los tesauros gestionados, que podrá ampliarse con un vocabulario específico para cada tesoro. Es posible recuperar en línea los elementos de cada vocabulario, siempre y cuando la especificación del mismo esté localizada en una URI asociada a un Namespace, pero por motivos de operatividad los vocabularios se cargarán previamente en el sistema y se almacenarán en la base de datos del sistema. Este proceso de carga de vocabularios debe poder hacerse automáticamente (a través del Namespace) o manualmente.

4.2.4. Funciones de gestión e indización de repositorios de recursos.

El modelo de aplicación debe abarcar tanto los procesos de gestión y mantenimiento de un tesauro como su aplicación en la indización de recursos agrupados en estructuras a las que denominaremos repositorios. Se tratan de colecciones de objetos identificados mediante URIs a los que se les asocian metadatos descriptivos y conceptos de uno o varios tesauros.

El esquema de metadatos a utilizar será, en un principio, Dublin Core, pudiéndose incorporar nuevos esquemas, dejando de esta manera abierta la ampliación del nivel descriptivo de la aplicación.

Cada recurso podrá caracterizarse con diversos elementos de Dublin Core, o en su caso de otro vocabulario, respetando siempre las restricciones de integridad que contemple cada modelo. Los recursos también estarán sujetos al mismo flujo de trabajo descrito con anterioridad. Podrán incorporarse descripciones de recursos mediante URIs externas para la importación de metadatos. Una alternativa a la importación sería la vinculación sincronizada a partir de la URI original.

De manera regular, la descripción de los recursos incorporados mediante este método, serían actualizados por la aplicación. Se garantiza, en cierta medida, que el registro estará actualizado por parte de la institución que realizó la descripción. Además, este tipo de recursos participarán en los procesos de búsqueda a partir del contenido descriptivo. También será posible establecer hipervínculos, etiquetados de forma libre, entre recursos.

La indización de los recursos de los repositorios es otra función que contempla nuestra propuesta. En el proceso de creación de un registro correspondiente a un recurso o en su posterior modificación se permitirá la asignación de conceptos de uno o varios tesauros.

Para ello se hará uso del elemento *Subject*⁸⁰ de Dublin Core, para la asignación de conceptos del tesauro a los documentos de los repositorios. Dicha asignación se realizará manualmente, seleccionando un concepto de una vista de navegación del tesauro. También podrá asignarse el contenido del grupo de marcadores de indización

⁸⁰ El elemento *Subject* permite asociar un recurso descrito mediante Dublin Core a otro recurso que representa una materia o, en nuestro caso, un concepto perteneciente a un esquema conceptual descrito mediante SKOS.

para la asignación, en una sola operación, de varios descriptores a un recurso.

```
1. <rdf:Description rdf:about="http://www.w3c.rl.ac.uk/examples/web/page_java.html">
2.   <dc:subject rdf:resource="http://www.w3c.rl.ac.uk/examples/thesaurus#002"/>
3. </rdf:Description>
```

Ejemplo 17: Indización conceptual de un recursos mediante el elemento dc:subject

Este proceso también puede ser contemplado desde el punto de vista del tesoro. Es decir, durante la consulta o mantenimiento del repositorio o repositorios el usuario añade recursos en un grupo de marcadores específico para estos objetos. Posteriormente, durante la consulta del tesoro puede indizar los elementos del grupo de marcadores de recursos con el concepto visualizado, o bien asociar dicho grupo de marcadores con el de conceptos.

Opcionalmente el usuario podrá recurrir a una indización asistida, donde la aplicación, a través de diversas técnicas que no son objeto de este trabajo, propondrá al usuario una lista de conceptos para indizar un determinado recurso. Esta lista podrá ser alterada por el usuario.

Para la asignación de conceptos a varios recursos el sistema ofrecerá un grupo de marcadores, similar a los marcadores de conceptos disponibles desde la consulta del tesoro. El usuario podrá agregar a este grupo aquellos recursos que desee en cualquier momento de la consulta de los mismos.

4.2.5. Servicios web de consulta

Los servicios web que plantea este modelo están orientados a la explotación de los contenidos públicos de tesauros y repositorios. Por lo tanto, no se incluye ningún tipo de operación de mantenimiento.

La recuperación de un concepto implica también la recuperación de las etiquetas asociados al mismo. De igual forma, la recuperación de una relación de cualquier tipo supondrá la recuperación de las etiquetas de los conceptos que conforman la relación.

- Directorio de recursos públicos. Este servicio permite conocer aquellos tesauros y repositorios cuya consulta está permitida a cualquier usuario. No es necesario

especificar ningún parámetro de entrada y el servicio devolverá un listado que especificará el identificador del recurso y si se trata de un tesoro o un repositorio.

- Recuperar metadatos y conceptos cabecera de un tesoro. Recupera todas las anotaciones de metadatos y conceptos cabecera de un tesoro público. Ha de especificarse un identificador de tesoro válido.
- Recuperar colecciones de un tesoro. Devuelve todos los identificadores de colecciones de un tesoro público, junto con la etiqueta y metadatos correspondientes. Ha de especificarse un identificador de tesoro válido.
- Recuperar componentes de una colección. Con este servicio se obtienen las etiquetas y metadatos de una colección, así como las referencias de todos los conceptos (junto con sus etiquetas) pertenecientes a la misma. Ha de especificarse un identificador de colección válido.
- Recuperar concepto. Este servicio posibilita el acceso a la información de toda la información relativa a un concepto determinado: etiquetas, metadatos, relaciones semánticas y léxicas, pertenencia a colecciones y esquemas de conceptos. Ha de especificarse un identificador de concepto válido.
- Buscar en etiquetas. Recupera aquellos conceptos, junto con sus etiquetas y metadatos, que contenga la cadena especificada en alguna de sus etiquetas preferentes o alternativas. Ha de especificarse un identificador de tesoro válido y una cadena de búsqueda.
- Búsqueda ampliada por relaciones semánticas. Devuelve, junto con las etiquetas y metadatos de un concepto, idéntica información de aquellos conceptos que son jerárquicamente inferiores y/o relacionados con el concepto indicado. Ha de especificarse un identificador de concepto válido y el tipo de búsqueda ampliada (jerárquica, relacionada o la combinación de ambas).
- Lista de directorios y elementos de metadatos disponibles para operaciones de navegación y búsqueda. Proporciona una lista de aquellos elementos de metadatos configurados para su consulta a través de una navegación alfabética o para su uso en los procesos de búsqueda. Ha de especificarse un identificador

de repositorio válido.

- Listado de recursos ordenados por un campo de metadatos. Recupera una lista de recursos ordenados por un elemento de metadatos configurado para la navegación. Es necesario especificar un identificador de repositorio válido, un campo de metadatos correcto, el número de orden del primer recurso del listado y el número de recursos a listar.
- Búsqueda de recursos por metadatos. Obtiene una lista de recursos coincidente con los criterios de búsqueda por elementos de metadatos. Es necesario especificar un identificador de repositorio y los elementos de metadatos sobre los que se realiza la búsqueda han de estar configurado para ello.
- Metadatos descriptivos de un recurso. Devuelve todos los metadatos descriptivos de un determinado recurso incluyendo los hipervínculos etiquetados y la lista de identificadores de aquellos conceptos que asignados en la indización del recurso. Ha de especificarse un identificador de recurso válido.
- Indización cotemática en un repositorio. Recupera una lista de identificadores de todos los recursos de un repositorio, indizados por un concepto concreto, o por alguno de los utilizados para indizar un recurso. Ha de especificarse un identificador de repositorio, así como un identificador de concepto o de recurso válidos.
- Indización específica en un repositorio. Similar al servicio web de indización cotemática ampliando la cobertura conceptual a los conceptos específicos. Ha de especificarse un identificador de repositorio, así como un identificador de concepto o de recurso válidos.
- Indización relacionada en un repositorio. Similar al servicio web de indización cotemática ampliando la cobertura conceptual a los conceptos relacionados. Ha de especificarse un identificador de repositorio, así como un identificador de concepto o de recurso válidos.

Todos los servicios orientados a la recuperación de recursos de repositorios ofrecerán de forma opcional el filtrado de resultados. Esto se llevará a cabo especificando como parámetros el contenido buscado en determinados campos de metadatos.

Por defecto, los servicios ofrecerán los resultados en RDF utilizando cuando sea necesario el vocabulario correspondiente de SKOS o Dublin Core. Para aumentar la interoperabilidad con otros sistemas también podría contemplarse la posibilidad de utilizar otras especificaciones (CERES, LIMBER, ILRT, GEM, etc) en la recuperación de información relativa a los tesauros, simplemente añadiendo un parámetro que especifique el formato solicitado.

4.3. La interfaz de usuario como núcleo funcional integrador

Una aplicación web implementa la interfaz de las distintas funcionalidades, aplicando un esquema de acceso basado en el hipertexto. Por tanto, los procesos de navegación son los que definen la forma en la que el usuario interacciona con contenidos informativos y funcionales (Lamarca Lapuente, 2006). La navegación se complementa con la búsqueda de diversos elementos, tanto del tesoro como del repositorio, a partir de la integración de ambos. Desde el punto de vista de la Arquitectura de la Información esta aproximación contempla la combinación de las dos vías utilizadas en los sistemas de hipertexto para el acceso a la de información (Pastor y Saorín, 1998; Rosenfeld y Morville, 2002).

Hay que considerar que en esta propuesta las funcionalidades de la aplicación web se organizan a lo largo de dos dimensiones. Por un lado, la que abarca el intervalo de aspectos que van desde las operaciones de consulta hasta las de mantenimiento y, por el otro, la que cubre las funciones relativas al tesoro o al repositorio de recursos. La interfaz de usuario ha de integrar ambas dimensiones a partir de los considerados de apuntados sobre Accesibilidad, Usabilidad y Arquitectura de la Información. Por tanto, esta integración de los procesos **consulta** ↔ **mantenimiento** ampliada con la de los objetos **tesauro** ↔ **repositorio** divide la estructura funcional de la aplicación y por tanto la organización de la interfaz de usuario en cuatro secciones:

- Consulta de tesauros.
- Mantenimiento de tesauros.
- Consulta de repositorios.
- Mantenimiento de repositorios.



Ilustración 34: Ejes funcionales del modelo con el proceso de Indización como elemento integrador

En la presente propuesta se contemplan las tareas de mantenimiento y consulta de un objeto relacionadas a través de una serie de procesos cognitivos de retroalimentación del propio usuario, que deben describirse y formalizarse para su modelado. Tanto los tesauros como los repositorios podrán ser construidos de forma simultánea a la consulta de los mismos por parte de un autor o editor.

Así pues, no hay que considerar las secciones anteriormente enumeradas como elementos funcionales aislados sino permeables, lo que facilita la necesaria integración de procesos. Precisamente será en dichas confluencias donde se realizarán los procesos de indización y se observarán los fenómenos sinérgicos en la construcción del tesoro, lo que nos conducirá desde una herramienta general de organización del conocimiento hacia otra, adaptable a un corpus textual determinado. Todo ello coordinado y supervisado a través de estructuras, elementos y procesos asociados a la dinámica de los sistemas de trabajo en grupo y flujo de trabajo.

De esta forma, la interfaz de la aplicación se erige como el núcleo integrador de las operaciones de consulta, mantenimiento e indización a partir de las funciones de navegación y búsqueda

4.3.1. Consulta de tesauros

Las operaciones de consulta de un tesauro comenzarán a partir de la visualización de los conceptos cabecera y metadatos del mismo. Mediante enlaces de hipertexto se accederá a la consulta del tesauro, centrándose, en un primer momento, en una vista en donde el concepto es la unidad básica de visualización de información. La visualización de un concepto incluye su identificador legible por máquina, etiquetas, relaciones, indicadores de existencia de elementos de documentación y metadatos del concepto y número total de documentos del sistema indizados por este concepto. Para simplificar la consulta de un concepto, la visualización de elementos de documentación, relaciones léxicas y metadatos debe ser activada por el usuario. También se incluirán las colecciones a las que está asociado el concepto con enlaces de hipertexto a la lista de conceptos de cada colección.

La navegación entre conceptos se realiza a través de enlaces de hipertexto con los que se representan las relaciones. Estos enlaces muestran el tipo de relación y la etiqueta preferente del concepto destino de la misma. Han de distinguirse las relaciones semánticas y las relaciones de correspondencia entre tesauros. Las relaciones léxicas no implicarán ningún tipo de navegación, visualizándose como un texto asociado de ambas etiquetas. Asimismo, será posible acceder a un concepto en particular a partir de la consulta de la lista de conceptos incluidos en una colección.

La consulta de un tesauro a partir de las etiquetas, elementos, relaciones y colecciones de un concepto se complementará con dos herramientas que aportarán visión de contexto. Por un lado se ofrece una panorámica completa de la estructura jerárquica del tesauro, indicando la ubicación del concepto consultado en la misma. La segunda herramienta sitúa el concepto en su entorno más inmediato: conceptos genéricos y conceptos específicos de los mismos. Esta presentación muestra un contexto más concreto, pero más útil en los procesos de consulta, mostrando el entorno conceptual en diferentes grados de proximidad y constituyendo un mecanismo de redefinición y expansión conceptual durante la búsqueda y la indización.

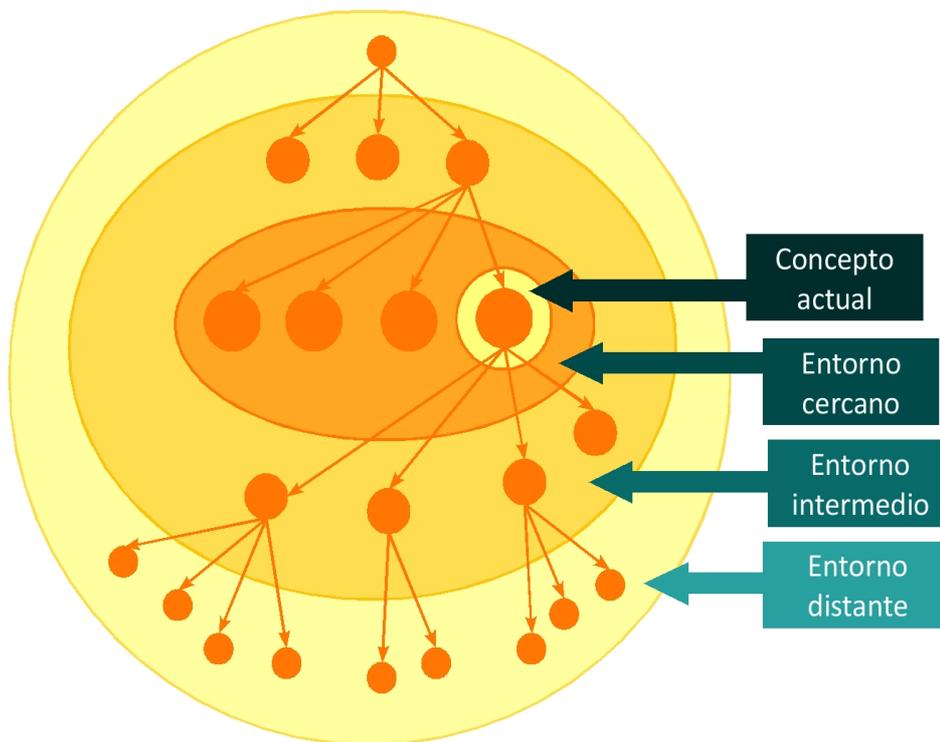


Ilustración 35: Niveles de proximidad en la vista de contexto de un concepto

Una alternativa interesante a esta dinámica de consulta la constituyen las colecciones de conceptos, desde el punto de vista que pueden constituir un punto de inicio en el proceso de navegación. Será necesario ofrecer un enlace desde la página principal del tesoro a la lista de colecciones del mismo desde donde se podría acceder a los conceptos incluidos la colección seleccionada por el usuario.

También encontramos útil la posibilidad de regresar al concepto cabecera de la jerarquía del concepto actual⁸¹. Además de la navegación, esta funcionalidad ofrecería cierta información de contexto durante el proceso de consulta.

Otra posibilidad, al inicio o durante el transcurso del proceso de navegación por el tesoro, es la de realizar una búsqueda en el texto de las etiquetas asignadas a los conceptos. El resultado se ofrecería como una lista, donde las etiquetas preferentes tomarían forma de enlaces que conducirían a su respectivo concepto, mientras que las

81 Por “*concepto actual*” se entiende aquel concepto, que en un momento determinado de cualquier proceso de consulta o mantenimiento, está siendo visualizado por el usuario como objeto de la operación desarrollada en un tesoro. Igualmente se entiende la expresión “*recurso actual*” para los contenidos de los repositorios.

etiquetas alternativas se mostrarían como un texto sin enlace y junto al mismo un enlace con el texto de la etiqueta preferente del concepto.

La interfaz dispondrá de un acceso directo a las colecciones de conceptos que se definan. Dentro de cada una de estas colecciones será posible consultar los conceptos que la componen. De igual manera, la aplicación mostrará las colecciones a las que pertenezca el concepto actual.

Por otro lado, los marcadores de conceptos mencionados anteriormente, se gestionarán a través de cuatro botones o enlaces. Tres de ellos añadirían el concepto actual a un grupo de marcadores concreto (mantenimiento, búsqueda e indización). El cuarto dará acceso a una página donde se visualizan los conceptos incluidos en cada grupo y desde la que se podrá eliminar dicha asignación.

Partiendo de la consulta de un tesauro se llegará a la consulta de los recursos de repositorios, siempre en virtud de la asignación de conceptos que se realice en los procesos de indización. La interfaz nos mostrará esta función desde la vista de concepto actual, en forma de enlace que nos conducirá a una página en donde se mostrarán una lista de objetos (agrupados por repositorios) indizados por dicho concepto, con enlaces a los registros completos de los recursos. En esta misma página se ofrecerá la posibilidad de especificar filtros según el contenido de ciertos elementos de metadatos.

La definición y ejecución de búsquedas complejas será otra de las características de la aplicación. No solamente podrá utilizarse un concepto concreto para recuperar los recursos indizados por el mismo, sino que además podrán combinarse varios conceptos utilizando el grupo de marcadores de búsqueda o añadiendo nuevos conceptos a búsquedas ya definidas, a partir de la consulta del concepto actual. La funcionalidad de búsqueda avanzada permitirá especificar varios parámetros

- Operador lógico para la combinación de los conceptos que definen la búsqueda: todos los conceptos (AND), alguno de los conceptos (OR), ninguno o exclusión de los conceptos (NOT) indicados.
- Repositorio de recursos sobre el que se realizará. Las búsquedas podrán construirse para involucren a todos los repositorios gestionados por la aplicación.

- **Ámbito de alcance conceptual de la búsqueda.**
- **Ampliación de la búsqueda a partir de relaciones de correspondencia**

El ámbito de alcance conceptual permitirá aprovechar las relaciones semánticas del tesoro para ampliar la extensión de la búsqueda mas allá de un determinado concepto. De esta manera distinguimos cuatro posibilidades:

- **Exacta:** Recupera aquellos recursos indizados por un determinado concepto. Es el tipo de búsqueda básico.
- **Jerárquica:** Recupera aquellos recursos indizados con un determinado concepto y con aquellos que le son jerárquicamente específicos.
- **Asociativa:** Devuelve como resultado los recursos indizados con un concepto concreto y aquellos otros con los que está relacionado.
- **Completa:** Combina las búsqueda jerárquica y asociativa.

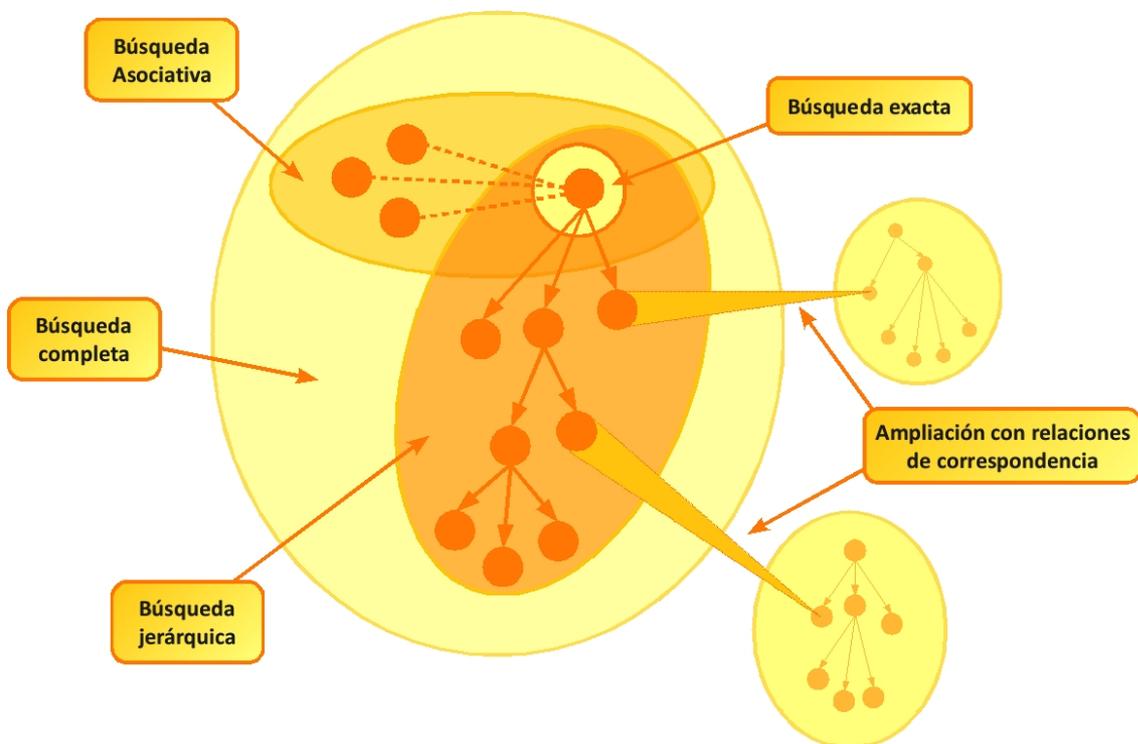


Ilustración 36: Tipos de búsqueda conceptuales en repositorios de recursos y su posible ampliación a partir de relaciones de correspondencia

La delimitación de las búsquedas también tendrá en cuenta su posible ampliación, siguiendo las relaciones de correspondencia que se establecen conceptos de diferentes tesauros.

Los resultados se mostrarán en forma de enlaces construidos a partir del título, o nombre legible por máquina en su defecto, de los recursos recuperados. Las entradas se agruparán por el título del repositorio, en caso de que se hayan obtenido entradas referentes a más de uno. Al seguir el enlace los usuarios accederán al registro completo de metadatos del recurso consultado.

4.3.2. Mantenimiento de tesauros

Los procesos de mantenimiento del tesoro se realizarán a partir de la vista de concepto actual. Recordemos que en esta presentación se mostrará toda la información referente a etiquetas, relaciones, elementos de documentación y metadatos, por lo que editores y autores tendrán una visión completa. Según la dinámica de consulta del tesoro, detallada en el apartado anterior, los usuarios podrán ir de un concepto a otro siguiendo las relaciones semánticas o realizando búsquedas de conceptos a partir de las etiquetas léxicas.

En cualquier momento de la consulta de un tesoro se podrán gestionar los conceptos que lo conforman a través de un enlace. Este enlace de gestión de conceptos permitirá la creación de nuevos conceptos, su eliminación y por supuesto su consulta. Para la creación de conceptos, la página incluirá un campo para la introducción del nombre legible por máquina, la indicación del carácter restringido (al propio tesoro gestionado) o compartido del concepto y un campo opcional para la creación de la etiqueta preferente en el idioma principal del tesoro.

Esta misma página también incluirá un listado completo paginado con los conceptos del tesoro ordenados alfabéticamente por la etiqueta preferente del idioma principal o en su defecto el nombre legible por máquina. Para facilitar la localización de conceptos el listado anterior incluirá una utilidad de filtrado de conceptos, indistintamente por etiqueta preferente o nombre legible por máquina.

La asignación de etiquetas se realizará desde la página de concepto actual. El usuario seleccionará de una lista el idioma de la etiqueta (por defecto aparecerá seleccionado el idioma principal), el tipo de etiqueta (preferente o alternativa) y el propio texto de la

etiqueta.

Un enlace o botón ubicado en la página de concepto actual (concepto origen de la relación), dará acceso a la página de gestión de relaciones. En esta página se seleccionará el tipo de relación y el concepto destino. La lista de términos sobre la que se podrá aplicar un filtro a partir de las etiquetas preferente o alternativas, variará dependiendo de la naturaleza de la relación. Si se trata de una relación semántica solo se visualizarán conceptos del mismo tesoro o aquellos otros que se hayan definido como compartidos aunque estén incluidos en otro esquema conceptual.

En el caso de relaciones de correspondencia se mostrará una lista adicional con los tesauros públicos o para los que el usuario tenga, como mínimo, privilegios de lector. Tras seleccionar un tesoro concreto se visualizará la lista de conceptos correspondiente.

También es posible utilizar el grupo de marcadores de mantenimiento para almacenar uno o varios conceptos como destino de una relación. En esta situación, la aplicación solo utilizará aquellos conceptos compatibles con la relación semántica o de correspondencia escogida.

A partir de un enlace situado en el propio texto de las etiquetas se accederá a la página de gestión de etiquetas. Desde ella podremos bien eliminar o modificar una etiqueta, bien crear relaciones léxicas, para lo que se escogerá el tipo de relación a crear y otra etiqueta (como destino de la relación) entre las que se encuentren asociadas al concepto actual.

Otro de los aspectos relevantes es la gestión de los diversos elementos de documentación y metadatos. Desde la vista de concepto actual se podrán añadir, eliminar o modificar los mismos, indicado el tipo de elemento y su contenido. Asimismo, podrán gestionarse los metadatos referentes al propio tesoro desde la página “raíz” del tesoro, donde se inicia la navegación a partir de enlaces a los conceptos cabecera. Desde esta misma página administradores y gerentes dispondrán de un acceso a las opciones de importación de elementos del tesoro.

La aplicación también permitirá, de forma parecida a la gestión de grupos de marcadores, incluir conceptos en las colecciones. Observamos que los marcadores de conceptos desempeñan un papel similar al de las colecciones. Mientras que éstas

forman parte de la propia definición de la estructura del tesauro, el ámbito de los marcadores de conceptos viene definido por la funcionalidad de la aplicación.

Por este motivo, pensamos en otra alternativa que permitiría simplificar la interfaz. El usuario seleccionaría de una única lista el marcador o colección al que quiere añadir el concepto actual (distinguiendo ambos tipos) y con un botón realizaría la asignación. Con un enlace se accedería a una página en la que se podría gestionar la asignación de conceptos a grupos de marcadores y colecciones, además de añadir o eliminar colecciones del tesauro.

El control dinámico de las restricciones semánticas durante la construcción del tesauro garantizará la integridad del tesauro. De esta forma la aplicación siempre avisará de la incorrección de aquellas relaciones que infrinjan alguna restricción y, por supuesto, no permitirá su establecimiento. Los mensajes de error informarán al usuario acerca de la restricción semántica que se ha intentado incumplir y lo que implicaría su violación en cuanto a la integridad del tesauro, así como recomendaciones o sugerencias para evitar este error.

4.3.3. Consulta de repositorios

Los registros de los recursos de repositorios podrán consultarse mediante búsqueda o navegación alfabética entre diversos elementos descriptivos de metadatos. Los usuarios con rol de gerente, establecerán en la configuración del repositorio, sobre qué elementos de Dublin Core podrán realizarse estas operaciones.

La búsqueda a través de un formulario mostrará una lista de resultados. La navegación alfabética por elementos de metadatos, como si de un directorio se tratara, mostrará como enlaces los valores de los mismos, de forma agregada e indicando el número de recursos que contienen dicho valor en el campo de metadatos.

En ambos casos el usuario accederá a una página de resultados con una lista de enlaces a partir del título de los recursos, hacia la ficha descriptiva de metadatos. Desde esta ficha, o desde las diferentes listas de navegación o de resultados, el usuario podrá incorporar recursos al grupo de marcadores correspondiente.

Durante la consulta del registro es posible lanzar búsqueda a partir del contenido de un elemento concreto de metadatos del recurso. La indización de conceptos a recursos

puede ser explotada durante la consulta del repositorio, conformando estructuras de navegación entre registros. Junto con los campos descriptivos del recurso se visualizarán, como enlaces, las etiquetas preferentes de los conceptos que lo indizan.

Esta presentación tiene tres propósitos: consultar los datos completos del registro, conformar un acceso rápido hacia otros recursos indizados por dichos conceptos y proveer un enlace hacia la respectiva entrada de los mismos en el tesauro, lo que llevaría a la vista de concepto actual. El acceso a otros recursos desde la indización puede enfocarse como una forma de encontrar registros similares al consultado. Esta similitud se enfocará a través de las siguientes alternativas:

- Recursos cotemáticos. Aquellos que están indizados por uno o varios de los conceptos que indizan al recurso.
- Recursos específicos. Aquellos que están indizados por algún concepto específico de uno o varios de los conceptos que indizan al recurso.
- Recursos relacionados. Aquellos que están indizados por algún concepto relacionado con uno o varios de los conceptos que indizan al recurso.

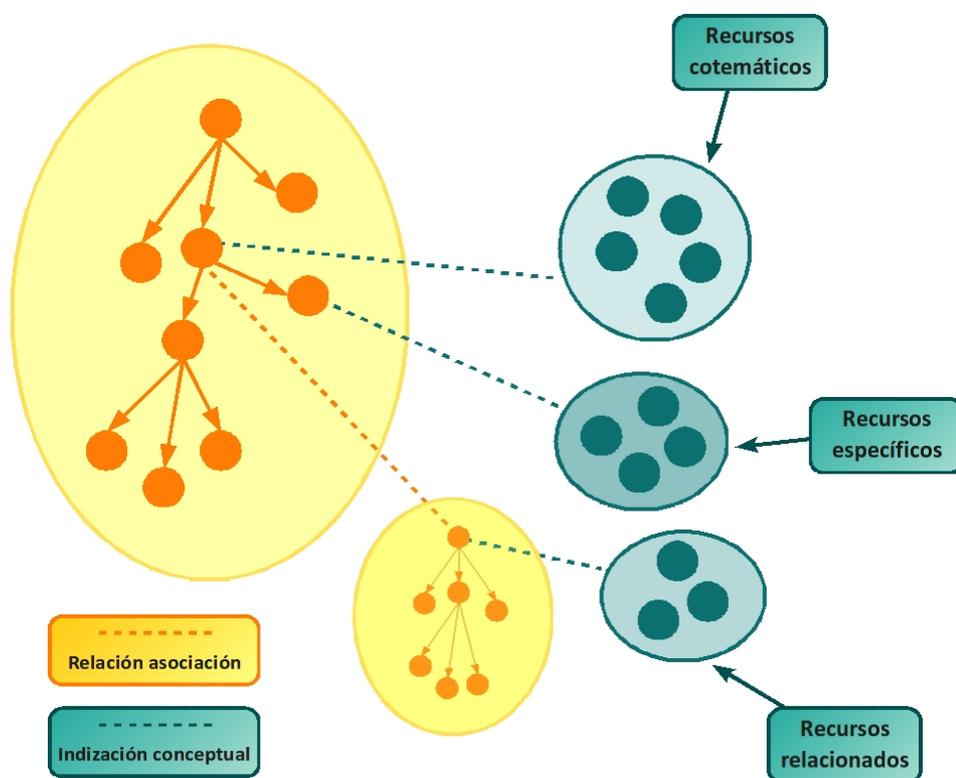


Ilustración 37: Similitud de recursos a partir de la indización conceptual

El alcance de esta búsqueda dependerá de la selección que haga el usuario de los conceptos que indizan al recurso consultado. Si el usuario selecciona uno o varios conceptos manualmente, de la lista de conceptos asignados a un recurso, la búsqueda se realizará de forma exacta. Cuando no se seleccione ninguno, se recuperarán los recursos indizados al menos por alguno de los conceptos asignados al registro visualizado, ordenándose por número de conceptos coincidentes en la indización.

4.3.4. Mantenimiento del repositorio

La introducción de nuevos elementos en el repositorio podrá hacerse de forma manual, importando ficheros, o mediante la técnica de vinculación sincronizada explicada en el apartado 4.2.3. Las dos últimas opciones requerirán una validación previa, con la finalidad de verificar que las fuentes estén codificadas en RDF.

Durante la consulta del repositorio, los usuarios con privilegios para ello, dispondrán un acceso para crear nuevos recursos. Para crear un nuevo registro será necesario especificar un nombre legible por máquina para identificar el recurso. Al igual que en el caso de los conceptos del tesauro, si no se proporciona dicho nombre, el sistema asignará automáticamente un identificador numérico al recurso. Adicionalmente, en el caso de los recursos vinculados, será necesario especificar la URI de sincronización.

Una vez creado el recurso, la aplicación ubicará al usuario en la página de la ficha descriptiva de registro de metadatos. Desde esta página será posible la consulta del registro completo, así como añadir, eliminar o modificar elementos descriptivos asociados al recurso. El vocabulario base a utilizar será Dublin Core, aunque también podrán incorporarse elementos de otras especificaciones, previamente introducidas en el sistema y habilitadas para su aplicación en el repositorio gestionado.

También será posible copiar recursos de un repositorio a otro desde la ficha descriptiva de cada registro o a partir del contenido del grupo de marcadores de recursos. El usuario deberá seleccionar de una lista de repositorios para los que tiene permiso de autor o editor, aquel en los que desea copiar los registros. La eliminación de recursos se realizará desde la página descriptiva del recurso. Esta operación implicará el borrado de todos los registros de indización en los que participe el recurso a eliminar.

En principio, las relaciones entre recursos se realizan a través del esquema conceptual entretrejado por el tesauro y la asignación de conceptos durante la indización. Este

aspecto ha sido tratado en apartado 4.2.5.3. al hablar de recursos cotemáticos, jerárquicos o relacionados. No obstante, de forma manual será posible establecer relaciones entre recursos. Dichas relaciones podrán estar etiquetadas para denotar la naturaleza de la relación. La tipología de relaciones será establecida por los gerentes del repositorio. Ambas soluciones se complementan mutuamente. La primera ofrece un mecanismo para el establecimiento de relaciones jerárquicas y asociativas, de manera implícita a través de las relaciones entre conceptos que indizan por los que se indizan los recursos. Además, este tipo de establecimiento manual de relaciones permite un mayor grado de flexibilidad en la definición de estructuras de interrelación de recursos que la que ofrece el elemento de metadatos *Relation*⁸² propuesto para su inclusión en Dublin Core.

4.3.5. Indización

En este apartado se describe la dinámica de asignación de conceptos a los recursos de información de los repositorios. El proceso de indización podrá realizarse de 3 formas diferentes:

- Durante la consulta de un recurso del repositorio, podrán indizarse el mismo con los conceptos del grupo de marcadores de indización.
- Durante la consulta de un concepto del tesoro, podrá el mismo a los recursos del grupo de marcadores de recursos.
- Durante la consulta del tesoro o del repositorio podrán asociarse todos los conceptos del grupo de marcadores de indización a todos los recursos del grupo de marcadores correspondiente.

Otro aspecto a destacar es la integración, en una sola operación, de la creación de recursos y su indización durante la consulta del tesoro. Esta funcionalidad permitirá, mientras el usuario consulta un concepto de un tesoro, crear el registro correspondiente a un recurso de información de un repositorio concreto, al tiempo que se indiza el nuevo recurso con el concepto actual.

⁸² El elemento *Relation* permite asociar un recurso descrito mediante Dublin Core a otro recurso. Sin embargo no permite incorporar ningún tipo de etiqueta para denotar la naturaleza de la relación.

La interfaz ofrecerá esta funcionalidad a través de un enlace o a través de un pequeño formulario integrado en la página de consulta del concepto. Sea cual sea la forma en la que se implemente esta característica el usuario solo tendrá que especificar el título del recurso y el repositorio al que deberá añadirse.

En el caso de que la indización se realice de forma asistida esta tarea se realizará desde la página de metadatos del recurso a indizar. Será necesario especificar el tesoro o tesauros que intervendrán en la indización del recurso. El resultado de esta operación se mostrará como una lista con las etiquetas preferentes de los conceptos seleccionados por el sistema y los agrupará según el tesoro al que pertenezcan.

El usuario seleccionará aquellos conceptos con los que desea indizar el recurso actual. Asimismo, durante la consulta del registro de metadatos de un recurso, se mostrarán los conceptos que indizan el mismo, siendo posible eliminar dicha asignación.

4.3.6. Flujo de trabajo

En apartados anteriores se ha hecho referencia a operaciones funcionales de mantenimiento de tesauros y repositorios tales como añadir conceptos, crear relaciones o eliminar metadatos entre otras. Hasta el momento no se ha hecho referencia a diferencia alguna en dichas operaciones dependiendo del rol asignado al usuario que las realice.

En la práctica cualquier tipo de operación de adición, modificación o borrado de elementos, tanto de tesauros como de repositorios, realizada por usuarios que operen bajo el rol de autor no repercutirá de forma inmediata en las acciones de consulta o mantenimiento del resto de usuarios.

Esto se debe a que los autores no realizan este tipo de cambios de forma inmediata, sino que en realidad generan una lista de propuestas que han de ser validadas por otros usuarios de perfil superior (editor, gerente o administrador).

Cuando un autor consulte un tesoro o repositorio los elementos que haya propuesto, y que todavía no se encuentren validados, podrán ser visualizados con una indicación que informe de tal hecho. Asimismo, el autor podrá consultar una página accesible desde la página “raíz” de los tesauros y repositorios, con un listado completo de sus propuestas pendientes de validación, así como información estadística completa

acerca de los elementos propuestos y validados por el usuario en cuestión. Desde esta misma página el autor podrá eliminar las propuestas que aún no hayan sido validadas.

El usuario no podrá utilizar elementos no validados propuestos por otros autores para realizar sus propuestas. Es decir, un autor podrá proponer un concepto pero hasta que no se valide no podrá proponer etiquetas asociadas al mismo o relaciones con otros conceptos válidos. De este modo, no podrá utilizar propuestas que hayan realizado otros usuarios hasta que se validen, momento en el que pasarán a formar parte del conjunto de elementos públicos disponibles tanto para los procesos de consulta como de mantenimiento. Únicamente Para agilizar el proceso de creación del tesoro se permitirá la propuesta simultánea de un concepto y una etiqueta preferente, cuya validación supondrá de forma automática la validación del concepto y viceversa.

El proceso de validación será realizado por editores, gerentes o administradores. Durante la consulta de tesoros y repositorios los editores visualizarán las propuestas que se les hayan asignado con un estilo visual diferente del utilizado para representar a los elementos ya revisados. También podrán acceder a una página de gestión de propuestas, para cada tesoro y repositorio, desde donde podrán validar, rechazar o reasignar las mismas.

Los gerentes y los administradores podrán visualizar todas las propuestas realizadas en cada tesoro y repositorio. Para estos usuarios la respectiva página de gestión de propuestas permitirá, además de las funciones propias de los editores, realizar la reasignación manual de propuestas a los propios editores para su revisión.

Los gerentes tendrán un acceso a una página de gestión de grupos de trabajo, desde donde podrán crear o borrar grupos de trabajo y asignar usuarios del sistema a los mismos. Además deberá asociar a cada grupo de trabajo un determinado rol (editor, autor o lector). En el proceso de creación de un nuevo tesoro o repositorio se creará asimismo un grupo que tendrá asociado el perfil de gerente. Este grupo no podrá ser eliminado por los gerentes pero podrán gestionar los usuarios asociados al grupo. Los administradores dispondrán de una página de control total sobre grupos de trabajo y usuarios del sistema.

A efectos de conservar un registro de las diferentes operaciones realizadas el sistema podría gestionar automáticamente un historial de los diferentes elementos de tesoros y repositorios. Esta información podría ser de utilidad para la elaboración de

estadísticas de uso del sistema o para añadir automáticamente metadatos de edición sobre conceptos y recursos. Sin embargo, pensamos que en una primera aproximación conceptual esta funcionalidad puede resultar compleja para una mejor comprensión de nuestra propuesta, por lo que no será incluido en el modelado de la aplicación.

4.4. Formalización de la propuesta con UML

Las descripciones de las principales funciones de la aplicación, así como su estructuración en torno a la interfaz de usuario constituyen un paso previo al proceso de modelado de la aplicación. El modelo propuesto debe ser expresado en un lenguaje que permita simplificar, abstraer, representar y comunicar las funciones anteriormente expuestas.

Es conveniente utilizar un lenguaje que permita al mismo tiempo representar en un primer nivel aspectos conceptuales del sistema y su descripción más detallada en niveles más próximos al desarrollo de la aplicación. Esto permitirá obtener una especificación normalizada de las características del sistema.

En un principio nuestra propuesta se orienta al desarrollo futuro de una aplicación web para la gestión y uso de tesauros representados con SKOS. Sin embargo pensamos que resulta necesario aplicar un lenguaje de modelado de ámbito general, de amplia difusión y que permita la descripción y representación de aspectos conceptuales asociados a los elementos estructurales, el comportamiento del sistema en relación a los usuarios y la interacción entre los mismos, los elementos del sistema y el flujo de control de la aplicación.

En el presente trabajo se hace uso de UML⁸³ para la especificación de los aspectos anteriormente referidos. UML realiza la especificación de un sistema de software mediante diagramas que abarcan la representación de diversas características. En la versión 2.0 de UML existen trece tipos de diagramas:

- Diagrama de clases: describe la estructura del sistema mostrando las clases, atributos y relaciones entre las clases del sistema.

- Diagrama de componentes: representa la estructura del sistema y como se

⁸³ UML: *Unified Modeling Language*. Se trata de un lenguaje gráfico, basado en diferentes tipos de diagramas y que permite la especificación de sistemas de software. La versión UML 1.1 fué adoptado por OMG en noviembre de 1997. La versión 2.0 fue adoptada en 2005.

relacionan los diversos componentes.

- Diagrama de estructura compuesta: muestra la estructura interna de las clases del sistema y la colaboración con otras.
- Diagrama de despliegue: permite modelar el hardware utilizado en el desarrollo del sistema de software, así como los componentes desplegados y las relaciones que se producen entre sí.
- Diagrama de objetos: describe la estructura del sistema en un determinado momento.
- Diagrama de paquetes: muestra la distribución mediante agrupaciones lógicas y las dependencias entre dichas agrupaciones.
- Diagrama de actividades: representa el flujo de trabajo producido durante los procesos de negocio y las operaciones de los componentes del sistema modelado.
- Diagrama de casos de uso: permite modelar las funcionalidades del sistema en forma de actores y operaciones realizadas por ellos.
- Diagrama de estados: se trata de una notación normalizada para la descripción de clases, sistemas o procesos de negocio.
- Diagrama de secuencia: describe como los objetos se comunican entre sí mediante mensajes a lo largo de una línea de tiempo.
- Diagrama de comunicación: muestra las interacciones entre objetos en forma de mensajes secuenciales y representa una combinación de información suministrada por los diagramas de clases, los diagramas de casos de uso y los diagramas de secuencia.
- Diagrama de vista general de interacción: es un diagrama especial en donde los nodos son diagramas de secuencia relacionados con arcos propios de un diagrama de actividades.
- Diagrama de tiempos: indica los tiempos que transcurren entre y durante las transacciones y mensajes entre elementos del sistema.

Estos diagramas se organizan en tres categorías:

- Diagramas de estructura: se centran en los elementos del sistema, identificando, describiendo y estructurando los mismos a través de los diagramas de clases, los diagramas de componentes, los diagramas de objetos, los diagramas de estructura compuesta y los de despliegue y paquetes.
- Diagramas de comportamiento: representan los eventos y actividades que se desarrollan dentro del sistema. Dentro de esta categoría entran los diagramas de casos de uso, los diagramas de actividades y los diagramas de estados.
- Diagramas de interacción: forman una subcategoría dentro de los diagramas de comportamiento, ya que representan los mismos aspectos pero desde el punto de vista del flujo de control de las acciones de la aplicación y los datos intercambiados entre los elementos del sistema. Son los diagramas de secuencia, los diagramas de comunicación, los de vista general de interacción, y los de tiempos.

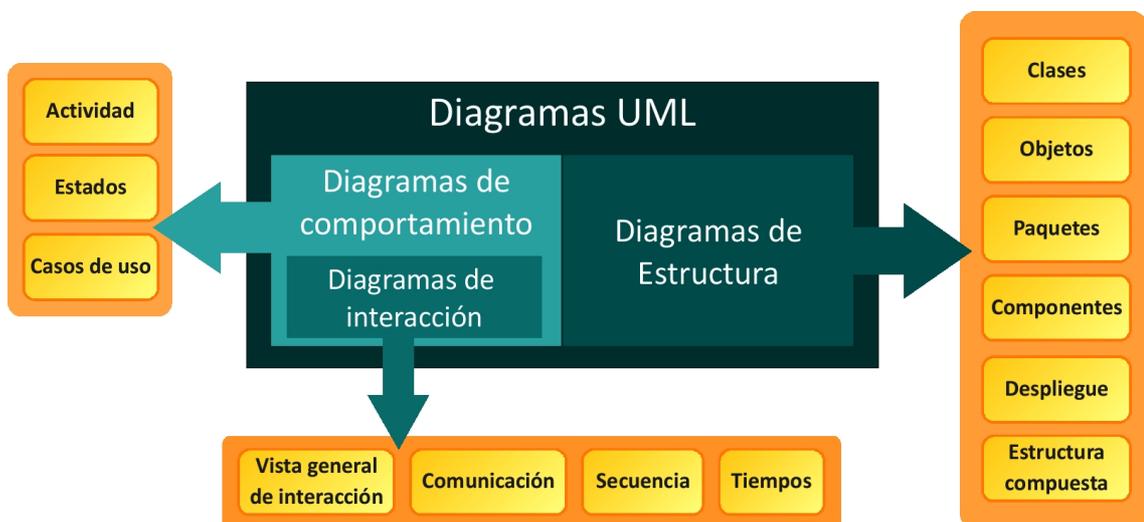


Ilustración 38: Tipología de diagramas UML

También se dispone de un mecanismo de ampliación denominado perfiles que permite la inclusión de nuevos tipos de descripciones y diagramas. Si bien UML no se diseñó de

forma específica para el desarrollo de aplicaciones web existen ciertos perfiles orientados a este fin como UWE⁸⁴ (Kroiß y Koch, 2008). Hemos escogido UML como lenguaje de modelado por este motivo además de su amplia difusión en el campo de desarrollo de sistemas y aplicaciones informáticas de todo tipo, junto con la existencia XMI⁸⁵ lo que permite el intercambio de este tipo de diagramas.

Para modelar conceptualmente una aplicación no es necesario elaborar todos los tipos de diagramas. No obstante, es recomendable realizar al menos los diagramas de clases y los diagramas de casos de uso. De esta forma es posible plantear una descripción conceptual completa adecuada para un análisis más profundo dirigido a la implementación del sistema en el que se elaborarán otros tipos de diagramas como los de secuencia o los de acciones y estados.

El diagrama de clases permite identificar los diferentes objetos del sistema, sus propiedades y las operaciones asociadas a los mismos. Por su parte, el diagrama de casos de uso organiza los principales procesos realizados por los usuarios (actores en terminología UML) ya se traten de personas o de otras aplicaciones.

Para la elaboración del modelo se ha procedido en primer lugar al diseño de los diferentes diagramas de casos de uso. Esto permite integrar las funcionalidades expuestas hasta ahora, mostrando una descripción de la estructura organizativa de la interfaz de usuario.

En un segundo paso se han descrito las diferentes clases de objetos necesarios para el desarrollo de los casos de uso identificados. Esta descripción incluye los atributos y operaciones más relevantes y las relaciones que se establecen entre las clases.

Finalmente se ha incluido un diagrama de secuencia que muestra aspectos lógicos de la dinámica del modelo. A pesar de que en principio con los dos primeros tipos de diagramas (casos de uso y clases) sería suficiente para realizar una primera aproximación conceptual al modelo (Arlow y Neudstadt, 2006), hemos optado por aportar el ejemplo mencionado a fin de mostrar un ejemplo de modelado de la dinámica específica de un caso de uso representativo. Ello permitirá una mejor

84 UWE: *UML-based Web Engineering*. Se trata de una metodología para el modelado y desarrollo de aplicaciones web. Está basada en UML y contempla aspectos relacionados con la interfaz navegacional del usuario.

85 XMI: *XML Metadata Interchange*. Es una especificación XML orientada al intercambio de diagramas.

comprensión de ciertos aspectos necesarios para el desarrollo completo del resto de diagramas de secuencia y de otro tipo, como los de actividades y estados, más cercano al desarrollo e implementación de aplicaciones. Todo lo anterior, junto con el uso de perfiles especializados (como UWE), personalizados para el diseño de los aspectos relacionados con la organización de los elementos de la interfaz navegacional o adaptados a cualquier otro entorno de desarrollo.

Para la elaboración de los distintos diagramas se ha utilizado la aplicación Umbrello⁸⁶, versión 2.1.3. Se trata de una herramienta con licencia GPL para el diseño de diagramas UML. Está desarrollada para Linux, principalmente para KDE⁸⁷, aunque es perfectamente operativa en otros entornos de escritorio como Gnome⁸⁸.

Umbrello permite el diseño de nueve tipos de diagramas UML, entre los que se encuentra los de casos de uso, clases, secuencia, actividades y estados o componentes. Esta herramienta permite el uso de UML 2 y XMI, lo que supone una gran ventaja para el intercambio de especificaciones para futuros desarrollos. Además asiste en cierta medida en la generación de esquemas de código para lenguajes de programación como Java o PHP.

Aunque según UML no es necesario realizar ningún tipo de descripción sobre los diferentes diagramas⁸⁹ se han incluido algunas aclaraciones sobre ciertos aspectos que pueden ayudar a encaminar de un modo más eficiente los desarrollos y aplicaciones posteriores del modelo.

86 Esta herramienta fue desarrollada por Paul Hensgen. Puede encontrarse más información en el sitio web: <http://uml.sourceforge.net/>

87 KDE: *K Desktop Environment*. Se trata de un entorno de escritorio para sistemas X-Windows de Unix/Linux. Tiene licencia GPL aunque en su origen incorporaban librerías que no lo eran totalmente. La primera versión apareció en 1997 y actualmente está disponible la versión 4.1.4. Permite un alto grado de configuración visual de los diferentes elementos del escritorio del usuario.

88 Gnome: *GNU Network Object Model Environment*: Es un entorno de escritorio surgido como alternativa a KDE y cuyo desarrollo se rige bajo la filosofía de la licencia GPL. Permite personalizar ciertos aspectos, poniendo especial énfasis en la accesibilidad y facilidad de uso y configuración.

89 UML es una herramienta con un lenguaje de modelado visual.

4.4.1. Diagramas de casos de uso

Los diagramas de casos de uso se organizan en forma de procesos relacionados entre sí y en los que intervienen diferentes actores. Las relaciones permiten ampliar el nivel de detalle, generalizar procesos o indicar su naturaleza opcional.

Se han realizado seis diagramas de casos de uso en los que se ha aplicado una técnica de generalización de los diferentes roles introduciendo un actor abstracto al que se ha denominado “*usuario*”. Para ello, se ha introducido un actor abstracto denominado usuario. El número de roles se restringe a medida que los diagramas de clases representan ámbitos con permisos operativos más amplios.

Los seis diagramas de casos de uso elaborados son los siguientes:

- ESC-00 Consulta de tesauros y repositorios.
- ESC-01 Gestión de tesauros y repositorios.
- ESC-02 Funciones de flujo de trabajo.
- ESC-03 Configuración de tesauros y repositorios.
- ESC-04 Configuración del sistema.
- ESC-05 Servicios web.

Los actores se representan mediante un icono específico, mientras que para cada caso de uso se utilizan elipses. Los elementos se vinculan mediante relaciones que pueden estar etiquetadas. La relación de comunicación (<<communicate>>) vincula a un actor con un caso de uso.

También es posible establecer relaciones entre casos de uso. La relación de inclusión (<<include>>) sirve para enriquecer un caso de uso con otro de forma imperativa. Por su parte la relación de extensión (<<extend>>) enriquece el caso de uso de forma análoga a la de inclusión pero de forma opcional. Para facilitar la claridad y comprensión de los diagramas únicamente se han etiquetado las relaciones de comunicación e inclusión. Por tanto, aquellas relaciones que no han sido etiquetadas son de extensión.

Los actores y casos de uso se identifican con una notación textual. Para los actores se ha utilizado la notación "ACT-" junto con un código numérico de dos dígitos que permite identificar de forma unívoca a cada actor. De esta forma puede consultarse en qué casos de uso interviene cada uno de los roles del sistema.

De igual forma se ha asignado una notación para las operaciones representadas en cada uno de los diagramas de casos de uso. Dicha notación combina la notación "CU-", una letra identificativa del diagrama en el que se define la operación y un código con dos dígitos numéricos, como por ejemplo "CU-C12". En ocasiones ha sido necesario ampliar el nivel de profundidad mediante indicaciones descendentes.

Se ha utilizado la técnica de herencia para generalizar los diferentes actores en un único actor abstracto denominado "Usuario" al que se le ha asignado el código "ACT-00". Esta técnica simplifica la complejidad de los diagramas de casos de uso. Algunos casos de uso son reutilizados en varios diagramas para una estructuración óptima y reutilizable de los componentes del modelo.

Puede observarse como los usuarios comienzan a consultar los contenidos a partir de una lista de tesauros y repositorios (CU-C01). En el caso de usuarios que utilicen el sistema sin identificarse previamente únicamente podrán consultar los tesauros y repositorios públicos. A los usuarios autenticados se les ofrecerá, además, una lista de los elementos generales sobre los que tienen algún tipo de permiso.

4.4.1.1. ESC-00 Consulta de tesauros y repositorios

Este diagrama muestra la interacción entre los usuarios y el sistema para la consulta de tesauros y repositorios. Se trata del único escenario de casos de uso (junto con el de servicios web) en donde no es necesario la autenticación para utilizar la aplicación.

Los usuarios anónimos y los clientes de servicios web únicamente podrán acceder a los tesauros y los repositorios marcados como públicos, mientras que el resto de usuarios podrán acceder además a los tesauros y repositorios para los que tienen, al menos, privilegios de lector.

Puede observarse como los casos de uso centrales de este escenario se corresponden con la consulta de conceptos (CU-C05) y la de metadatos de recursos (CU-C13). La interacción entre los procesos de consulta de los tesauros y los repositorios se producen en los casos de uso de búsqueda conceptual (CU-C18) y consulta de la indización de un recurso (CU-C18).

Únicamente los marcadores de búsqueda son utilizados en este escenario, por lo que en el caso de los usuarios anónimos la persistencia de los mismos se limitará a la duración de la sesión de consulta del sistema.

La diferencia entre el usuario anónimo (ACT-06) y el usuario lector (ACT-05) radica en que el primero solo tiene acceso a la consulta de aquellos tesauros y recursos marcados como públicos, mientras que un usuario lector podrá también consultar aquellos a los que se le haya concedido los permisos de lectura, sean o no públicos.

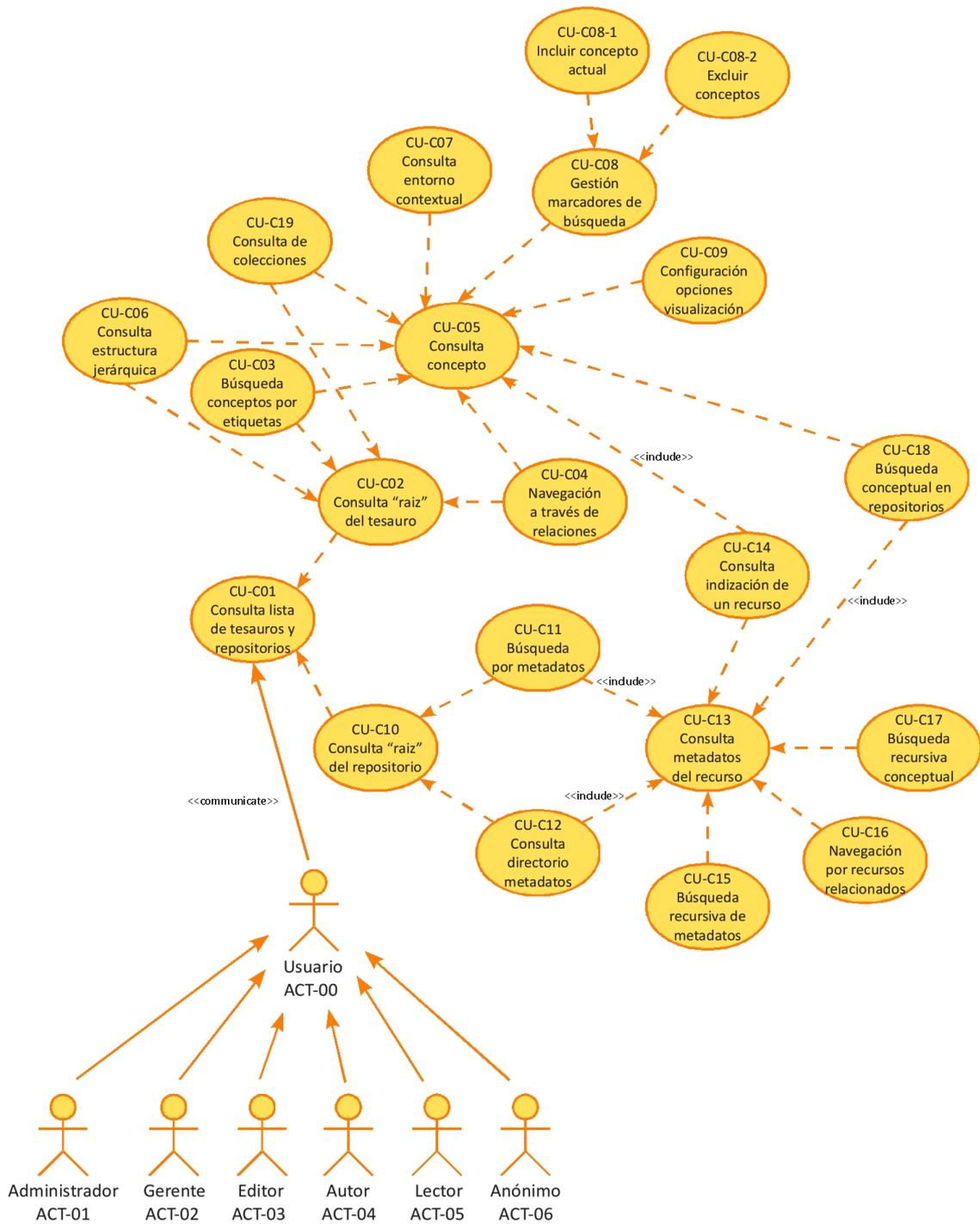


Ilustración 39: Diagrama de casos de uso de consulta de tesauros y repositorios

4.4.1.2. ESC-01 Gestión de tesauros y repositorios

Este escenario representa las funciones de gestión de los diversos elementos que componen los tesauros y los repositorios. En esta ocasión únicamente los usuarios con un nivel de privilegios de autor, editor, gerente o administrador podrán acceder a las funciones de gestión. Los permisos de acceso a dichas funciones se definen de forma separada para tesauros y para repositorios.

El diseño se ha realizado teniendo en cuenta que los procesos de consulta están integrados con los de mantenimiento, por lo que la creación de conceptos, etiquetas, relaciones, recursos, metadatos y colecciones se realiza desde los casos de uso de consulta de conceptos y recursos (CU-C05 y CU-C13 respectivamente).

Los pares de elementos tesauros-conceptos y repositorios-recursos comparten la gestión de metadatos, tal y como puede comprobarse en los casos de uso CU-G02 y CU-G16.

No existen casos de uso para la modificación de elementos dentro de la gestión de relaciones (CU-G04), conceptos (CU-G03) y colecciones (CU-G07) ya que dichos elementos únicamente puede crearse o eliminarse al no contener atributos susceptibles de ser modificados.

En el caso del actor Autor (ACT-04) la creación de cualquier elemento implicará en realidad una propuesta. De esta forma el elemento es creado, pero nunca publicado en tanto no se realice el proceso de validación posterior. Todas las propuestas que realiza un mismo usuario pueden ser consultadas o borradas accediendo al caso de uso CU-G18 (común a tesauros y repositorios).

Por último indicar la equivalencia entre los casos de uso CU-C19 y CU-G07 tanto para la consulta como para la gestión de conceptos.

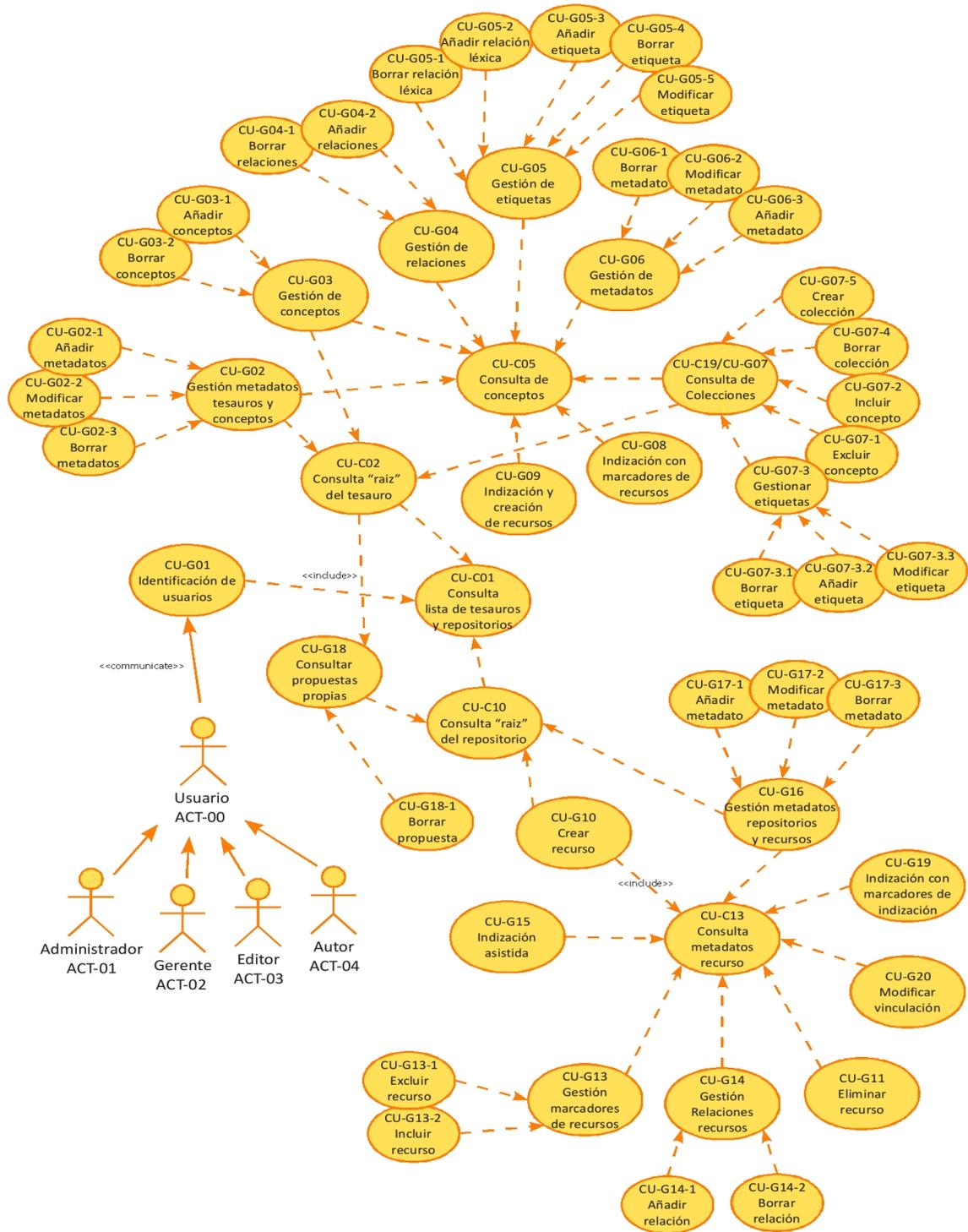


Ilustración 40: Diagrama de casos de uso de gestión de tesauros y repositorios

4.4.1.3. ESC-02 Funciones de flujo de trabajo

A partir de la consulta de la “raíz” del tesoro o repositorio es posible acceder (por parte de administradores, gerentes y editores) a las listas de elementos pendientes de validar. De un modo más específico también será posible acceder a las propuestas pendientes asociadas a conceptos y repositorios para su validación. Los casos de uso para la validación/rechazo (CU-T04) y la reasignación (CU-T05) de propuestas es compartida tanto en la gestión de tesauros como de repositorios. También se ha incluido en este escenario los casos de uso para la importación de elementos (CU-T01 y CU-T06) al coincidir los actores permitidos para la realización de estas operaciones con los de los procesos de validación de propuestas.

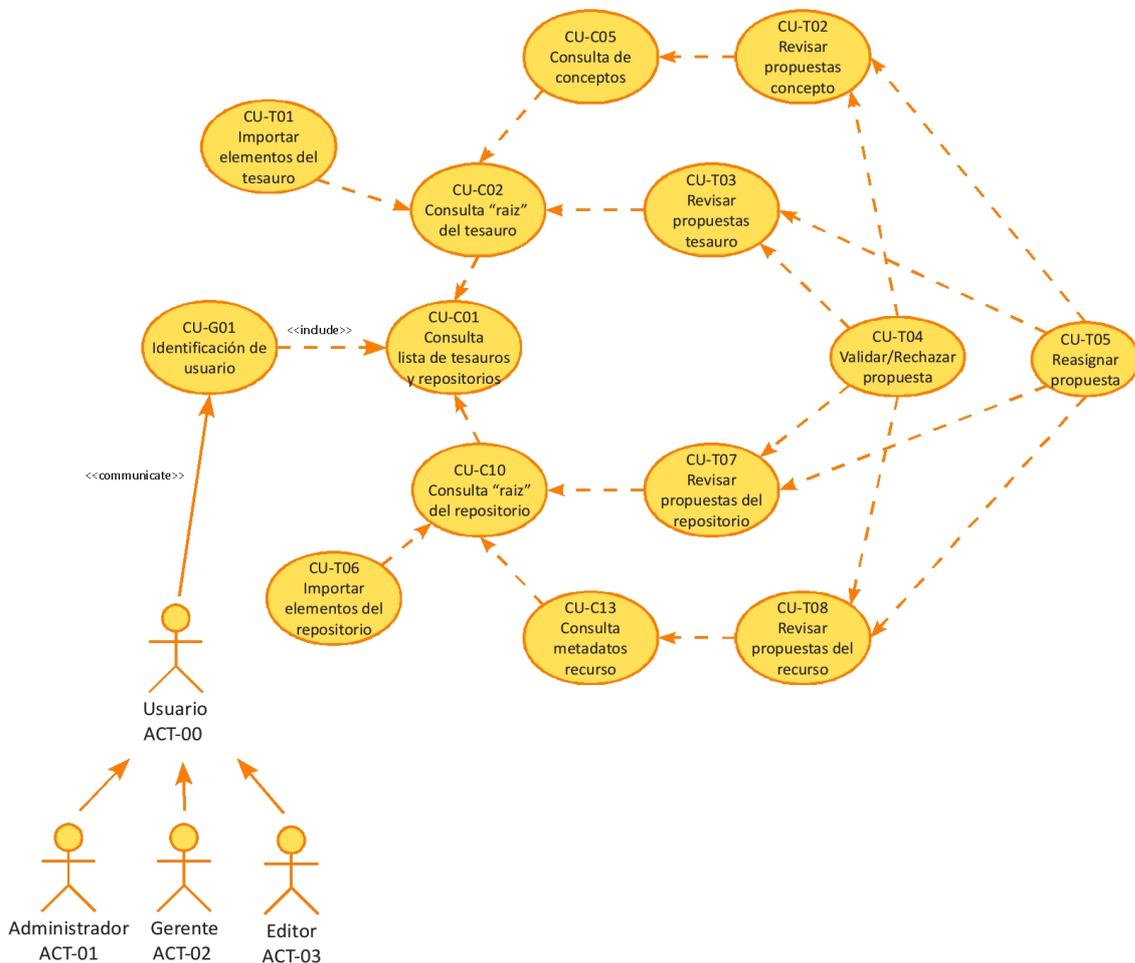


Ilustración 41: Diagrama de casos de uso de funciones de flujo de trabajo

4.4.1.4. ESC-03 Configuración de tesauros y repositorios

Este escenario tiene como objetivo modelar los aspectos relativos a las opciones de configuración de tesauros y repositorios para la gestión de grupos de usuarios, modelos de metadatos aplicables, sistema de validación de propuesta, comunicación a través de correo electrónico de los procesos de flujo de trabajo, tipología de relaciones del tesoro y selección de los elementos de metadatos disponible para la búsqueda y consulta de directorios en los repositorios. Únicamente los usuarios con rol de administrador o gerente podrán tener acceso a las funcionalidades planteadas en este escenario.

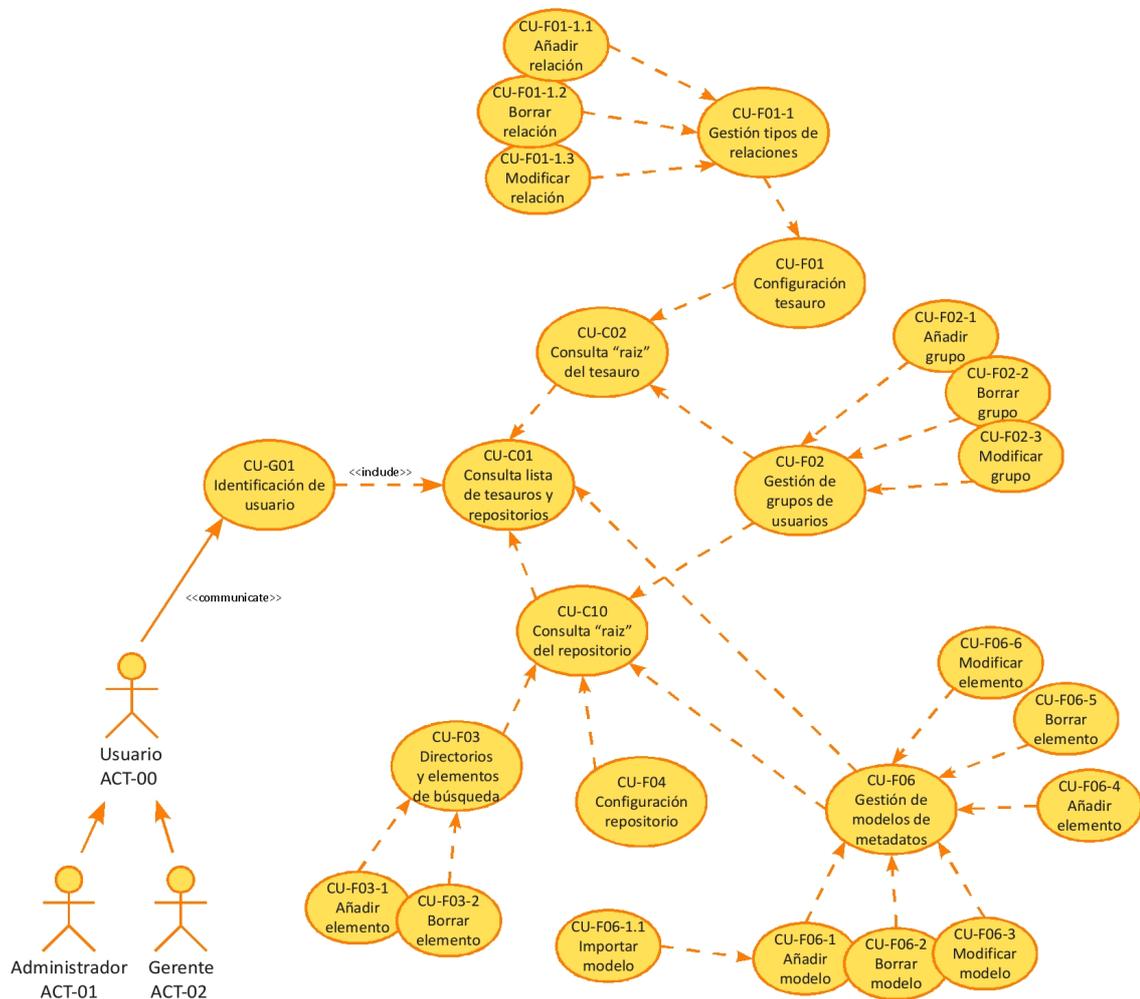


Ilustración 42: Diagrama de casos de uso de configuración de tesauros y repositorios

4.4.1.5. ESC-04 Configuración del sistema

Los usuarios administradores son los únicos que pueden utilizar las funciones de este escenario, tal y como se indicó en el análisis de requisitos. Así, la creación de tesauros y repositorios corresponde al administrador, al igual que la gestión general de grupos y usuarios (CU-S03).

Otros aspecto gestionado en este escenario es la gestión de modelos de metadatos a nivel general, es decir, aprovechables desde cualquier elemento (CU-F06), así como los idiomas disponibles en el sistema (CU-S05).

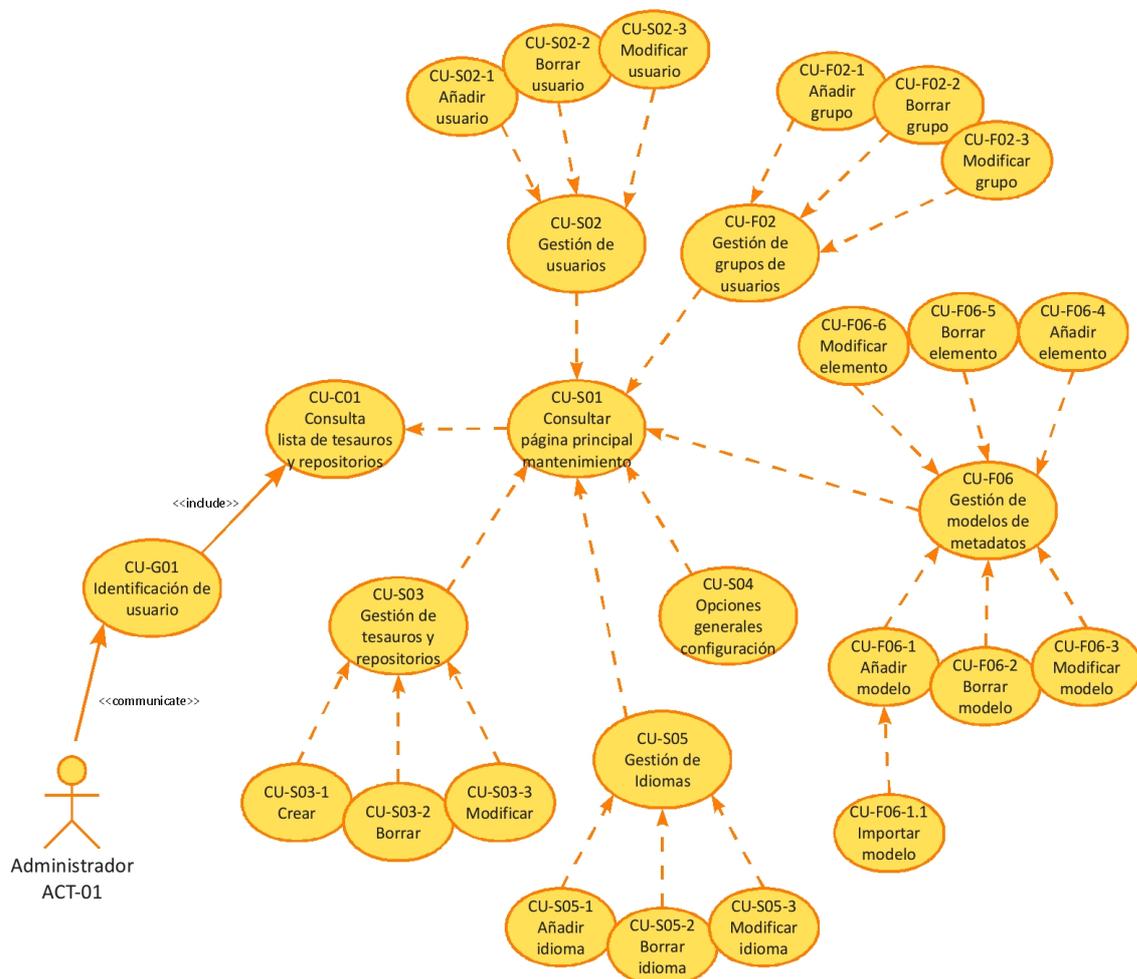


Ilustración 43: Diagrama de casos de uso de configuración del sistema

4.4.1.6. ESC-05 Servicios web

El último escenario presentado se corresponde con los servicios web disponibles. En un primer análisis superficial puede apreciarse una similitud con la consulta de los contenidos del sistema. No obstante, se han seguido las propuestas del apartado 4.2.5.

El diseño del escenario obedece al intento de establecer un equilibrio entre el volumen de datos descargados en una sola petición y el número de peticiones necesarias para explotar las funcionalidades del sistema. Este es el motivo por el que la recuperación de un concepto o una relación implica la recuperación de las etiquetas correspondientes. Finalmente indicar que la estructuración lógica de los diferentes casos de uso obedece a la secuencia lógica de acceso a la información del sistema por parte de cualquier aplicación externa.

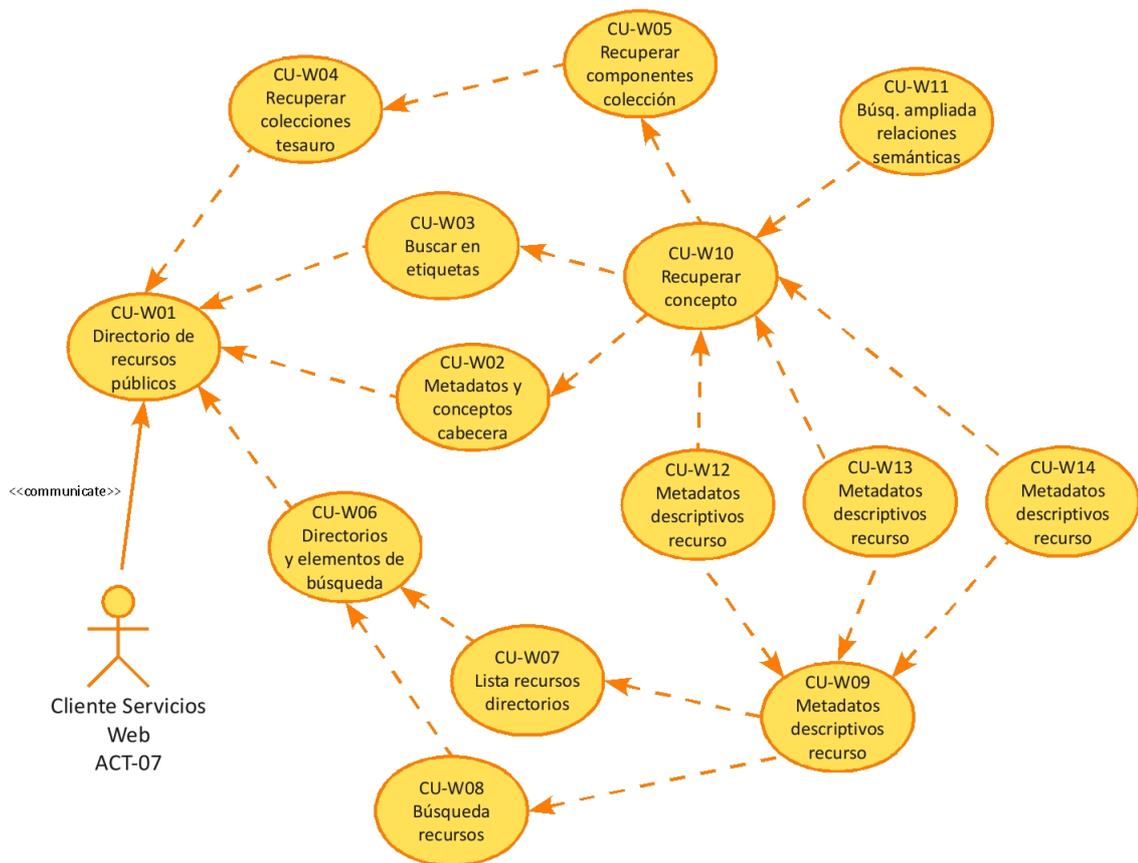


Ilustración 44: Diagrama de casos de uso de servicios web

4.4.2. Diagramas de clases

Los diagramas de clases permiten representar los diferentes objetos del sistema mediante clases. Una clase es un conjunto de objetos similares, con la misma estructura y comportamiento, y por lo tanto con los mismos atributos y métodos. Un atributo se refiere a una determinada cualidad de un objeto, mientras que un método agrupa una serie de sentencias para realizar determinadas operaciones sobre atributos de la clase que representa el objeto.

Cada clase se representa con un rectángulo dividido en tres secciones verticales. La primera sección o encabezado contiene el nombre de la clase, la segunda enumera los atributos y la tercera sección se reserva para la lista de métodos. Tanto los métodos como los atributos pueden ser públicos (visibles para cualquier otro objeto del sistema), protegidos (visibles en las subclases de la clase), privados (visibles únicamente en la clase) o empaquetados (visibles solo en las clases del mismo paquete de clases). Los atributos pueden representarse con tipos de datos asociados.

Las clases pueden relacionarse entre sí a través de asociaciones con diversos tipos de cardinalidad. Además el diagrama de clases permite representar aspectos tales como la generalización de clases o la composición de objetos. Las asociaciones entre clases también pueden describirse en ambos sentidos. Generalmente las asociaciones son bidireccionales, pudiéndose dar el caso de asociaciones en un único sentido, lo cual se representa mediante una flecha que indica el sentido de la navegación entre clases.

La existencia de SKOS como modelo de datos para la representación de tesauros ha permitido una identificación previa de gran parte de las clases de objetos del sistema. Para facilitar la elaboración y comprensión de la estructura global de clases de la aplicación se han confeccionado nueve diagramas de clases que muestran diferentes aspectos del sistema:

- CLA-00 Estructura de nodos
- CLA-01 Estructura de etiquetas
- CLA-02 Estructuración de relaciones
- CLA-03 Usuarios, grupos y permisos
- CLA-04 Elementos de documentación

- CLA-05 Elementos y asignación de metadatos
- CLA-06 Indización conceptual de recursos
- CLA-07 Control de flujo de trabajo
- CLA-08 Marcadores

Se han representado exhaustivamente todas las clases de objetos del sistema con sus atributos y métodos. Por supuesto el modelo puede ser enriquecido una vez se haya aplicado en el desarrollo de herramientas concretas, por lo que la modificación de los diagramas de clases puede ser un hecho que se produzca en el futuro con la finalidad de completar o adaptar determinadas características mediante la modificación de la estructura de atributos y métodos.

Los elementos susceptibles de ser propuestos y validados tienen asociado un atributo que indica tal hecho y que hemos denominado “estado”. Los posibles valores que puede tomar ese atributo indicarán el estado de validación del elemento. Aunque dichos códigos serán establecidos en el momento de implementación de la aplicación a modo de ejemplificar la descripción de las diferentes posibilidades realizaremos la siguiente propuesta:

- 00: Elemento validado sin ninguna restricción para ser consultado o realizar propuestas sobre el mismo.
- 01: Validado pero inhabilitado para realizar propuestas de cambio de estado.
- 10: Creación pendiente de validación.
- 11: Eliminación del elemento pendiente de validación.
- 12: Cambios sobre el elemento pendientes de validación.
- 20: Validado pero inhabilitado para su consulta o para realizar propuestas de cambio de estado.

Ciertos métodos hacen uso de estos códigos para acceder a determinados objetos. A continuación se muestran varios ejemplos de gran utilidad para comprender la sintaxis de invocación. Con el primero se recuperan aquellas etiquetas preferentes de un determinado concepto que han sido validadas y están habilitadas para ser consultadas.

4. Propuesta de modelo conceptual de aplicación

El segundo ejemplo permite acceder a las etiquetas preferentes pendientes de validación y propuestas por un usuario en concreto y asignadas a un concepto. El último ejemplo muestra como recuperar cualquier tipo de etiqueta (preferente o alternativa) se encuentre validada o no.

```
1. ListaEtiquetas1 = etiquetaNodo(ConceptoX, "prefLabel", "0*")
2. ListaEtiquetas2 = etiquetaNodo(ConceptoX, "prefLabel", "1*", UsuarioZ)
3. ListaEtiquetas3 = etiquetaNodo(ConceptoX, "*")
```

Ejemplo 18: Ejemplos de invocación del método "etiquetaNodo"

Desde el punto de vista dinámico la propuesta de creación, eliminación o modificación de un elemento por parte de un usuario con rol de autor (correspondiente al actor ACT-04) implicaría la creación de dicha propuesta y del objeto correspondiente. Sin embargo, el valor del atributo "*estado*" de dicho objeto indicará que la propuesta está pendiente de validación. Ciertos atributos pueden ser susceptibles de ser modificados en dicha propuesta, por lo que será necesario duplicar dichos campos para almacenar el valor actual y el valor propuesto. Por este motivo, en algunas clases (como en "*Etiqueta*", "*Notación*", "*Documentación*", etc) puede observarse dicha duplicación. En los campos con sufijo "*...previo*" se almacenarán los valores propuestos a través de la invocación del método "*previo*". Para actualizar los valores de los atributos referidos a los valores actuales con los valores propuestos se ha de utilizar el método "*actualizarPrevios*".

A continuación se incluyen los diagramas de clases del modelo y las descripciones sobre las estructuras de objetos y relaciones entre clases, así como los aspectos más relevantes acerca de los atributos y métodos definidos. En cada diagrama se representan determinados aspectos del modelo, habiéndose incluido únicamente las clases y relaciones relevantes para ello.

4.4.2.1. CLA-00 Estructura de nodos y etiquetas

Puesto que una de las características de los elementos de SKOS es su identificación con el concepto de recurso RDF hemos trasladado dicha estructura a la organización de clases del modelo. De esta forma hemos cosntruido la clase “*Nodo*” para generalizar las subclases “*Repositorio*”, “*Recurso*”, “*Colección*”, “*Tesaurus*” y “*Concepto*”.

La (relación “*contenido*”) muestra la asociación de recursos de información a sus repositorios. Un recurso pertenece a un único repositorio. La eliminación de un repositorio implica el borrado de los recursos de información contenidos en aquel.

Un concepto puede estar incluido en varias colecciones, pero el borrado de una colección o un concepto no implica al otro elemento de la relación. Esta relación se ha caracterizado para permitir los procesos de propuesta y validación de forma independiente a conceptos y colecciones mediante la clase de asociación “*InclusiónConcepto*”.

En este diagrama se ha incluido la clase “*Notación*” para representar esta característica de SKOS, y se ha asociado a la subclase “*Concepto*” a través de la relación “*notaciónConcepto*”. La supresión de un concepto conlleva la eliminación de las notaciones asociadas al mismo. La clase “*Nodo*” incluye el atributo “*tipo*” para denotar las diferentes subclases. A fin de incluir la función de asignar a cada nodo un identificador único se ha definido un atributo de igual nombre. La subclase “*Concepto*” incluye un atributo para marcar el carácter compartido de un objeto de este tipo.

En las subclases “*Tesaurus*” y “*Repositorio*” se ha incluido el atributo “*umbralValidación*” que indica el número de confirmaciones necesarias por parte de editores distintos para la validación de un elemento. En estas mismas subclases se ha proporcionado un mecanismo para almacenar la URI base para referenciar los elementos que contengan (“*uri_base*”).

En “*Colección*” el atributo “*tipoColección*” permite indicar el tipo de colección (convencional u ordenada). La clase de asociación “*InclusiónConcepto*” contiene ciertos métodos para gestionar la pertenencia y orden de conceptos a una colección.

Las clases incluyen los métodos necesarios para crear, eliminar y actualizar los atributos de los objetos correspondientes. La subclase “*Repositorio*” incluye los

4. Propuesta de modelo conceptual de aplicación

métodos para obtener los elementos de metadatos definidos para los procesos de búsqueda y la descripción de metadatos (“*camposBúsqueda*” y “*camposDirectorios*”), así como para obtener todos los objetos de la clase “Recurso” asociados a un repositorio.

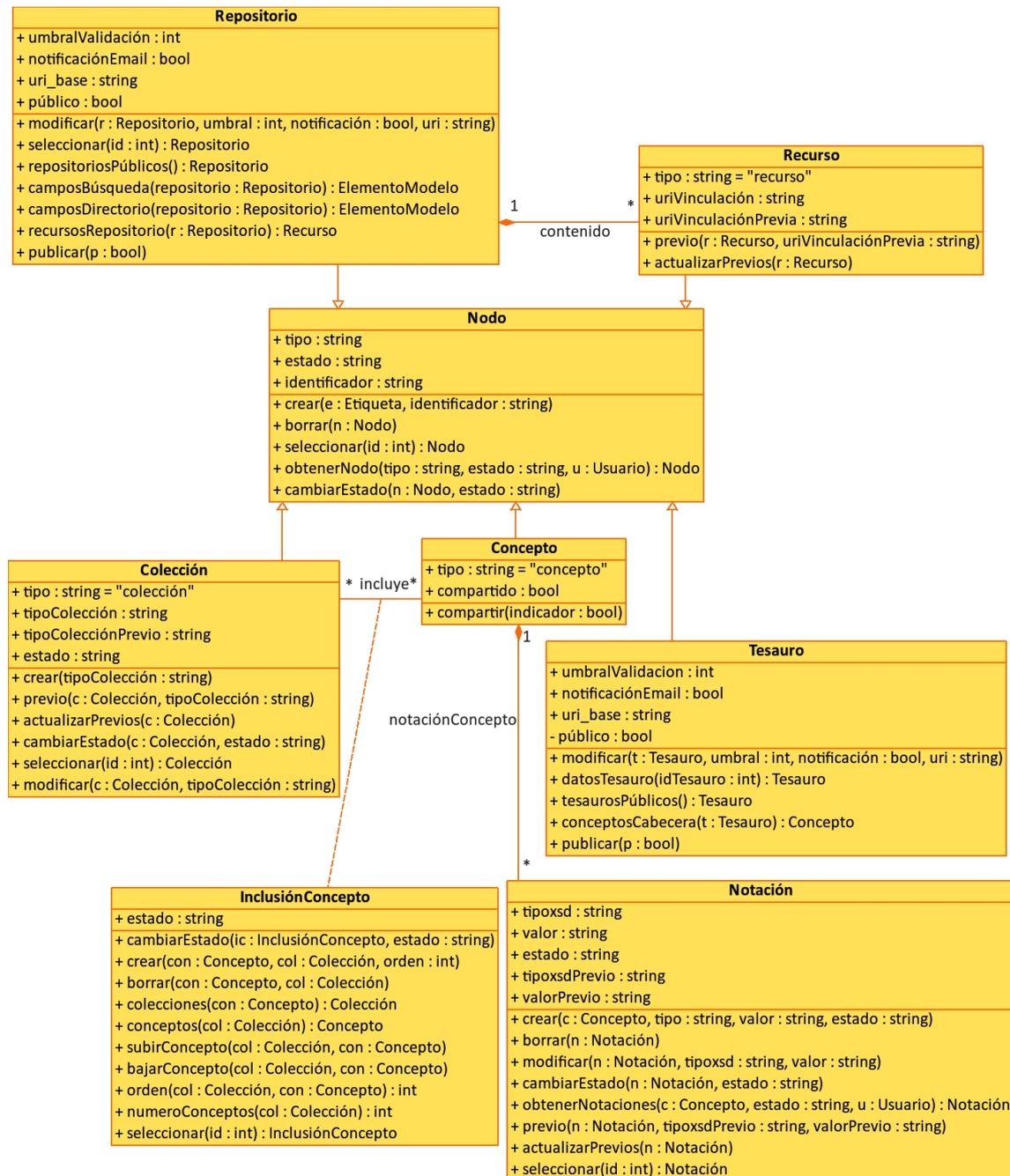


Ilustración 45: Diagrama de clases con la estructura de nodos

4.4.2.2. CLA-01 Estructura de etiquetas

Las subclases definidas en el diagrama anterior y asociadas a la clase “*Nodo*” son susceptibles de ser etiquetas. Por tanto se ha definido la clase “*Etiqueta*” que generalizan las subclases “*EtiquetaAlternativa*” y “*EtiquetaPreferente*”. Estas subclases se asocian a un objeto de la clase “*Nodo*” (relación “*etiquetaNodo*”) que puede tener asociadas varias etiquetas de cada tipo, sin embargo es necesario tener en cuenta que para cada idioma únicamente podrá existir una etiqueta preferente. Cada etiqueta está asociada a un único idioma (relación “*idiomaEtiqueta*”). La eliminación de un nodo conllevará el borrado de las etiquetas que tuviera asignadas.

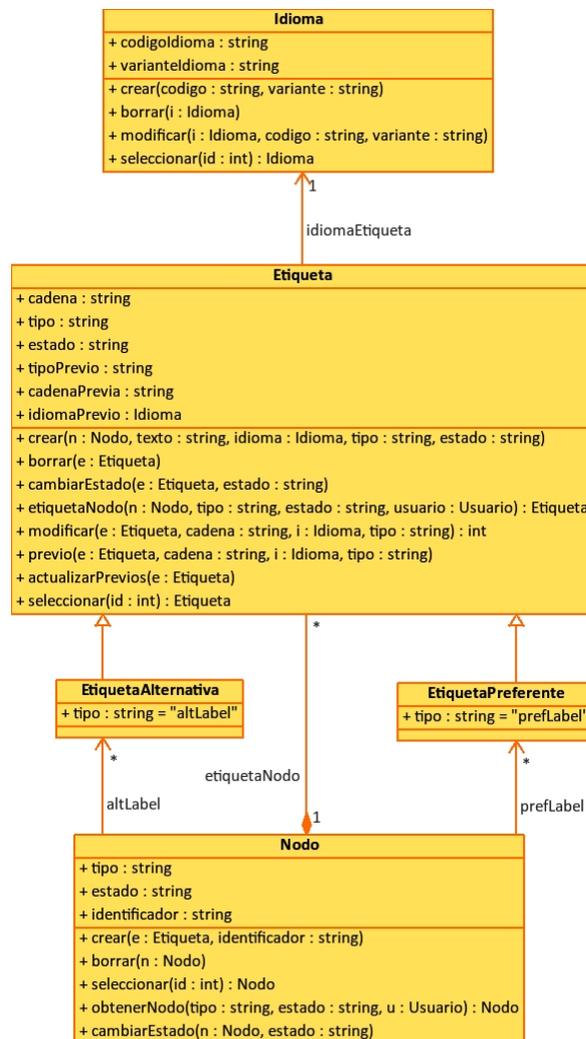


Ilustración 46: Diagrama de clases con la estructura de etiquetas

4.4.2.3. CLA-02 Estructura de relaciones

Para modelar las relaciones de cualquier tipo entre elementos del sistema se ha definido una clase “*Relación*” que generaliza la tipología de relaciones que pueden crearse entre conceptos tanto a nivel semántico como de correspondencia (“*RelaciónConceptual*”), entre conceptos y tesauros a través de la propiedad `skos:inScheme` (“*RelaciónPerteneencia*”), las relaciones entre etiquetas (“*RelaciónLéxica*”), o entre recursos (“*RelaciónRecurso*”).

Puede observarse en este diagrama que los distintos tipos de relaciones se han construido como clases asociadas a las relaciones entre los objetos del sistema. Este es el motivo por el que los métodos de creación se encuentra en cada una de las clases asociadas. La eliminación de algún elemento del sistema implicará la eliminación automática de las relaciones en las que participe.

Asimismo, la supresión de un tesauro supondrá el borrado de los conceptos asociados al mismo, y según se ha indicado anteriormente de sus relaciones y etiquetas.

Las características de cada tipo de relación se describen mediante la clase “*TipoRelación*” que contiene los atributos necesarios para ello. El atributo “*código*” permite asignar una etiqueta descriptiva a la relación, mientras que “*relaciónFija*” permite identificar las relaciones que no pueden ser eliminadas del sistema.

El método “*relaciones*” presente en las clases asociadas “*RelaciónLéxica*”, “*RelaciónConceptual*” y “*RelaciónRecurso*” permite acceder a los objetos con los que está relacionado un determinado elemento.

Resultan de especial relevancia los métodos “*jerarquía*” y “*esTérminoCabecera*”, ya que permiten obtener los conceptos ubicados en una misma línea jerárquica o averiguar si un concepto no tiene establecida ninguna relación jerárquica genérica. De igual forma, la clase “*Tesauro*” ofrece el método “*conceptosCabecera*” para recuperar aquellos conceptos cabecera de un tesauro.

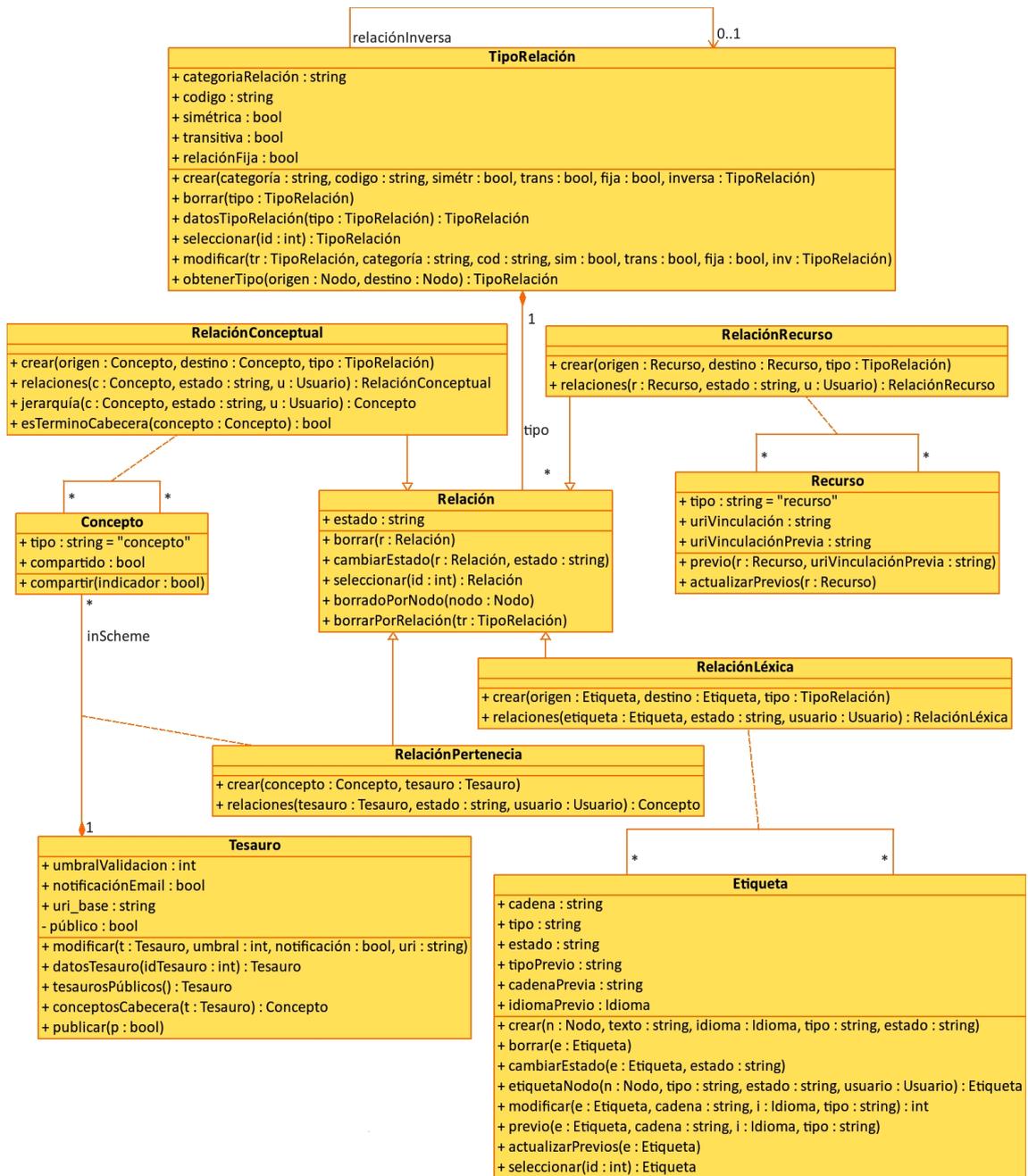


Ilustración 47: Diagrama de clases con la estructura de relaciones

4.4.2.4. CLA-03 Usuarios, grupos y permisos

Los usuarios del sistema pueden asociarse a tantos grupos como sea necesario. La definición del rol de usuario se define sobre los grupos a los que pertenezca. Cada grupo puede tener permisos a un número ilimitado de tesauros y repositorios. Se ha definido una clase abstracta *"nodoGeneral"* que generaliza las clases *"Tesauro"* y *"Repositorio"*.

Los permisos entre la clase *"Grupo"* y la clase abstracta mencionada se realizan a través de una relación caracterizada con la clase de asociación *"Permiso"*. Las clase *"Usuario"* y *"Grupo"* disponen de métodos para gestionar la pertenencia de los usuarios a los grupos, averiguar los miembros de un grupo o en cuales está incluido un determinado usuario. La relación *"grupoEditores"* asocia cada grupo con rol de autor con un grupo con rol de editor.

Los funcionalidad de los métodos para la inclusión y separación de usuarios en un grupo se ha duplicado ya que se han incluido en la clase *"Usuario"* (*"asociarGrupo"*, *"separarGrupo"*) y en la clase *"Grupo"* (*"asociarUsuario"*, *"separarUsuario"*). Esto permite una mayor flexibilidad para la asignación de permisos ya que mientras se gestionan los usuarios pueden asociarse a grupos y a la inversa. Este diagrama también muestra algunos aspectos adicionales, como la asignación de idiomas preferentes a cada tesauro, repositorio y usuario.

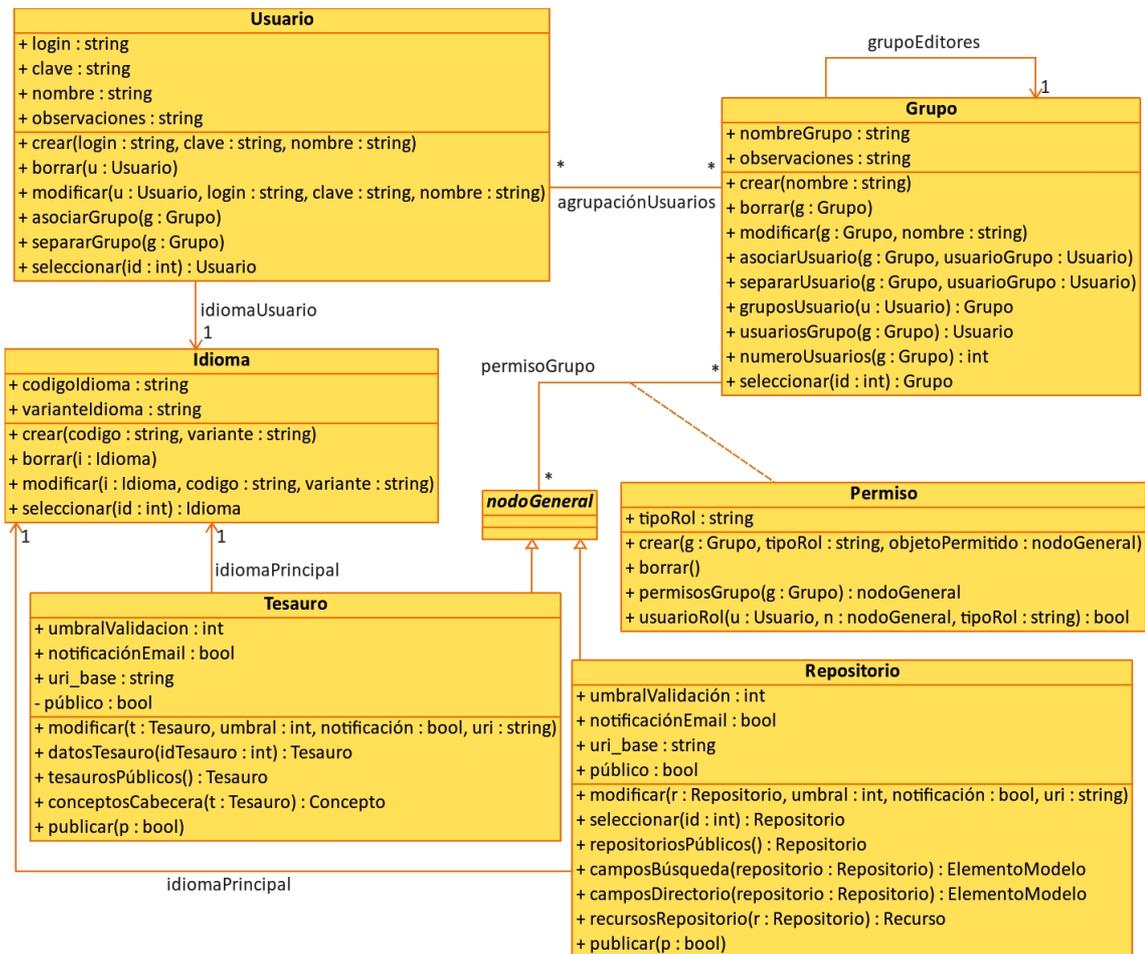


Ilustración 48: Diagrama de clases para usuarios, grupos y permisos

4.4.2.5. CLA-04 Elementos de documentación

Los elementos de documentación se asocian a cualquier tipo de nodo del sistema. La relación “documentaciónNodo” indica que un elemento de este tipo únicamente puede estar definirse únicamente para un nodo y además está elaborado en un idioma determinado. La clase “Documentación” generaliza los distintos tipos de elementos de documentación propuestos por SKOS a través de diferentes subclases sobre los que no se han definido ningún tipo de método, únicamente el atributo “tipo” para identificar el tipo de cada objeto.

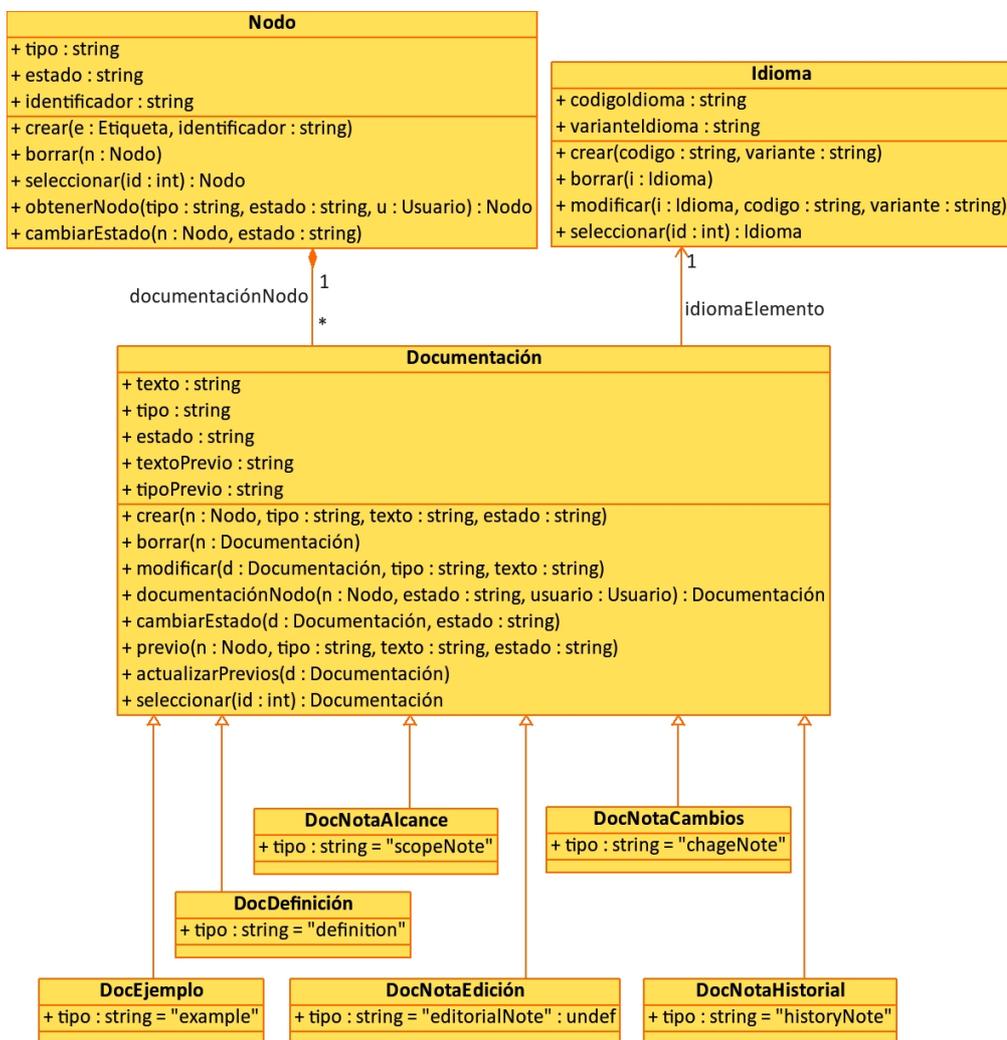


Ilustración 49: Diagrama de clases para los elementos de documentación

4.4.2.6. CLA-05 Elementos y asignación de metadatos

El modelo propuesto permite el almacenamiento de modelos de metadatos. Para ello se han definido las clases “*ModeloMetadatos*” para identificar el modelo y “*ElementoModelo*” para los diferentes elementos del vocabulario de dicho modelo.

Los atributos “*uri_namespace*” y “*descripción*” de la clase “*Modelo metadatos*” permiten almacenar la URI de un determinado modelo y una descripción para identificarlo (por ejemplo: “*http://purl.org/dc/elements/1.1/*” y “*Dublin Core*”).

Los modelos de metadatos se relacionan con aquellos tesauros en los que pueden aplicarse a través de la relación “*modelosTesauro*”. Los repositorios se relacionan con la clase “*ModeloMetadatos*” de tres formas distintas: para permitir el uso de un modelo determinado (“*modelosRepositorio*”) y para definir los elementos habilitados para la realización de búsquedas (“*campoBúsqueda*”) y la construcción de directorios (“*campoDirectorio*”).

A partir de la clase “*ElementoModelo*” pueden instanciarse los objetos de la clase “*Metadato*” que tendrán asociados un idioma concreto. Estos objetos pueden asociarse a nodos mediante la relación “*descripción*”.

4. Propuesta de modelo conceptual de aplicación

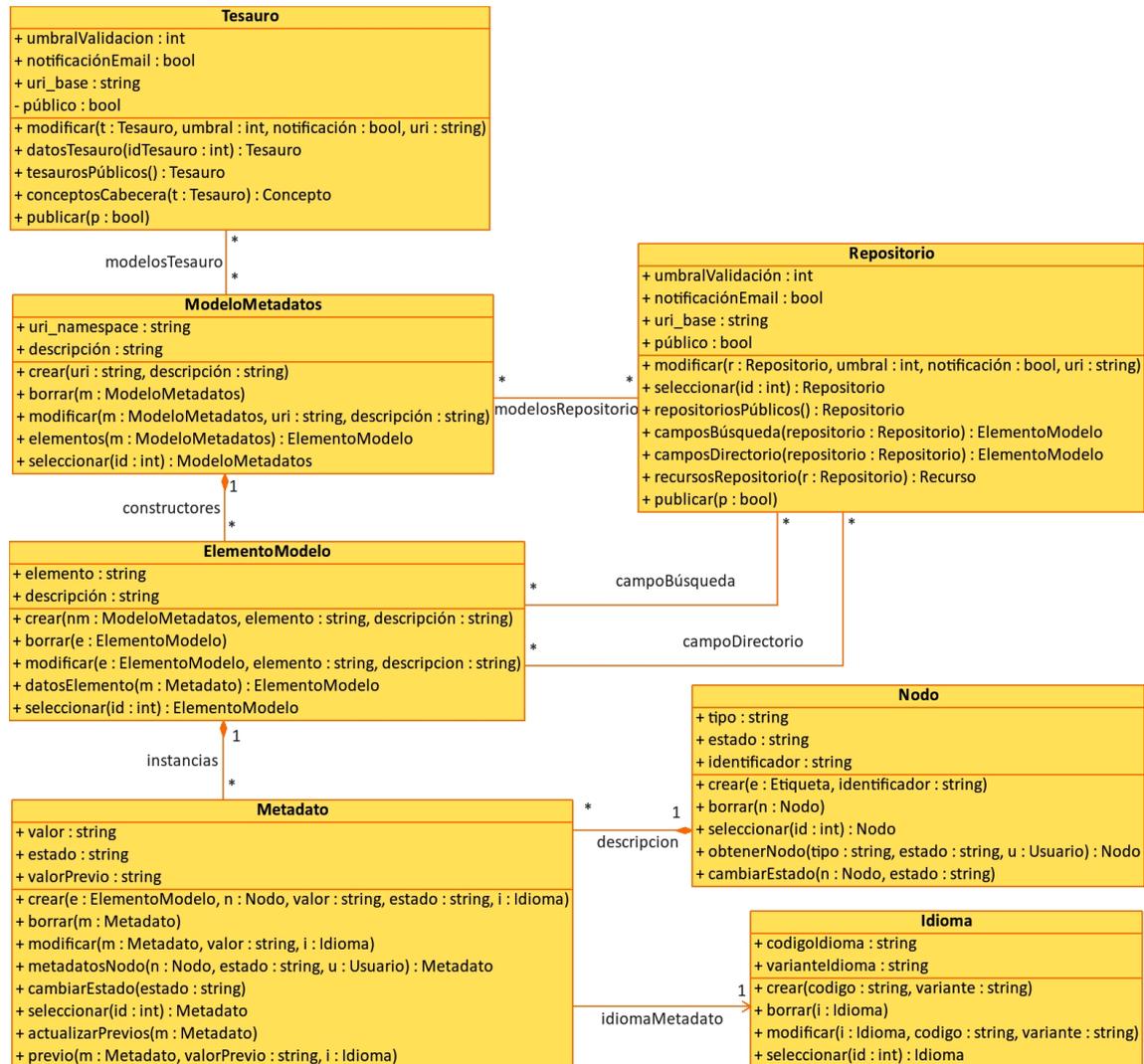


Ilustración 50: Diagrama de clases para los elementos y asignación de metadatos

4.4.2.7. CLA-06 Indización

Los conceptos tesauros pueden asignarse a recursos para realizar tareas de indización. La relación entre ambas clases de objetos se representa con la clase de asociación “Indización”. La eliminación de un concepto o un recurso hará que también desaparezca los índices correspondientes.

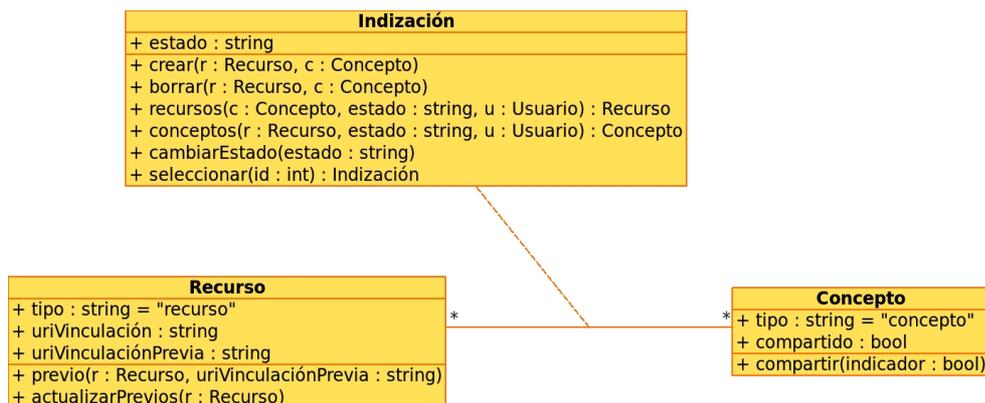


Ilustración 51: Diagrama de clases para la indización de recursos con conceptos

4.4.2.8. CLA-07 Control de flujo de trabajo

El modelo contempla las estructuras para el desarrollo de las funciones de flujo de trabajo. Los elementos susceptibles participar en este proceso son las relaciones, etiquetas, nodos, notaciones, asociaciones a colecciones y metadatos de nodos. Por tanto, en este diagrama participan las clases correspondientes a los elementos indicados anteriormente, generalizadas en la clase abstracta *“ElementoPropuesto”*.

Se ha declarado una clase para representar las propuestas. Cada propuesta se relaciona con la clase *“Usuario”* (para representar al usuario que la realiza o revisa), con el elemento objeto de la propuesta y con la clase *“Validación”*.

Así pues, los autores podrán realizar propuestas, que serán asignadas a los editores para su revisión, para que éstos, gerentes o administradores las validen. Por tanto, la clase *“Usuario”* debe relacionarse con las clase *“Propuesta”* y *“Validación”*.

La eliminación del usuario o el nodo objeto de la propuesta supone la supresión de las propuestas o validaciones asociadas. El borrado de una propuesta implica la eliminación de las validaciones correspondientes.

Es necesario considerar que una vez que se ha validado la propuesta no es posible rechazarla, aunque si hacer nuevas propuestas sobre el elemento que ha intervenido en la propuesta original.

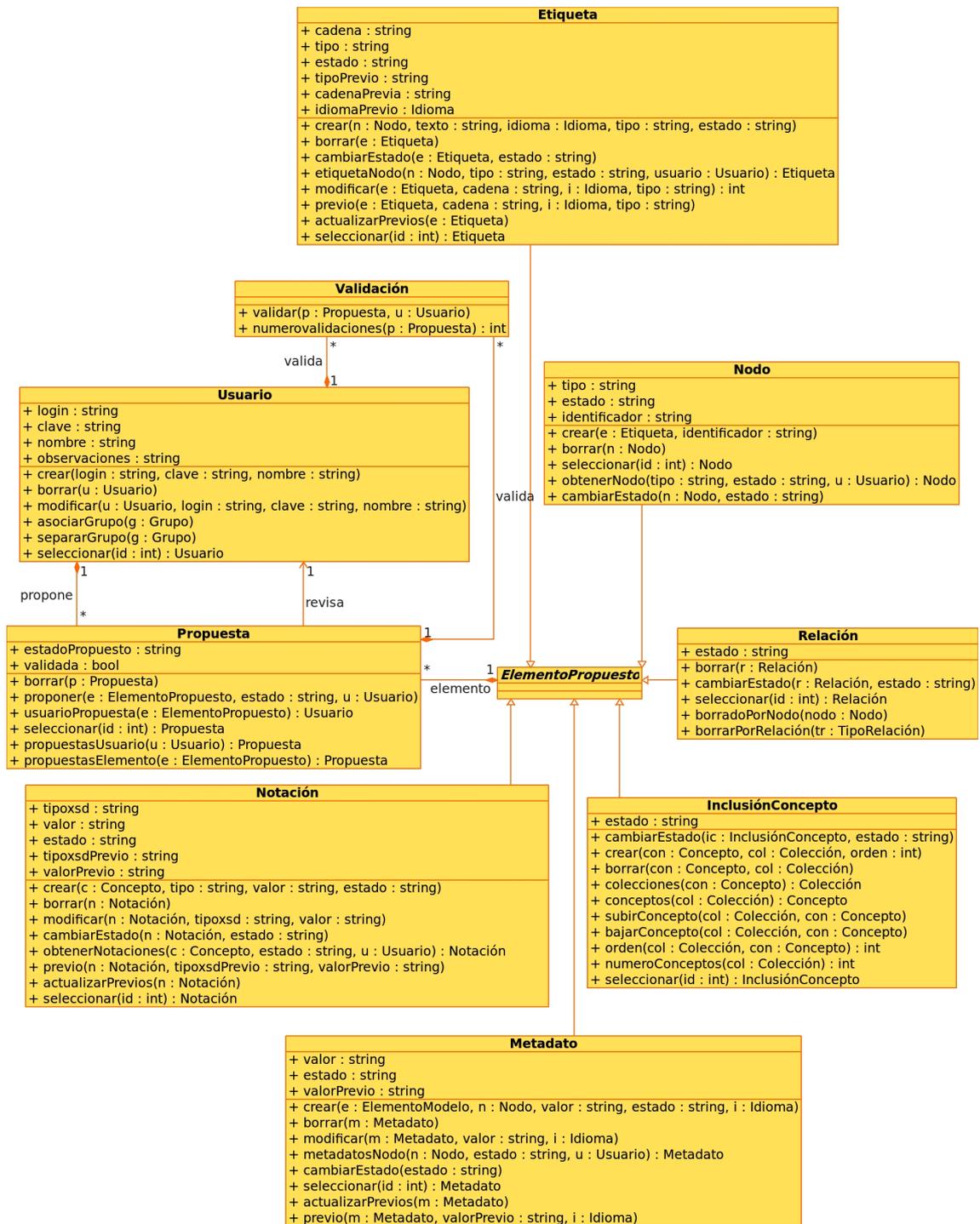


Ilustración 52: Diagrama de clases para los procesos de flujo de trabajo

4.4.2.9. CLA-08 Marcadores

Los marcadores propuestos en el modelo permiten almacenar referencias a conceptos o a recursos. Existen dos clases para representar los diferentes tipos de marcadores “*MarcadorRecursos*” y “*MarcadorConceptos*”. Esta última generaliza todas las clases de marcadores de conceptos para permitir distinguir diferentes funcionalidades (indización, mantenimiento y búsqueda). Cada usuario solo puede tener asociado un marcador de cada tipo.

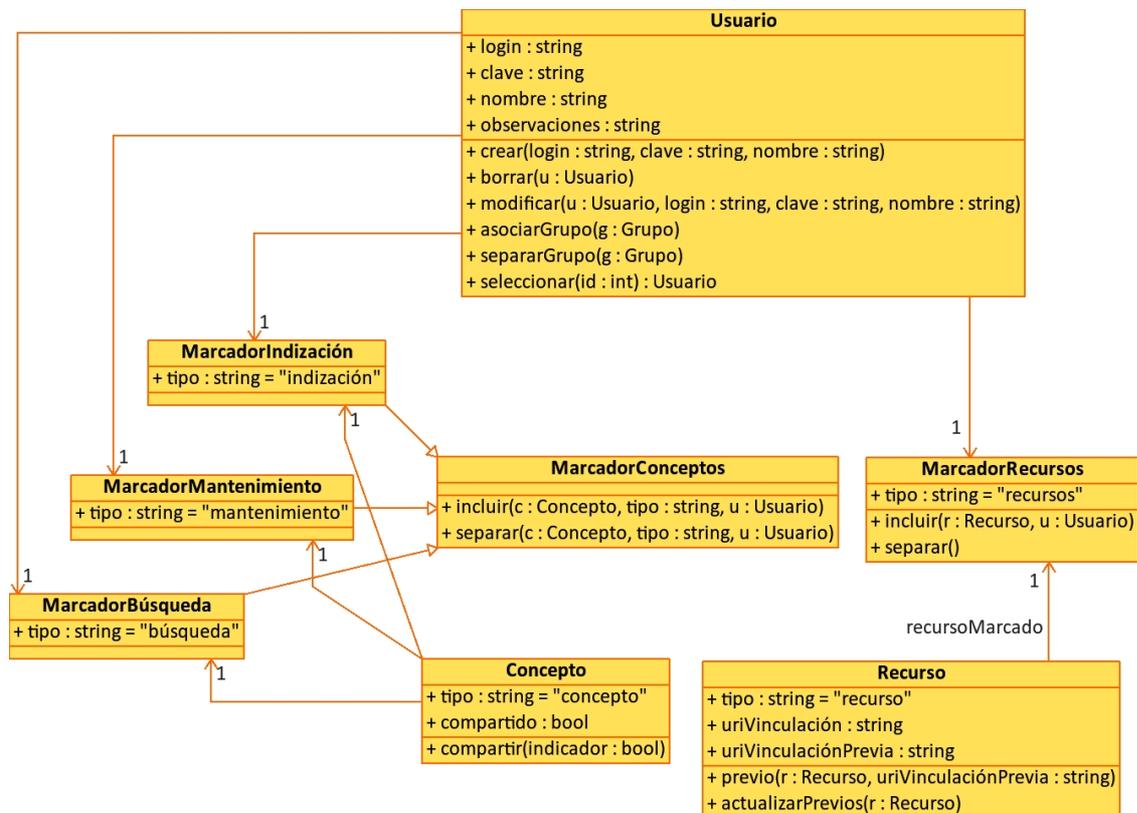


Ilustración 53: Diagrama de clases para los marcadores del sistema

4.4.3. Diagramas de secuencia

Hasta ahora se han identificado principalmente aspectos estáticos, si bien los diagramas de casos de uso pueden indicar algunas características de la dinámica general del modelo. Tal y como hemos indicado anteriormente el modelo conceptual propuesto constituye una base de trabajo para el desarrollo de una aplicación, entrando en aspectos de diseño e implementación en donde se abordan de lleno la totalidad de los aspectos dinámicos de la aplicación.

El siguiente paso sería la elaboración de los diagramas de secuencia, pensando en el desarrollo de una aplicación concreta. Si bien la elaboración de dichos diagramas no depende de la adopción de una determinada plataforma o entorno de desarrollo es cierto que su confección dependerá en gran medida de la orientación del sistema. No obstante, aunque el modelado de los aspectos asociados a los diagramas de secuencia queda fuera del alcance de este trabajo hemos incluido una descripción general del papel y componentes de los mismos, así como un ejemplo simplificado de un caso de uso

Un diagrama de secuencia tiene como objetivo la descripción de la dinámica del sistema. La interacción entre los objetos y actores se produce en forma de mensajes, por lo que se muestran asimismo el flujo de los datos intercambiados. A cada instancia de un objetos se asocia una línea de vida que indica el periodo o periodos de actividad. Los diagramas de secuencia en UML 2 introducen el concepto de marco de interacción para la representación de estructuras de control condicionales, alternativas, cíclicas o recursivas que amplían la dinámica lineal de los diagramas secuenciales de versiones anteriores de UML. Existen diversos tipos de mensajes entre objetos del sistema:

- Mensajes sincrónicos: el expedidor del mensaje espera a que el destinatario finalice el método invocado antes de continuar.
- Mensajes asincrónicos: el expedidor no espera a que el destinatario finalice el método invocado antes de continuar con la actividad. Es útil para modelar sistemas en donde los objetos pueden trabajar en paralelo.
- Mensajes de retorno: representan la respuesta producida por el destinatario una vez finalizado un evento activado por un mensaje sincrónico o asincrónico. No todas las activaciones de eventos producen un mensaje de retorno.

4. Propuesta de modelo conceptual de aplicación

- Mensajes reflexivos: modelan el envío de mensajes de un objeto a sí mismo.

A continuación se incluye el diagrama de secuencia correspondiente a la consulta de conceptos (CU-C05). Se ha decidido aportar una versión simplificada en la que no se representan los metadatos, colecciones y elementos de documentación del concepto, con la finalidad de hacer viable su inclusión en este documento.

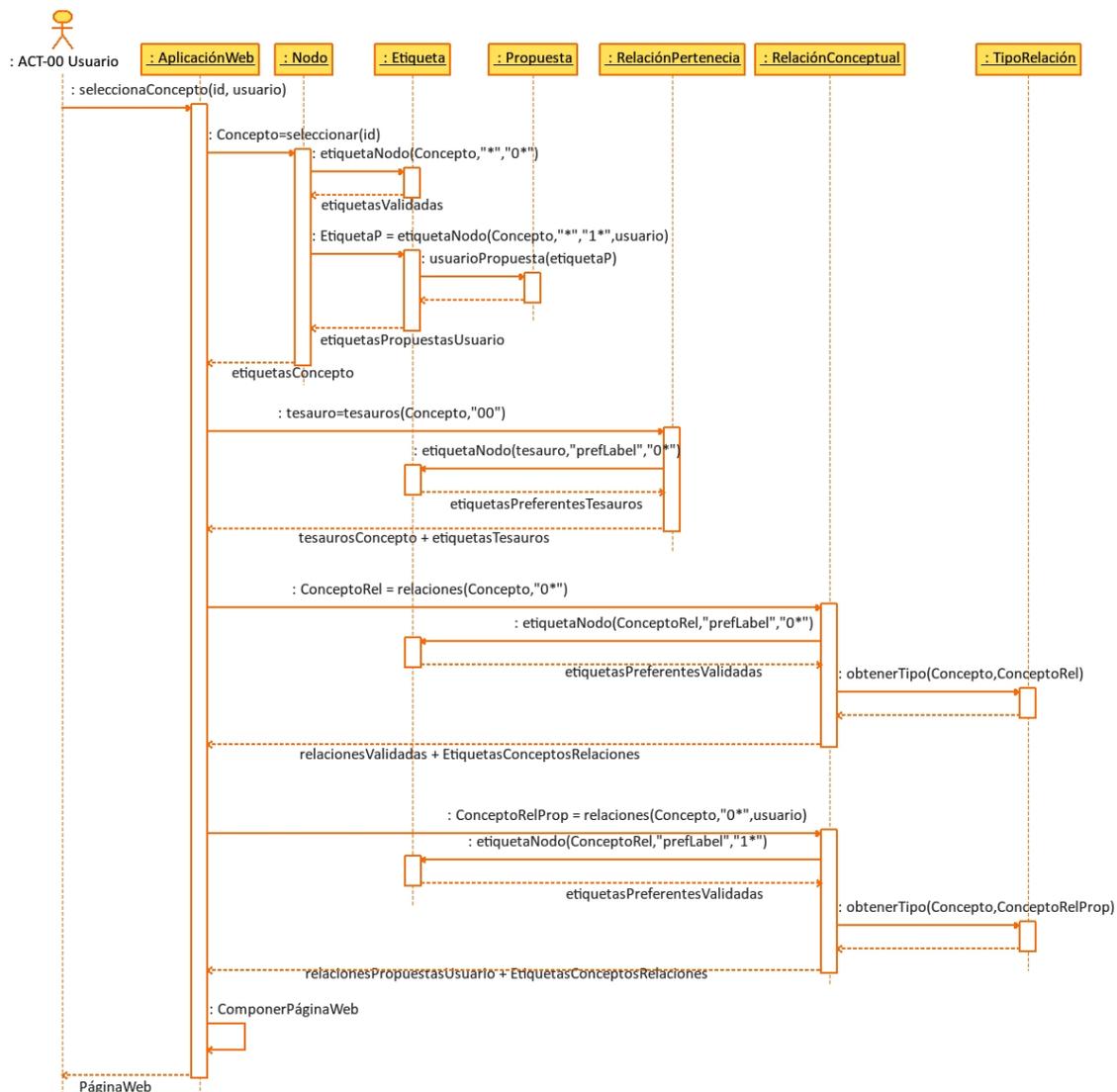


Ilustración 54: Diagrama de secuencia para el caso de uso CU-C03: Búsqueda de conceptos por etiquetas

En el ejemplo anterior puede observarse como se representa el actor “Usuario” y los diferentes objetos que intervienen en el diagrama de secuencia. En las líneas de

tiempo se insertan los periodos de actividad de los elementos junto con los mensajes intercambiados entre los mismos. Los mensajes se representan como flechas etiquetadas con el nombre del método activado que puede incluir o no parámetros. Cuando ha sido necesario se ha hecho uso de diversos tipos de marcos de interacción.

Se ha introducido un objeto denominado “Aplicación web” que no tiene su equivalente en el diagrama de clases. La finalidad de dicha inclusión es la representación del papel de la aplicación web como un elemento intermediario entre el usuario y los objetos del sistema.

4.5. Indicaciones para el desarrollo de la aplicación

Pese a que el diseño de una aplicación web concreta no es el objetivo inmediato de este trabajo si se ha considerado apuntar algunas consideraciones para su desarrollo en un futuro. Estas observaciones se basan en ciertos aspectos tecnológicos y conceptuales apuntados a lo largo de este capítulo y que pueden tener especial relevancia en el momento de la implementación del sistema.

El uso de Javascript o de cualquier tecnología de cliente destinada a incrementar el nivel de interacción del usuario con la interfaz deberá tener su consiguiente aumento en el grado de usabilidad de la aplicación. A este respecto resaltamos la conveniencia de utilizar aquellas tecnologías disponibles de forma común en la mayoría de clientes web, siempre y cuando no requieran el uso de plugins⁹⁰ para incrementar las funcionalidades de un cliente.

El uso conjunto de CSS, Javascript junto con AJAX y los lenguajes de programación del lado de servidor ofrecen suficientes herramientas para la creación de entornos altamente interactivos. No obstante, la aplicación también se diseñará pensando en su usabilidad para aquellas personas que no puedan disponer de este tipo de características en su navegador. Esto generalmente se hace mediante un uso adecuado de las hojas de estilo CSS y el desarrollo de código Javascript mediante técnicas no obstructivas⁹¹. De esta forma, la funcionalidad del sistema permanecerá inalterable

90 Un *plugin* o conector es un software destinado a extender las características del navegador por medio de contenidos diseñados para su consulta a través de este tipo de programas.

91 Las técnicas de programación mediante Javascript no obstructivo se basan en la creación de código Javascript que no utilice en su interior la definición de atributos visuales utilizando para ello una adecuada planificación de las clases CSS.

cuando se utilice a través de clientes que no soporten o tengan desactivada la características de ejecución de Javascript.

El uso de los servicios web planteados en el apartado 4.2.4 constituye una alternativa al desarrollo de una API⁹². El sistema ofrecerá una serie de servicios para la consulta de los tesauros y repositorios que devolverá los resultados en formato RDF para que puedan ser procesados por cualquier aplicación. Estos servicios podrían utilizarse a través de SOAP. Aunque en un principio los servicios web cubre los aspectos referentes a la interrogación de la información del sistema también puede plantearse, tal y como hace Pérez Agüera (2004 y 2007), la ampliación de las funcionalidades del sistema hacia operaciones de normalización terminológica y gestión de tesauros y repositorios, lo que permitiría un mayor grado de interoperabilidad entre sistemas.

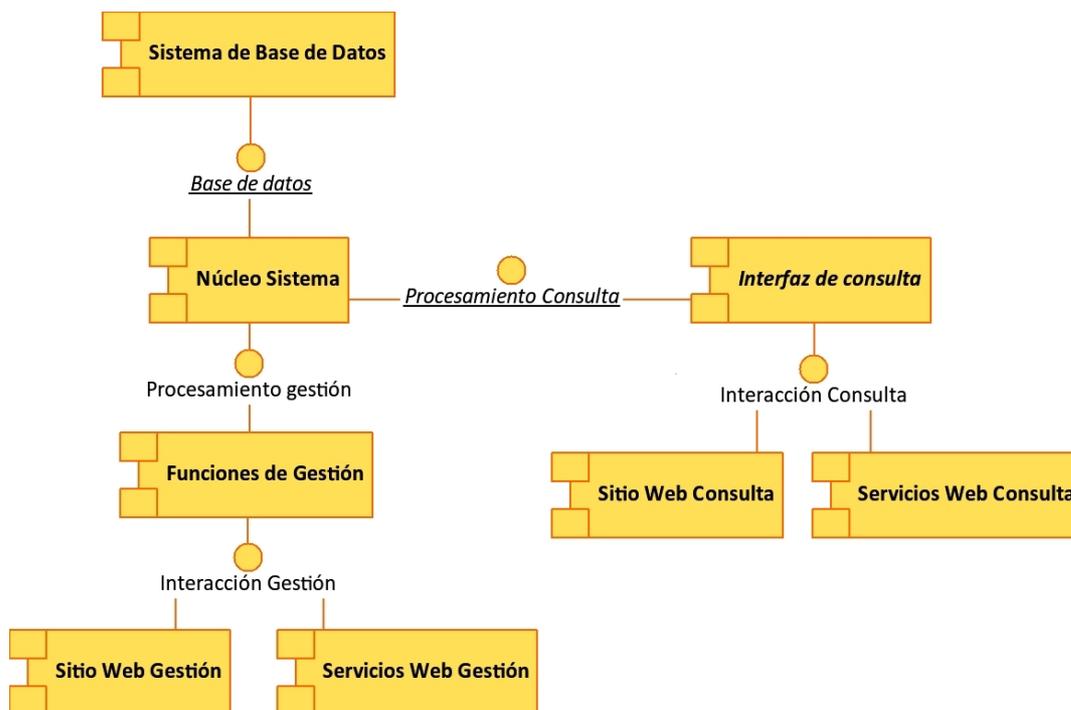


Ilustración 55: Diagrama UML de componentes del sistema

El sistema se estructuraría en un núcleo operativo que estaría directamente en contacto con la base de datos donde se almacenaría la información. El resto de componentes se clasificarían en aquellos destinados a la consulta de la base de datos y

92 API: *Application Programming Interface*. Es un conjunto de funciones y procedimientos que constituye una capa de abstracción y que permite utilizar un software determinado.

los dedicados a las operaciones de mantenimiento. Para ambas vertientes existirían componentes encargados de actuar de interfaz con usuarios o aplicaciones para procesar y gestionar los elementos de los tesauros y repositorios.

Otro aspecto de interés es el acercamiento a MDA⁹³ que proporciona UML. Esto también se plasma en la base de datos utilizada para el almacenamiento de la información gestionada por el sistema. A partir del diagrama de clases pueden implementarse diversas soluciones de diseño lo cual refuerza el carácter de independencia del modelo propuesto con respecto a la plataforma de desarrollo que se utilice.

La solución más inmediata y puesta en práctica actualmente es la conversión de los distintos diagramas de clases en un esquema relacional apto para la mayoría de sistemas de bases de datos actualmente existentes. Existen métodos para la transformación de los diagramas de clases UML propuestos a un esquema apto para cualquier sistema de bases de datos basado en el paradigma relacional (Shah y Slaughter, 2003).

Sin embargo es posible plantear una alternativa que suponga la adopción de RDF para representar estructura de objetos del sistema. Esta alternativa resulta de gran interés debido a la naturaleza de la información gestionada por el sistema: si se gestionan elementos descriptivos basados en el modelo RDF ¿por qué no utilizar RDF para estructurar toda la información del sistema? Esta idea surgió desde la aparición de RDF con el objetivo de almacenar declaraciones basadas en este modelo a través de esquemas relacionales (Melnik, 2000). Actualmente el interés de planteamiento es aun mayor teniendo en cuenta que existen soluciones orientadas al almacenamiento y gestión de información mediante RDF.

En efecto, existen sistemas de gestión de bases de datos que ofrecen la posibilidad de estructurar la información mediante RDF. Tal es el caso de *Virtuoso Universal Server*⁹⁴.

93 MDA: *Model-Driven Architecture*. Se trata de una metodología en el diseño de software propuesto por OGM que permite definir el modelo de la aplicación de forma independiente a la plataforma de desarrollo lo que facilitaría modificaciones en el diseño sin que influyera en la arquitectura del sistema y a la inversa.

94 Virtuoso Universal Server: Sistema híbrido de base de datos que combina RDF, XML, gestión de bases de datos relacionales e incluso bases de datos orientadas a objetos desarrollado por Openlink Software.

También existen marcos de desarrollo como Jena, Sesame o JRDF⁹⁵. El elemento común que ha permitido el desarrollo de estas entornos de desarrollo ha sido la aparición de SPARQL⁹⁶. Esta tecnología ofrece una herramienta para la construcción de consultas en un lenguaje con un alto grado de expresividad y adaptado a la consulta de información estructurada con RDF. Se trata de un mecanismo común que incrementa la interoperabilidad de aplicaciones destinadas a la Web Semántica.

Con las especificaciones para la representación de SKOS y para la serialización de Dublin Core se dispone de una base de trabajo para utilizar RDF de forma nativa en el desarrollo de una aplicación que implemente el modelo propuesto la gestión y uso de tesauros en Internet. Únicamente sería necesario desarrollar algunas pequeñas especificaciones RDF para la representación de los objetos asociados a la gestión de usuarios, grupos, permisos, repositorios y marcadores.

95 Se tratan de marcos de desarrollo que proporcionan una serie de APIs y librerías, junto con la infraestructura de almacenamiento de información estructurada mediante RDF. Permite el desarrollo de aplicaciones desarrolladas en Java o PHP (éste último lenguaje de programación únicamente en el caso de Sesame).

96 SPARQL: *SPARQL Protocol and RDF Query Language*. Se trata de una tecnología basada en un lenguaje para la consulta de contenidos almacenados mediante el modelo RDF y un protocolo para transmitir dichas consultas y los resultados obtenidos. Ha sido desarrollado por el Grupo de Trabajo de Acceso a Datos RDF (DAWG) del W3C. La primera recomendación definitiva fue presentada el 15 de enero de 2008.

5

Conclusiones

5. Conclusiones

"Aquella teoría que no encuentre una aplicación práctica en la vida, es una acrobacia del pensamiento." (Swami Vivekananda)

Resumen: Las conclusiones recogidas a continuación suponen una recapitulación de diversos aspectos analizados a lo largo de la presente tesis. Para su exposición se ha seguido el orden discursivo de este trabajo, partiendo de la consideración de la Web Semántica como una posible solución a la problemática de la recuperación de información en la Web, el planteamiento del modelo de tesoro conceptual propuesto por SKOS aplicable a este fin y su integración en un modelo conceptual para el desarrollo de una aplicación Web para la gestión y uso de tesauros en Internet.

Primera

El principal problema de los procesos de Recuperación de Información es la representación del contenido semántico de los recursos de información mediante estructuras que posibiliten su almacenamiento y comunicación, dotándolas además de un mecanismo común para la representación de las necesidades de información de los usuarios, esto posibilitaría obtener el grado de similitud entre la consulta realizada y los recursos de información del sistema.

Segunda

Los lenguajes documentales, los esquemas y mapas conceptuales, las herramientas para la desambiguación lingüística, los lematizadores o los etiquetadores morfológicos, etcétera son algunas de las herramientas que, por un lado, permiten organizar y describir los recursos de información a partir de su contenido, y por otro solventan los problemas de ambigüedad y diversidad del lenguaje natural permitiendo su uso en entornos automatizados.

Tercera

La complejidad de desarrollo e integración de las herramientas ya mencionadas, es subyacente al bajo grado de estructuración de los recursos con los que operan los sistemas de recuperación de información. En ocasiones estos sistemas trabajan con texto sin ningún tipo de marcado que permita definir formalmente aspectos semánticos. Ya, en el ámbito de la Web, lenguajes como HTML y XHTML no están diseñados para representar dichos aspectos, sino únicamente para su comunicación mediante una estructuración básica y la aplicación de estilos visuales. Esto obliga a trabajar casi únicamente en técnicas basadas en la frecuencia de aparición de expresiones, perdiéndose estructuras de organización conceptual que pueden ser de gran utilidad en los procesos de búsqueda y reutilización de la información.

Cuarta

A pesar del inconveniente de las técnicas señaladas, éstas han sido utilizadas para desarrollar variantes de modelos de recuperación de información que han tenido una amplia difusión. Ello se debe al carácter puramente automático del proceso de análisis de los documentos, que no requiere prácticamente ningún tipo de estructuración previa. Por tanto, estos modelos son válidos en sistemas en los que se controla y asegura la homogeneidad de ciertas características de los documentos entrantes, tales como la cobertura temática, nivel de detalle o tipo de lenguaje utilizado.

Quinta

La Web ha evolucionado desde una posición estática y homogénea hacia otra dinámica y diversa. Las causas de este hecho se encuentran en la difusión de la tecnología Web y el acceso cada vez mayor de los usuarios a Internet, tanto a contenidos como a tecnologías de desarrollo. Éstos ya no se limitan a la consulta de información sino que forman parte activa en la publicación de contenidos de temáticas, estructuras y formatos muy variados, todo ello de un modo sencillo y en ocasiones sin que sea necesaria una formación tecnológica.

Sexta

Los motores de búsqueda web utilizados hoy día no están preparados para trabajar en un entorno tan dinámico como el surgido con el desarrollo de servicios asociados a la Web 2.0. Esto se debe a la falta de control de la calidad de los contenidos, formatos, estructuras de los documentos y recursos disponibles en la Web, que provocan la obtención de resultados duplicados o de escasa utilidad en el proceso de búsqueda.

Séptima

Los problemas actuales de recuperación de información en la Web tienen su origen en la carencia o nula aplicación de mecanismos de representación del contenido. La Web Semántica tiene como objetivo la formalización del contenido de los recursos de manera que las personas y las máquinas puedan encontrar, intercambiar, integrar, reutilizar y operar la información de los mismos, utilizando para ello vocabularios de metadatos al tiempo que se separa estructura, contenido, relaciones y formato visual.

Octava

La arquitectura general de la Web Semántica se estructura en una capa de metadatos donde se describen los recursos de información, una capa de esquema en la que se describen las relaciones entre dichos recursos y una capa lógica donde se definen ontologías para la ejecución de inferencias. Por tanto, la Web Semántica permite un proceso de adaptación gradual de los contenidos informativos sin que sea necesario la representación de los mismos en forma de ontologías desde un primer momento.

Novena

Desde el punto de vista tecnológico, los recursos de información se localizan a través de direcciones identificadas mediante URIs. La sintaxis de la Web Semántica se define mediante el lenguaje XML lo que permite separar los datos y la definición de la estructura. El contexto de aplicación de los identificadores definidos con XML se realiza con los Namespaces. El modelo RDF/RDFS permite representar los atributos y relaciones de los recursos. Con OWL se expresan y representan relaciones y restricciones complejas. La verificación y la seguridad de los datos con los que se opera en la Web Semántica se certifica mediante la firma digital y métodos de encriptación.

Décima

El concepto de metadato no es una idea nueva, puesto que las tareas de análisis documental contemplan la descripción de aspectos formales de documentos. Lo novedoso resulta de su aplicación en la Web a través de un modelo formal con un conjunto de elementos descriptivos normalizados. La iniciativa Dublin Core ofrece un modelo de este tipo con un alto grado de normalización, a través de las normas ISO 15836-2003 y ANSI/NISO Z3950.85-2001. Constituye una parte esencial de la infraestructura de la Web Semántica, con un alto grado de implantación en sistemas de repositorios de recursos, pero con escasa aplicación en la descripción de recursos de información de índole general.

Undécima

Las ontologías permiten la representación del conocimiento de un dominio mediante conceptos, relaciones, funciones, instancias y axiomas. Poseen una doble vertiente: semántica, para formalizar, expresar y estructurar objetos y hechos; y pragmática, para definir las aplicaciones de una ontología de forma metódica. Estas, permiten el desarrollo y comunicación de estructuras complejas y su procesamiento tanto por personas como por máquinas. No obstante, la dificultad en su elaboración y mantenimiento es directamente proporcional al grado de abstracción y estructuración.

Duodécima

Metadatos y ontologías se complementan y constituyen el núcleo del procesamiento abstracto de los contenidos informativos de la Web Semántica, que contempla el desarrollo de nuevas herramientas descriptivas como paso intermedio entre metadatos y ontologías. En la práctica, los procesos simples de descripción de recursos de información de Internet se realizan mediante los elementos de Dublin Core, las descripciones más elaboradas y precisas, donde incluso es necesario un acercamiento al concepto de clase y relaciones entre recursos, se llevan a cabo con una herramienta de ámbito general como RDF. Las tareas de formalización complejas donde intervienen diversas operaciones de inferencia y procesamiento lógico se detallan con OWL.

Decimotercera

El incremento del nivel de estructuración y descriptivo de los metadatos es proporcional al grado de abstracción en la búsqueda y procesamiento de la información y su reutilización en procesos lógicos, con el consiguiente incremento de la interoperabilidad entre sistemas y usuarios. Desde este punto de vista la normalización de los metadatos constituyen un factor crítico en el desarrollo de la Web Semántica

Decimocuarta

Tanto la Web 2.0 como la Web Semántica pueden ser consideradas como estados o pasos intermedios y necesarios para un objetivo a más largo plazo: la Web Ubicua, que formaría parte esencial del núcleo operativo de los sistemas informáticos. En esta nueva orientación de la Web a los procesos de conectividad social se combinarían con los de formalización de la información y la representación del conocimiento. En estos entornos el concepto de *Ordenador Personal* desaparecerá y se producirá una deslocalización de aplicaciones e información.

Decimoquinta

Para alcanzar los objetivos de la Web Ubicua y, de un modo más inmediato, los de la Web Semántica es necesario reorientar los procesos de representación y recuperación de información. Aunque esta meta se presume posible a largo plazo de forma inmediata es posible utilizar herramientas que ayuden a describir y estructurar de manera formal y compartida el contenido semántico de los recursos de información para una comunicación más eficaz del conocimiento que el que ofrece la Web actual. Dicho objetivo puede llevarse a cabo con las propuestas de la Web Semántica basadas en Dublin Core, RDF/RDFS y OWL. Los tesauros y, de un modo más general los esquemas conceptuales, constituyen un modelo para la aplicación de estas tecnologías en el rediseño de los mecanismos de representación, organización y búsqueda de información.

Decimosexta

Los tesauros se han desarrollado y aplicado desde los años 50 en procesos de indexación de documentos y de recuperación de información. Desde entonces se ha producido una evolución de este concepto hasta su normalización durante la década de los 80, concibiéndose ahora como un vocabulario controlado y postcoordinado organizado en unidades léxicas (términos preferentes y no preferentes) que conforman estructuras a través de relaciones semánticas que denotan asociaciones entre los conceptos que representan los términos.

Decimoséptima

Las nuevas tecnologías de la información han supuesto un punto de inflexión en el desarrollo y aplicación de los tesauros. Los nuevos soportes y formatos digitales han propiciado un rápido y considerable incremento en el volumen de información gestionada en entornos informáticos. Los tesauros considerados como herramientas léxicas de carácter estático, muy aplicados en entornos controlados y con contenidos informativos homogéneos, resultan de escasa utilidad en sistemas con estas nuevas necesidades.

Decimoctava

Las tecnologías de la información también han intervenido en el desarrollo de modelos de datos e interfaces informáticas para la implantación, consulta y aplicación de tesauros. El hipertexto se ha demostrado como un método muy eficaz y amigable para la navegación entre los elementos del tesoro y la consulta de los recursos de información indizados. A lo anterior hay que añadir las múltiples aplicaciones informáticas que facilitan la creación y mantenimiento de tesauros.

Decimonovena

El concepto de tesoro es necesario redefinirlo para su aplicación en entornos altamente dinámicos, heterogéneos y en constante cambio, como la Web. Los esquemas conceptuales permiten la representación de tesauros léxicos al tiempo que ofrecen una herramienta más flexible con estructura multinivel, dirigida hacia la organización y representación conceptual, que además puede aplicarse en los procesos de indización, recuperación de información y expansión de consultas.

Vigésima

La aplicación de tesauros conceptuales en la Web permitiría el desarrollo de sistemas estructurados en tres niveles: el nivel conceptual, donde se representan los conceptos y relaciones del tesoro, el nivel navegacional, en el que se localizan los documentos y los hipervínculos, y el nivel estructural, que muestra la estructura interna de los documentos. Esta arquitectura permitirá delimitar claramente los procesos desarrollados en cada nivel, posibilitando el diseño de sistemas de información a través de herramientas modulares para la recuperación, navegación y consulta de la información.

Vigésimo primera

Los tesauros también pueden emplearse en sistemas de gestión de contenidos Web. La organización e indización de los contenidos a través de conceptos puede complementarse con el desarrollo de ontologías para la construcción automática de sistemas de navegación dinámicos.

Vigésimo segunda

Es necesario integrar los tesauros con otras herramientas. Los mapas conceptuales pueden permitir mejorar aspectos asociados a la representación gráfica y la personalización de los niveles conceptual, léxico (e incluso de ocurrencia) de los tesauros. Los Topic Maps pueden ser aplicados en la representación conceptual desde un punto de vista estructural adaptado a cada documento del sistema. Las ontologías pueden definir las interrelaciones a través de la definición y formalización de los mecanismos y características de los nexos de unión existentes en las interrelaciones entre tesauros, mapas conceptuales y topic maps, en un entorno que combine la recuperación de información con técnicas estadísticas y esquemas conceptuales.

Vigésimo tercera

La Web actual debe sufrir una serie de transformaciones y estados intermedios que la acerquen a los objetivos y funciones de la Web Semántica. El uso de esquemas conceptuales son un paso intermedio de gran valor, ya que supondrá el desarrollo de aplicaciones pragmáticas destinadas a la descripción de recursos de información a nivel conceptual.

Vigésimo cuarta

Para representar tesauros en el ámbito de la Web Semántica es necesario disponer de un marco de desarrollo en el que no sea necesario determinar previamente el tipo de entorno de aplicación. RDF propone un modelo general perfectamente aplicable para el desarrollo de especificaciones que pueden integrarse entre si y una sintaxis para su aplicación mediante RDF/XML. Esto resulta de gran interés para diseñar especificaciones para representar tesauros para su creación y uso de forma distribuida, en cualquier sistema y conjuntamente con especificaciones RDF/XML de otro tipo.

Vigésimo quinta

Desde la aparición de RDF comenzaron a desarrollarse especificaciones para la representación de tesauros basadas en dicho modelo. Mientras que en unos casos se continuaba con el modelo basado en términos de la visión tradicional de tesoro basados en términos (CERES, GEM, CALL Thesaurus Ontology), en otros se adoptaban visiones más genéricas destinadas a la creación de esquemas conceptuales (LIMBER, ILRT, KAON, ETB). Estas iniciativas constituyen el antecedente de SKOS, una aplicación de RDF para representar esquemas conceptuales de todo tipo, entre los que podemos encontrar a los tesauros.

Vigésimo sexta

SKOS propone un modelo en donde los conceptos se asocian con etiquetas preferentes y alternativas. A su vez los conceptos se relacionan entre sí mediante relaciones semánticas de diverso tipo, disponiendo de relaciones semánticas especiales de correspondencia para conceptos de diferentes tesauros. Entre las etiquetas pueden establecerse relaciones léxicas personalizadas. Por tanto es posible identificar tres niveles de estructuración de un tesoro: un nivel conceptual (que contiene los conceptos y relaciones semánticas) un nivel terminológico (con las etiquetas y relaciones léxicas) y un nivel de correspondencia terminológica (que conecta los niveles conceptual y terminológico).

Vigésimo séptima

SKOS contempla otros aspectos complementarios de gran interés como la creación de estructuras multilingües asignando identificadores de idioma a las etiquetas, la creación de estructuras de agrupación de conceptos denominadas colecciones y la posibilidad de utilizar una amplia tipología de elementos de documentación para anotar conceptos. Además, puede integrarse perfectamente con otras serializaciones RDF/XML como por ejemplo, la desarrollada a partir del modelo de Dublin Core.

Vigésimo octava

Los esquemas conceptuales podrían representarse directamente mediante ontologías, más concretamente con OWL. De hecho SKOS es una ontología OWL Full, pero su comprensión y aplicación directa resultan mucho más sencillas que hacerlo directamente con OWL ya que esta herramienta no proporciona un modelo de datos de fácil comprensión.

Vigésimo novena

SKOS puede ampliarse usando OWL de manera que es posible utilizar conjuntamente ambas soluciones para representar estructuras que contemplen la organización conceptual de un dominio de conocimiento y los aspectos de la semántica formal de dicho dominio, algo de lo que carecen los tesauros.

Trigésima

De esta forma, con SKOS los conceptos, esquemas conceptuales y colecciones son recursos RDF, identificables a través de una URI. Sobre dichos recursos es posible definir propiedades como son las etiquetas, las relaciones semánticas y la pertenencia a conceptos. En el caso de las relaciones léxicas es necesario declarar las etiquetas como recursos sobre las que se definen las formas literales y las relaciones como propiedades.

Trigésimo primera

SKOS no define ningún tipo de restricción o regla de integridad, más allá de las propiedades permitidas para cada elemento o la disyunción entre las relaciones *skos:related* y *skos:broaderTransitive*. Esto se debe al carácter general y flexible del modelo propuesto para la construcción de esquemas conceptuales. No obstante, consideramos que para la construcción de tesauros es necesario aplicar una serie de restricciones que garanticen su coherencia estructural a nivel léxico y semántico.

Trigésimo segunda

Se puede afirmar que una aplicación informática para la creación y mantenimiento de tesauros basados en el modelos de SKOS encuentra en la tecnología Web un marco ideal de desarrollo. A los beneficios de la Web como cliente/servidor universal hay que unir la existencia de herramientas de programación y almacenamiento de información potentes y escalables. Todo lo anterior se combina de forma integrada en un entorno de trabajo donde son claves la conectividad entre usuarios y los recursos de información en Internet. Una aplicación web, para la finalidad que estamos apuntando, es mucho más ventajosa que una clásica aplicación de escritorio.

Trigésimo tercera

Los sistemas de gestión de contenidos constituyen un paradigma aplicable en el desarrollo y estructura de las herramientas que gestionen algún tipo de información para la Web. De un modo más concreto, resulta de gran interés la organización de la dinámica de consulta y gestión de información con base a una estructura de perfiles de usuario y las posibilidades de gestión de contenidos mediante técnicas de flujo de trabajo.

Trigésimo cuarta

Cualquier sitio o aplicación Web, desde el punto de vista de la organización y estructuración, en lo referente a los contenidos informativos y la interfaz de usuario, ha de regirse por una serie de directrices fruto de una nueva visión que aúne las pautas y directrices propuestas por la Accesibilidad, la Usabilidad y la Arquitectura de la Información. Estas disciplinas se conjugan para dar lugar a resultados que contemplen las necesidades de información, organización, navegación e interacción, tanto para los usuarios que van a consultar los contenidos como para aquellos encargados de gestionarlos.

Trigésimo quinta

Se ha planteado un modelo conceptual de aplicación que coordina las funciones de gestión de tesauros a través de técnicas de trabajo en grupo y flujo de trabajo. Este, utiliza a su vez como modelo de representación de tesauros la propuesta de SKOS, teniendo en cuenta la necesidad de aplicar restricciones semánticas y léxicas para asegurar la coherencia del tesoro, junto con otras restricciones adicionales para garantizar la eficiencia operativa de la aplicación. Asimismo, se contempla la integración con otras especificaciones de metadatos, la ampliación de las funciones de la aplicación con las funciones de gestión e indización de repositorios de recursos de información, y la consulta de tesauros y recursos desde un sitio web público o por medio de servicios web basados en una arquitectura de tipo SOA.

Trigésimo sexta

Para la organización de un entorno de trabajo en grupo es necesario realizar una jerarquización de los permisos de gestión y consulta para los tesauros y los repositorios, definiendo una serie de roles que se asocian a cada grupo de usuarios. Dichos roles son los de administrador, gerente, editor, autor, lector y usuario anónimo. La dinámica de flujo de trabajo establece un proceso de validación por parte de los usuarios adscritos a grupos de editor, gerente o administrador para aquellos elementos de los tesauros y repositorios propuestos por los usuarios asociados al rol de autor. Los usuarios anónimos o los clientes de los servicios web del sistema únicamente han de tener permisos de consulta para aquellos tesauros públicos.

Trigésimo séptima

Las funciones de un sistema de las características descritas anteriormente deben organizarse en dos ejes que contemplen los procesos de gestión y consulta referidos tanto a los tesauros como a los repositorios. La organización y coordinación de estos procesos se supervisa y controla mediante las funciones de trabajo en grupo y las características de los procesos de propuesta, revisión y validación de las operaciones de flujo de trabajo.

Trigésimo octava

La interfaz de usuario de la aplicación es el elemento catalizador que permite la integración de las funciones del sistema a partir de procesos comunes, compartidos por múltiples escenarios de uso de la aplicación. Además, durante el desarrollo e implantación de la aplicación ha de tenerse en cuenta que la interfaz de usuario deberá contemplar, en cualquier momento, las pautas y directrices de Accesibilidad, Usabilidad y Arquitectura de la Información.

Trigésimo novena

Una aplicación web desarrollada a partir del modelo conceptual planteado, implementará mecanismos que permitirán explotar la estructura jerárquica de los tesauros en los procesos de búsqueda, así como identificar relaciones implícitas entre recursos de información a partir de la indización de los mismos con los conceptos de los tesauros.

Cuadragésima

UML ofrece un lenguaje de modelado adecuado para la formalización de la propuesta descrita en el análisis funcional, debido a su flexibilidad, las posibilidades que ofrece su uso en las fases de desarrollo e implantación de la aplicación y la existencia de perfiles orientados al diseño de aplicaciones web. Para la formalización del modelo conceptual con UML es necesario realizar al menos los diagramas de casos de uso para el planteamiento de los posibles escenarios del sistema y los diagramas de clases para la identificación de los objetos del mismo. La elaboración de los diagramas de secuencia para los casos de uso más significativos resultan de gran interés para una primera aproximación de la dinámica lógica del sistema. El modelo conceptual propuesto es válido para el diseño tanto de una aplicación web como de otro tipo.

Cuadragésimo primera

Los diagramas UML de casos de uso realizados muestran que es posible organizar la mayor parte de las funciones de gestión y consulta de los elementos del sistema, en torno a los casos de uso de consulta de los conceptos del tesoro y los recursos de información de los repositorios. Por tanto, los procesos de gestión de tesauros y repositorios pueden realizarse de forma simultánea a la consulta de los mismos.

Cuadragésimo segunda

La propia naturaleza de los servicios web hace necesaria una visión equilibrada entre el suministro de amplios volúmenes de información y el establecimiento de un número excesivo de conexiones para ejecutar los procesos de consulta. El análisis de los casos de uso correspondientes a esta funcionalidad indica que el escenario ideal agrupa determinadas operaciones de consulta similares a las realizadas por parte de los usuarios. De esta manera es posible ofrecer a los clientes de servicios web información agregada lo suficientemente significativa sin tener que realizar más de una petición al servidor.

Cuadragésimo tercera

A partir de los diagramas de clases elaborados es posible concluir el gran peso que tienen los procesos en los que intervienen las etiquetas y la gestión propuestas debido al carácter colaborativo del modelo propuesto. En la implantación de la aplicación es necesario poner especial énfasis en optimizar los procesos y estructuras de base de datos en los que intervengan ambos objetos.

Cuadragésimo cuarta

Es posible realizar la implantación de los diagramas de clases utilizando sistemas de bases de datos o marcos de desarrollo que operen directamente mediante esquemas RDF, lo que supone una gran ventaja frente al uso de esquemas relacionales ya que el desarrollo de la aplicación se centraría en la semántica de la información gestionada por el sistema.

Conclusión final

Los tesauros conceptuales constituyen una herramienta de gran valor en los procesos de organización y recuperación de información, adaptándose a los objetivos, requisitos y estructura en el entorno de la Web Semántica. El modelo SKOS, pese a encontrarse todavía en desarrollo, ofrece una solución suficientemente flexible para la representación de esquemas conceptuales, como los tesauros. Por tanto, es posible realizar un modelo conceptual de aplicación web que integre los modelos de los vocabularios RDF/XML de SKOS y de otras especificaciones de metadatos, como Dublin Core, en torno a los procesos de gestión y consulta de tesauros y repositorios de recursos de información e indización de los mismos, todo ello mediante una dinámica colaborativa de trabajo en grupo y flujo de trabajo.

Reflexiones finales

Este trabajo constituye el punto de partida para una serie de futuros desarrollos conducentes a la creación y puesta en marcha de un sistema, en forma de aplicación web, para la gestión y uso de tesauros basados en SKOS. Como paso previo será necesaria la realización de un modelado completo de los aspectos más relevantes de la dinámica de la aplicación, a través del diseño de los diagramas UML de actividades y estados, así como una interfaz de usuario que contemple los requisitos y los aspectos conceptuales modelados en el presente trabajo.

En principio, la implantación del sistema debe realizarse utilizando un marco de desarrollo que permita el uso de RDF en la definición y manipulación de las estructuras de datos requeridas. Una vez se disponga de la aplicación se estará en disposición de diseñar e implementar mecanismos que utilicen los tesauros gestionados por el sistema para la indización de forma asistida de recursos de información. Para ello, podrían utilizarse diferentes métodos de indización por semántica latente en distintos ámbitos desde el punto de vista del nivel descriptivo de los recursos, la amplitud de la cobertura temática o el ámbito organizativo de uso del sistema.

El objetivo final sería medir y determinar los umbrales efectivos respecto a la cantidad de elementos indizados con un tesoro conceptual en los que la indización manual podría dar paso a una indización asistida y ésta a una indización automática. Estos umbrales podrán ser ponderados en relación al nivel descriptivo de los metadatos de los recursos indizados.

6

Referencias Bibliográficas

6. Referencias bibliográficas

*"El recuerdo que deja un libro a veces es más importante que el libro en sí."
(Adolfo Bioy Casares)*

Resumen: Las referencias bibliográficas recogidas en este capítulo se han elaborado siguiendo el formato recomendado por la Asociación Americana de Psicológica (APA) utilizado ampliamente en el ámbito de las Ciencias Sociales. Se ha seleccionado este formato por su difusión, su claridad y su integración flexible con las citas realizadas en el texto.

- **Aguilar González, R. (2002).** *Monografía sobre motores de búsqueda.* Recuperado el 1 de septiembre de 2008, de <http://www.geocities.com/motoresdebusqueda/inicio.html>
- **Aitchinson, J. & Viet, J. (1984).** *Tesauro de la Unesco.* Paris: UNESCO.
- **Aitchinson, J. & Gilchrist, A. (1987).** *Thesaurus construction. A Practical manual.* 2ª edición. London: ASLIB.
- **Albert, C., Goes, P.B. & Gupta, A. (2004).** *GIST: A Model for Design and Management of Content and Interactivity of Customer-Centric Web Sites.* Recuperado el 8 de noviembre de 2008, de http://miscr.umcn.edu/workingpapers/fullpapers/2004/0402_011404.pdf
- **Álvarez Espinar, M. (2005).** *Nuevos lenguajes de modelado y estándares para la codificación de lenguajes documentales. OWL y Skos-Core.* Recuperado el 12 de octubre de 2008, de <http://www.w3c.es/Presentaciones/2005/1212-OWLSkosSEDIC-MA/>

- **Ambite Arnal, A., Díaz Gavilanes, R., Naya Altuna, R. & Ruíz Luna, L.R. (2006).** Gestores de contenido CMS y C-CMS (Groupware). Recuperado el 10 de noviembre de 2008, de http://cv.uoc.edu/~cv052_75_000_01_r06/pf_myc_0606/proyectos/uocms_pf.doc
- **ANSI/NISO (2001).** *Z39.85-2001: The Dublin Core Metadata Element Set*. Bethesda, Maryland, NISO Press. Disponible en línea, recuperado el 23 de octubre de 2008, de <http://download.www.techstreet.com/cgi-bin/pdf/free/335284/Z39.85-2001.pdf>
- **ANSI/NISO (2005).** *Z39.19-2005: Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies*. New York: National Information Standards Organization. Disponible en línea, recuperado el 15 de julio de 2008, de <http://download.www.techstreet.com/cgi-bin/pdf/free/455225/Z39-19-2005.pdf>
- **Arano, S. & Codina, L. (2004).** La estructura conceptual de los tesauros en el entorno digital: ¿nuevas esperanzas para viejos problemas? En *IX Jornades Catalanes d'Informació i Documentació* (pp. 41-58). Barcelona: Col·legi Oficial de Bibliotecaris-Documentalistes de Catalunya. Disponible en internet, recuperado el 1 de octubre de 2008, de <http://www.lluiscodina.com/ontotesauros.doc>
- **Arano, S. (2005).** Los tesauros y las ontologías en la Biblioteconomía y la Documentación. *Hipertext.net*, 3. Recuperado el 17 de septiembre de 2008, de <http://www.hipertext.net/web/pag260.htm>
- **Arlow, J., Neudstadt, I. (2006).** *UML 2*. Madrid, Anaya Multimedia.
- **Asensi Artiga, V. & Pastor Sánchez, J.A. (1998).** Propuesta de un modelo de interfaz genérica para sistemas de recuperación de información. *Scire*, 4(1), 71-88.
- **Baeza-Yates, R. & Ribeiro-Neto, B. (1999).** *Modern information retrieval*. New York: ACM Press Series/Addison Wesley .

- **Baeza-Yates, R., Rivera Loaiza, C. & Velasco Martín, J. (2004).** Arquitectura de la información y usabilidad en la web. *El profesional de la información*, 13(3), 168-178.
- **Barbin, L. (2007).** *Introducción a Ajax con PHP*. Madrid: Anaya Multimedia.
- **Berners-Lee, T., Hendler, J. & Lassila, O. (2001).** The Semantic Web. *Scientific American*, May 2001. Recuperado el 4 de febrero de 2008, de <http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2>
- **Berners-Lee, T. & Miller, E. (2002).** The Semantic Web lifts off. *ERCIM News*, 51, 9-11. Disponible en línea, recuperado el 17 de junio de 2008 http://www.ercim.org/publication/Ercim_News/enw51/berners-lee.html
- **Berners-Lee, T., Fielding, R. & Masinter, L. (2005).** *Uniform Resource Identifier (URI): Generic Syntax. Internet RFC 3986, January 2005*. Recuperado el 9 de Julio de 2008, de <http://www.ietf.org/rfc/rfc3986.txt>
- **Blair, D.C. (1990).** *Language and representation in information retrieval*. Amsterdam: Elsevier Science Publishers.
- **Boiko, B. (2002).** *Content Management Bible*. New York, Wiley.
- **Booch, G., Rumbaugh, J., Jacobson, I. (2006).** *El lenguaje unificado de modelado: guía del usuario*. Madrid, Pearson Education.
- **Brin, S. & Page, L. (1998).** The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30, 107-117. Recuperado el 21 de septiembre de 2008, de <http://infolab.stanford.edu/~backrub/google.html>
- **Brink, T., Gergle, D., Wood, S.D. (2002).** *Design websites that work: Usability for the Web*. San Francisco, Morgan-Kaufmann.

- **Brownson, H. (1957).** *Proceedings of the International Study Conference on Classification for Information Retrieval* (pp. 99-100). London: ASLIB, 1957.
- **Bryson, J. (1997).** *Managing Information Services: An Integrated Approach*. Aldershot, Gower.
- **Burbano Proaño, D.J. (2006).** *Análisis comparativo de bases de datos de código abierto vs código cerrado (determinación de índices de comparación)*. Recuperado el 5 de octubre de 2008, de <http://www.mysql-hispano.org/articulos/num43/analisis-comparativo.pdf>
- **Burkowski, F.J. (1992).** An algebra for hierarchically organized text-dominated databases. *Information Processing and Management: an International Journal*, 28(3), 333-348.
- **Chang, G., Healey, M.J., McHugh, J.A.M. & Wang, J.T.L. (2001).** *Mining the World Wide Web: an information search approach*. Norwell, Massachusetts: Kluwer Academic Publishers.
- **Chaumier, J. (1986).** *Análisis y lenguajes documentales: el tratamiento lingüístico de la información documental*. Barcelona: Mitre.
- **Chisalita, C., Puerta M.C. & Veer, G.C. van der (2003).** Designing complex systems in industrial reality: a study of the DUTCH approach. *UPGRADE-The European Online Magazine for the IT Professional*, 4(1), 46-52. Recuperado el 27 de octubre de, <http://www.upgrade-cepis.org/issues/2003/1/up4-1Chisalita.pdf>
- **Chowdhury, G.G. (1999).** *Introduction to modern information retrieval*. London: Library Association.
- **Cobo Romani, J.C. (2005).** *Organización de la información y su impacto en la usabilidad de las tecnologías interactivas*. Tesis doctoral, Universidad Autónoma de Barcelona.

- **Codina, L. (2003).** La web semántica: una visión crítica. *El profesional de la información*, 12(2), 149-152. Disponible en línea, recuperado el 15 de junio de 2008 de, <http://www.lluiscodina.com/articulos/ws.doc>
- **Corbin, J.R. (1991).** *The Art of Distributed Applications*. New York: Springer-Verlag.
- **Croft, W.B. (1987).** Approaches to intelligent information retrieval. *Information Proccesing & Management*, 23(4), 249-254.
- **Cross, P., Brickley, D. & Koch, T. (2000).** Conceptual relationships for encoding thesauri, classification systems and organised metadata collections and a proposal for encoding a core set of thesaurus relationships using an RDF Schema. Recuperado el 4 de mayo de 2008, de <http://www.desire.org/results/discovery/rdfthesschema.html>
- **Cuerda García, X., Minguillón Alfonso, J. (2004).** Introducción a los Sistemas de Gestión de Contenidos (CMS). *Mosaic*, 34. Recuperado el 7 de diciembre de 2008, de <http://mosaic.uoc.edu/articulos/cms1204.html>
- **Cueva Martín, M. de la (1999).** Acceso y utilización de tesauros en Internet. *Revista española de documentación científica*, 22(4), 531-540.
- **Curras, E. (2005).** *Ontología, taxonomía y tesauros: manual de construcción y usos*. Gijón: Trea.
- **Davis, M. (2006).** *Semantic Wave 2006: Executive Guide to the Business Value of Semantic Technologies. Semantic Interoperability Community of Practice (SICoP) White Paper Series Module 2*. Recuperado el 3 de octubre de 2008, de <http://web-services.gov/SICOPsemwave2006v1.0.doc>
- **Davis, M. (2008).** *Semantic Wave 2008 Report: Industry Roadmap to Web 3.0 & Multibillion Dollar Market Opportunities (Executive Summary)*. Recuperado el 18 de octubre de 2008, de http://www.isoco.com/pdf/Semantic_Wave_2008-Executive_summary.pdf

- **DCMI (2008a).** *DCMI Metadata Terms. DCMI Recommendation.* Recuperado el 15 de junio de 2008, de <http://dublincore.org/documents/dcmi-terms/>
- **DCMI (2008b).** *Dublin Core Metadata Element Set, Version 1.1. DCMI Recommendation.* Recuperado el 15 de junio de 2008, de <http://dublincore.org/documents/dces/>
- **DCMI (2008c).** *DCMI Abstract Model. DCMI Recommendation.* Recuperado el 15 de junio de 2008, de <http://dublincore.org/documents/abstract-model/>
- **Decker, S., Harmelen, F van, Broekstra, J., Erdmann, M., Fensel, D., Horrocks, I., et al. (2000).** *The semantic web - on the respective roles of XML and RDF.* Recuperado el 10 de junio de 2008, de <http://www.ontoknowledge.org/oil/download/IEEE00.pdf>
- **Delgado Domínguez, A. (1998).** *Mecanismos de recuperación de Información en la WWW.* Memoria de investigación no publicada para la obtención de la Suficiencia Investigadora, Universitat de les Illes Balears, Palma de Mallorca. Disponible en línea, recuperado el 2 de febrero de 2008, de <http://servidorti.uib.es/adelaida/tice/modul6/memfin.pdf>
- **Diaz Ortuño, P.M. (2003).** Problemática y tendencias en la Arquitectura de Metadatos Web. *Anales de Documentación*, 6, 35-58. Disponible en línea, recuperado el 15 de junio de 2008, <http://www.um.es/ojs/index.php/analesdoc/article/viewFile/2051/2041>
- **Ding, Y. & Foo, S. (2002).** Ontology research and development. Part 1-a review of ontology generation. *Journal of Information Science*, 2002, 28(2), 123-136.
- **Ding, Y. & Foo, S. (2002b).** Ontology research and development. Part 2-a review of ontology map-ping and evolving. *Journal of Information Science*, 28(5), 375-388.
- **Drevauwer, L., Heyde, Fien van der (2005).** *UML 2: iniciación, ejemplos y ejercicios corregidos.* Barcelona, ENI.

- **Eckel, B. (2007).** *Piensa en Java*. Madrid: Pearson Education.
- **Engel, G. (1997).** A Personal Portable Information Interface: A Model for Meeting User Needs in the Networked Information Environment. *Cataloging & Classification*, 24(2), 141-161.
- **Euzenat, J. (2002).** A Few Words about the Semantic Web and its Development in the ERCIM Institutes. *ERCIM News*, 51, 7-8. Recuperado el 21 de diciembre de 2007, de http://www.ercim.org/publication/Ercim_News/enw51/intro.html
- **Feather, J. & Sturges, P. (1997).** *International Encyclopedia of Information & Library Science*. London: Rotledge.
- **García Jiménez, A. (2004).** Instrumentos de representación del conocimiento: Tesauros versus Ontologías. *Anales de Documentación*, 7, 79-95. Disponible en línea, recuperado el 15 de junio de 2008, de <http://www.um.es/ojs/index.php/analesdoc/article/viewFile/1691/1741>
- **García Gutiérrez, A.L. (1984).** *Lingüística Documental. Aplicación a la documentación de la comunicación social*. Madrid: Mitre.
- **García Marco, F.J. (1994).** Hypertexto y lenguajes documentales: Retos y sinergias. En *IV Jornadas Españolas de Documentación Automatizada, Documat 94*, 6, 7 y 8 de octubre de 1994 (pp.417-426). Gijón: Universidad de Oviedo.
- **García Marco, F. J. (1995a).** Interfaces amigables para la recuperación de la información bibliográfica. *Scire* 1(1), 127-148.
- **García Marco, F. J. (1995b).** Hacia un modelo de intervención en los procesos de transmisión del conocimiento. *Scire* 1(2), 105-138.
- **García Torres, A., Pareja Lora, A. & Pradana López, D. (2008).** Reutilización de tesauros: el documentalista frente al reto de la web semántica. *El profesional de la información*, 17(1), 8-21.

- **García Sánchez, F. (2007).** *Sistema basado en tecnologías del conocimiento para entornos de servicios web semánticos*. Tesis Doctoral, Universidad de Murcia.
- **Garret, J.J. (2002).** *The elements of user experience. User-centered design for the web*. Indianapolis, New Riders.
- **Gil Leiva, I. (1999).** *La automatización de la indización de documentos*. Gijón: Trea.
- **Gil Urdiciain, B. (1998).** Orígenes y evolución de los tesauros en España. *Revista General de Información y Documentación*, 8(1), 63-110.
- **Gilchrist, A. (1971).** *The thesaurus in retrieval*. London: ASLIB.
- **Gilchrist, A. (2003).** Taxonomies and information architecture. *Scire*, 9(1), 37-46. Disponible en línea, recuperado el 23 de septiembre de 2008, de <http://ibersid.eu/ojs/index.php/scire/article/view/1454/1432>
- **Gilliland-Swetland, A. (2003).** Metadata - Where Are We Going? En G.E. Gorman (ed.) *International yearbook of library and information management 2003-2004: Metadata applications and management* (pp. 17-33). London: Facet Publishing.
- **González Lorca, J. (2006).** *Sistemas workflow : funcionamiento y metodología de implantación*. Gijón: Trea.
- **Grabs, T. & Shek, H.J. (2002).** Generating Vector Spaces On-the-fly for Flexible XML Retrieval. En *Proceedings of the XML and Information Retrieval Workshop - 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, August 2002. Recuperado el 24 de agosto de 2008, de <http://www.cs.huji.ac.il/course/2002/sdbi/Papers/ir-xml/xmlirws.pdf>
- **Grahan, S. (2002).** *Building web service with Java: making sense of XML, SOAP, WSDL and UDDI*. Indianapolis: Sams.

- **Granollers i Saltiveri, T. (2004).** *MPlu+a. Una metodología que integra la Ingeniería del Software, la Interacción Persona-Ordenador y la Accesibilidad en el contexto de equipos de desarrollo multidisciplinares.* Tesis doctora, Universidad de Lérida. Disponible en línea, recuperado el 14 de noviembre de 2008, de http://griho2.udl.es/publicacions/2004/Tesis_Toni/TesiToniGranollers.pdf
- **Grossman, D.A.; Frieder, O. (1998).** *Information retrieval: algorithms and heuristics.* Boston: Kluwer Academia Publishers.
- **Gruber, T.R. (1993).** A translating approach to portable ontology specifications. *Knowledge Acquisition*, 5, 199-220. Disponible en línea, recuperado el 3 de febrero de 2008, de http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html
- **Halpin, H. & Thompson, H.S. (2006).** One document to bind them: combining XML, web services, and the semantic web. En *Proceedings of the 15th International Conference on World Wide Web*, Edinburgh, Scotland, May 23 - 26, 2006 (pp. 679-686). New York: ACM Press. Disponible en línea, recuperado el 21 de diciembre de 2007, de <http://www2006.org/programme/files/pdf/5060.pdf>
- **Hendler, J., Berners-Lee, T. & Miller, E. (2002).** Integrating Applications on the Semantic Web. *Journal of the Institute of Electrical Engineers of Japan*, 122(10), 676-680. Disponible en línea, recuperado el 15 de junio de 2008, de <http://www.w3.org/2002/07/swint.html>
- **Hu W-c., Yining, C., Schmalz, M.S. Ritter, G.X. (2001).** An overview of World Wide Web search technologies. En *Proceedings of the 5th World Multi-Conference on Systemics Cybernetics and Informatics, (SCI 2001)*, Orlando, Florida (pp. 356-361). Disponible en línea, recuperado el 11 de junio de 2008, de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.28.9602&rep=rep1&type=pdf>
- **ISO (1985).** *ISO 5964:1986 Guidelines for the establishment and development of multilingual thesauri.* Geneve, International Organization for Standardization.

- **ISO (1986).** *ISO 2788:1986 Guidelines for the establishment and development of monolingual thesauri.* Geneve, International Organization for Standardization.
- **ISO (1998a).** *ISO-9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 11: Guidance on usability.* Geneve, International Organization for Standardization.
- **ISO (1998b).** *ISO-9241-12: Ergonomic requirements for office work with visual display terminals (VDTs) -- Part 12: Presentation of information.* Geneve, International Organization for Standardization.
- **ISO (1999).** *ISO 13407: Human-centred design processes for interactive systems.* Geneve, International Organization for Standardization. **ISO (2000).** *ISO/TR 18529: Ergonomics -- Ergonomics of human-system interaction -- Human-centred lifecycle process descriptions.* Geneve, International Organization for Standardization.
- **ISO (2002a).** *ISO/TR 16982: Ergonomics of human-system interaction -- Usability methods supporting human-centred design.* Geneve, International Organization for Standardization.
- **ISO (2002b).** *ISO/TR 16982: Ergonomics of human-system interaction -- Usability methods supporting human-centred design.* Geneve, International Organization for Standardization.
- **ISO (2002c).** *ISO/TR 16982: Ergonomics of human-system interaction -- Usability methods supporting human-centred design.* Geneve, International Organization for Standardization.
- **ISO (2003).** *ISO 15836: Information and documentation - The Dublin Core Metadata Set.* Geneve, International Organization for Standardization.
- **ISO/IEC (2001).** *ISO/IEC 9126-1: Software engineering -- Product quality -- Part 1: Quality model.* Geneve, International Organization for Standardization.

- **Jabba Molinares, D., Alcocer Olaciregui, A. & Rojas Morales, C. (2004).** Análisis comparativo de las herramientas de programación Web: PHP, ASP y JSP, bajo los sistemas operativos Linux y Windows. *Ingeniería y desarrollo: revista de la División de Ingeniería de la Universidad del Norte*, 16, 104-115. Disponible en línea, recuperado el 2 de octubre de 2008, de http://ciruelo.uninorte.edu.co/pdf/ingenieria_desarrollo/16/analisis_comparativo_de_las_herramientas_de_programacion.pdf
- **Jones, S. (1993).** A Thesaurus data model for an intelligent retrieval system. *Journal of Information Science*, 19(3), 167-178.
- **Jurafsky, D. & Martin, J.H. (2000).** *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. New Jersey: Prentice-Hall.
- **Kalervo, J., Kristensen, J., Niemi, T., Sormunen, E. & Keskustalo, H. (1996).** A Deductive Data Model for Query Expansion. En *Proceedings of the 19th Annual International ACM–SIGIR Conference on Research and Development in Information Retrieval* (pp. 235-243). New York: ACM Press. Disponible en línea, recuperado el 15 de agosto de 2008, de http://www.info.uta.fi/tutkimus/fire/archive/ExpansionTool_Sigir96.pdf
- **Kobayashi, M. & Takeda, K. (2000).** Information retrieval on the web. *ACM Computing Surveys*, 32(2), 144-173.
- **Korfhage, R.R. (1997).** *Information Retrieval and Storage*. New York: Wiley Computer Publisher.
- **Korotkiy, M. & Top, J.L. (2004).** From Relational Data to RDFS Models. En *International Conference on Web Engineering. Volume 3140 of Lecture Notes in Computer Science* (pp. 430-434). Berlin, Heidelberg: Springer. Disponible en línea, recuperado el 6 de septiembre de 2008, de [http://www.cs.vu.nl/~maksym/pap/FDR2\(short\).pdf](http://www.cs.vu.nl/~maksym/pap/FDR2(short).pdf)
- **Kroiß, C. & Koch, N. (2008).** *UWE Metamodel and Profile: User Guide and Reference*. München: Ludwig-Maximilians-Universität. Recuperado el 12 de diciembre de 2008, de <http://www.pst.ifi.lmu.de/projekte/uwe/download/UWE-Metamodel-Reference.pdf>

- **de Laborda, C.P.; Conrad, S. (2005).** Relational OWL - A Data and Schema Representation Format Based on OWL. En S. Hartman & M. Stumppner (eds.) *Second Asia-Pacific Conference on Conceptual Modelling (APCCM2005), volume 43 of CRPIT* (pp. 89-96). Newcastle, Australia: ACS. Disponible en línea, recuperado el 6 de septiembre de 2008, de <http://crpit.com/confpapers/CRPITV43deLaborda.pdf>
- **Laguens García, J.L. (2006).** Tesoros y lenguajes controlados en Internet. *Anales de Documentación*, 9, 105-121.
- **Lamarca Lapuente, M.J. (2006).** Hipertexto: El nuevo concepto de documento en la cultura de la imagen. Tesis Doctoral, Universidad Complutense de Madrid. Disponible en línea, recuperado el 20 de abril de 2008, de <http://www.hipertexto.info/>
- **Lancaster, F.W. (1992).** *Vocabulary control for information retrieval*. (2nd. ed.) Arlington, Vancouver, Canada: Information Resources Press.
- **Lanker, L. van (2005).** *CSS en DHTML: Javascript aplicado a hojas de estilo*. Barcelona: Ediciones Eni, cop.
- **Laureilhe, M.T. (1977).** *Le thesaurus: son rôle, sa structure, son élaboration*. Paris: ENSB.
- **Lawrence, S. & Giles, C.L. (1998).** Searching the World Wide Web. *Science*, 280, 98-100.
- **Lefebvre, A. (1997).** *Intranet cliente-servidor universal*. Barcelona: Gestión 2000.
- **Leloup, C. (1998).** *Motores de búsqueda e indexación: entornos cliente servidor, Internet e Intranet*. Barcelona: Gestión 2000.
- **López Alonso, M.A. (2000).** Las estructuras conceptuales de representación del Conocimiento en Internet. *Scire*, 6(1), 107-123. Disponible en línea, recuperado el 28 de diciembre de <http://ibersid.eu/ojs/index.php/scire/article/view/1127>

- **López-Huertas Pérez, M.J. (1997).** Thesaurus structure design: a conceptual approach for improved interaction. *Journal of Documentation* 53(2), 139-177.
- **López-Huertas Pérez, M.J. (1999).** Potencialidad evolutiva del tesauro: hacia una base de conocimiento experto. En *La Representación y la Organización del Conocimiento en sus distintas perspectivas: su influencia en la Recuperación de la Información, Actas del IV Congreso ISKO-España EOCONSID'99*, 22-24 de abril de 1999 (pp. 133-140). Granada: ISKO, Facultad de Biblioteconomía y Documentación.
- **Maniez, J. (1993).** *Los Lenguajes documentales y de clasificación: concepción, construcción y utilización en los sistemas documentales*. Madrid: Pirámide, Fundación Germán Sánchez Ruipérez.
- **Marchinioni, G. (1992).** Interfaces for End-User Information Seeking. *Journal of the American Society for Information Science*. 43(2), 156-163.
- **Marchionini, G. (2004).** From Information Retrieval to Information Interaction. En *26th Annual European Conference on Information Retrieval* (pp. 1-11). Disponible en línea, recuperado el 24 de mayo de 2008, de http://www.ils.unc.edu/~march/ecir04/ecir_slides.htm
- **Martínez Méndez, F.J. (2004).** *Recuperación de Información: Modelos, Sistemas y Evaluación*. Murcia: Kiosko.
- **Mathes, A. (2004).** *Folksonomies - Cooperative Classification and Communication Through Shared Metadata*. *Computer Mediated Communication – LIS590CMC*, Graduate School of Library and Information Science, University of Illinois Urbana-Champaign, December 2004. Recuperado el 3 de agosto de 2008 de <http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html>
- **Mayfield, J. (1995).** *Two-level Hypertext Models as an Underpinning for AHSs*. Recuperado el 3 de septiembre de 2008, de <http://www.wis.win.tue.nl/ah94/Mayfield.html>

- **Mayfield J. (1997).** Two-Levels Models of Hypertext. En C. Nicholas & J. Mayfield (eds.) *Intelligent Hypertext: Advanced Techniques for the World Wide Web. Lecture Notes in Computer Science No. 1326* (pp. 90-108). New York: Springer-Verlag.
- **Mayhew, D.J. (1999).** *The Usability Engineering Lifecycle: A practitioner's Handbook for User Interface Design*. San Francisco, Morgan Kaufman.
- **Meadow, C.T. (1992).** *Text Information retrieval Systems*. San Diego: Academic Press.
- **Melnik, S. (2000).** Storing RDF in a relational database. Recuperado el 10 de agosto de 2008 de, <http://infolab.stanford.edu/~melnik/rdf/db.html>
- **Méndez Rodríguez, E.M. (1999).** RDF: un modelo de metadatos flexible para las bibliotecas digitales del próximo milenio. En *Jornades Catalanes de Documentació* (pp. 487-498). Barcelona: Col·legi Oficial de Bibliotecaris-Documentalistes de Catalunya. Disponible en línea, recuperado el 16 de junio de 2008, de <http://www.cobdc.org/jornades/7JCD/1.pdf>
- **Méndez Rodríguez, E.M. (2000).** Metadatos y Tesauros: aplicación de XML/RDF a los sistemas de organización del conocimiento en Intranets. En *La Gestión del Conocimiento: retos y soluciones de los profesionales de la información, Actas de las VII Jornadas Españolas de Documentación, FESABID 2000*, Bilbao, 19-20-21 octubre de 2000 (pp. 211-219). Bilbao: Universidad del País Vasco. Disponible en línea, recuperado el 16 de junio de 2008, de <http://www.bib.uc3m.es/~mendez/publicaciones/fesabid00/fesabid002.pdf>
- **Méndez Rodríguez, E.M. (2002).** *Metadatos y recuperación de información: estándares, problemas y aplicabilidad en bibliotecas digitales*. Gijón: Trea.
- **Miller, K. & Matthews, B. (2001).** Having the Right Connections: the LIMBER Project. *Journal of Digital Information*, 1(8), Article 37. Recuperado el 12 de mayo de 2008, de <http://jodi.tamu.edu/Articles/v01/i08/Miller/>

- **Miller, U. (1997).** Thesaurus construction: problems and their roots. *Information Processing & Management*, 33(4), 481-493.
- **Moreiro González J.A, Sánchez Cuadrado S, Morato J. (2003).** Panorámica y tendencias en topic maps. *Hipertext.net*, 1. Recuperado el 5 de septiembre de 2008, de <http://www.hipertext.net/web/pag229.htm>
- **Moreiro González, J.A., et al. (2004).** Mapas conceptuales, topic maps y tesauros. En *Actas de las I Jornadas Españolas de Tratamiento y Recuperación de Información*, febrero 2004. Recuperado el 5 de septiembre de 2008, de <http://www.fiv.upv.es/jotri/Ponencias/Mapas.pdf>
- **Moreiro González, J.A. (2006).** La representación y recuperación de los contenidos digitales: de los tesauros conceptuales a las folksonomías. En Tramullas Saz, J. (coord.) *Tendencias en documentación digital*, (pp. 81-108). Gijón, Trea.
- **Moya Martínez, G. & Gil Leiva, I. (2001).** Evaluación de softwares de gestión de tesauros. *Ciencias de la Información*, 32(3), 3-23.
- **Navarro, G. & Baeza-Yates, R. (1995).** A language for queries on structure and contents of textual databases. En *18th ACM Conference on Research and Development in Information Retrieval (SIGIR'95)*, Seattle, WA, USA, July 1995 (pp. 93-101). Disponible en línea, recuperado el 22 de agosto de 2008, de <ftp://sunsite.dcc.uchile.cl/pub/users/rbaeza/papers/TOIS97.ps.gz>
- **Nielsen, J. (1993).** *Usability Engineering*. Boston, Academic Press International.
- **Nielsen, J. (2002a).** *Usabilidad: diseño de sitios web*. Madrid, Prentice Hall.
- **Nielsen, J. (2002b).** *Top Ten Guidelines for Homepage Usability*. Recuperado el 8 de octubre de 2008, de <http://www.useit.com/alertbox/20020512.html>
- **Norlin, D. A. et al. (1992).** Interface design and development: The human factor. *Library Hi-Tech*, 70(3), 7-24.

- **ODGM (2000).** *The Object Data Standard: ODMG 3.0.* Editores: R.G.G. Cattell y D.K. Barry. San Francisco, Morgan Kaufmann.
- **Pastor Sánchez, J.A. & Saorín Pérez, T. (1993).** Un interfaz hipertexto para un sistema de gestión de tesauros. *Scire*, 2(1), 51-62. Disponible en línea, recuperado el 2 de enero de 2009, de <http://ibersid.eu/ojs/index.php/scire/article/view/1049>
- **Pastor Sánchez, J.A. & Saorín Pérez, T. (1995a).** El hipertexto documental como solución a la crisis conceptual del hipertexto. El reto de los documentos cooperativos en redes. *Cuadernos de Documentación Multimedia*, 4, 41-56. Disponible en línea, recuperado el 1 de septiembre de 2008, de <http://www.ucm.es/info/multidoc/multidoc/revista/cuadern4/hiperdoc.htm>
- **Pastor Sánchez, J.A. & Saorín Pérez, T. (1995b).** Exposición conceptual y aplicación de un modelo para la Organización del Conocimiento en Internet: El Hipertexto Documental. En *Organización del Conocimiento en Sistemas de Información y Documentación: Actas del II Encuentro de ISKO-España*, Getafe, 16 y 17 de noviembre de 1995 (pp. 247-269). Zaragoza: Universidad.
- **Pastor Sánchez, J.A. (1997).** Limitaciones del WWW en el ámbito de la información documental. *Information World en Español*, 6(4), 11-13. Disponible en línea, recuperado el 22 de febrero de 2008, de http://www.elprofesionaldelainformacion.com/contenidos/1997/abril/limitaciones_del_www_en_el_mbito_de_la_informacin_documental.html
- **Pastor Sánchez, J.A. & Saorín Pérez, T. (1998).** La escritura hipermedia. *Cuadernos de Documentación Hipermedia*, 6-7, 221-238. Disponible en línea, recuperado el 1 de septiembre de 2008, de <http://www.ucm.es/info/multidoc/multidoc/revista/cuad6-7/saorin.htm>

- **Pastor Sánchez, J.A. & Martínez Méndez, F.J. (2003).** ISGAT: Gestión colaborativa de tesauros en internet. *SCIRE*, 9(2), 85-98. Disponible en línea, recuperado el 19 de septiembre de 2008, de <http://ibersid.eu/ojs/index.php/scire/article/view/1468/1446>
- **Pérez Agüera, J.R. (2004).** Automatización de tesauros y su utilización en la web semántica. *BiD : textos universitaris de biblioteconomia i documentació*, 13. Recuperado el 1 de septiembre de 2008, de <http://www.ub.es/bid/13perez2.htm>
- **Pérez Agüera, J.R. (2007).** *Automatización de tesauros documentales para la Web Semántica*. Tesis Doctoral, Universidad Complutense.
- **Pérez-Carballo, J. & Strzalkowski, T. (2000).** Natural language information retrieval: progress report. *Information Processing and Management*, 36, 155-178.
- **Pérez-Montoro, M. (2006).** Gestión del Conocimiento, Gestión documental y Gestión de contenidos. En Tramullas Saz, J. (coord.) *Tendencias en documentación digital*, (pp. 109-133). Gijón, Trea.
- **Pilone, D, Pitman, N. (2005).** *UML 2.0 in a nutshell*. Sebastopol, California, O'Reilly.
- **Pizzato, L.A. (2003).** *Estrutura Multitesauro para Recuperação de Informações*. Tesis de Master, Pontifícia Universidade Católica do Rio Grande do Sul.
- **Rada, R., Zeb, A., You, G.-N., Michailidis, A. & Mhashi, M. (1991).** Collaborative hypertext and the MUCH system. *Journal of Information Science*, 17, 191-196.
- **Rao, R. (1996).** Cuando la Información Salta a la vista. *Mundo Científico*, 168, 452-459.
- **Rijsbergen, C.J. (1999).** *Information Retrieval*. Glasgow, University. Disponible en línea, recuperado el 12 de enero de 2008, de <http://www.dcs.gla.ac.uk/~iain/keith/>

- **Roberts, N. (1984).** Historical studies in documentation. The pre-history of the information retrieval thesaurus. *Journal of documentation*, 40(4), 271-285.
- **Robertson, S.E. & Sparck Jones, K.(1976).** Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27, 129-146.
- **Robertson, S.E. (1977).** The probability ranking principle in IR. *Journal of Documentation*, 33(4), 294-304.
- **Rodríguez Muñoz, J.V., Martínez Méndez, F.J. & Díaz Ortuño, P. (1990).** Los modelos de datos como alternativa en la construcción de tesauros. En *Actas de las III Jornadas Españolas de Documentación Automatizada*. Palma de Mallorca: FESABID.
- **Rodríguez Muñoz, J. V. (1992).** *Construcción del esquema conceptual del tesoro mediante un modelo de datos*. Tesis Doctoral, Universidad de Murcia.
- **Rosenfeld, L. & Morville, P. (2002).** *Information Architecture for the world wide web: designing large-scale web sites*. Sebastopol (California), O'Reilly.
- **Rosson, M.B. & Carroll, J.M. (2002).** *Usability Engineering: scenario-based development of Human-Computer Interaction*. San Francisco, Morgan Kaufmann.
- **Salton, G. & McGill, M.J. (1983).** *Introduction to Modern Information Retrieval*. New York: Mc Graw-Hill Computer Series.
- **Savy, J. & Picard, J. (2001).** Retrieval effectiveness on the Web. *Information Processing and Management*, 37(4), 543-569.
- **Schlögl, C. (2005).** Information and knowledge management: dimensions and approaches. *Information Research* 10 (4). Recuperado el 23 de octubre de 2008, de <http://informationr.net/ir/10-4/paper235.html>

- **Senso, J.A. & de la Rosa Piñero, A. (2003).** El concepto de metadato. Algo más que descripción de recursos electrónicos. *Ciência da Informação*, 32(2), 95-106. Disponible en línea, recuperado el 4 de agosto de 2008, de <http://revista.ibict.br/index.php/ciinf/article/view/126/107>
- **Shah, D., Slaughter, S. (2003).** Transforming UML class diagrams into relational data models. En Favre, L. *UML and the unified process*. Hershey, Pennsylvania, IGI Publishing.
- **Shiri, A.A. & Revie, C. (2000).** Thesauri on the Web: current developments and trends. *Online Information Review*, 24(4), 273-279. Disponible en línea, recuperado el 10 de octubre de 2008, de <http://dlist.sir.arizona.edu/archive/00000163/01/thesauri.pdf>
- **Shiri, A.A., Revie, C. & Chowdhury, G.(2002).** Thesaurus-assisted search term selection and query expansion: A review of user-centred studies. *Knowledge organization*, 29(1), 1-19.
- **Sklar, D. (2005).** *Introducción a PHP 5*. Madrid : Anaya Multimedia.
- **Slype, G. van (1991).** *Los lenguajes de indización, concepción, construcción y utilización en los sistemas documentales*. Pirámide: Madrid, 1991.
- **Spivack, N. (2005).** *Making Sense of the Semantic Web*. Recuperado el 3 de octubre de 2008, de http://novaspivack.typepad.com/nova_spivacks_weblog/files/nova_spivack_semantic_web_talk.ppt
- **Tosete Herranz, F. (2007).** *Santiago Calatrava, Arquitectura y la Arquitectura de la Información*. Recuperado el 12 de noviembre de 2008, de <http://tentandole.blogspot.com/2007/05/10/>
- **Tramullas Saz, J. (1997).** *Introducción a la Documática*. Zaragoza: Kronos. Disponible en línea, recuperado el 17 de agosto de 2008, de <http://tramullas.com/documatica/3-1.html>

- **Tramullas Saz, J. (2003).** Documentos y servicios digitales: de la usabilidad al diseño centrado en el usuario. *El profesional de la información*, 12(2), 107-110.
- **Tramullas Saz, J., Garrido Picazo, P. (2006).** Sistemas de Gestión de Contenidos. En Tramullas Saz, J. (coord.) *Tendencias en documentación digital*, (pp. 135-161). Gijón, Trea.
- **Tudhope, D., Harith, A. & Jones, C. (2001).** Augmenting Thesaurus Relationships: Possibilities for Retrieval. *Journal of Digital Information*, 1(8), Article 41. Recuperado el 5 de septiembre de 2008, de <http://jodi.tamu.edu/Articles/v01/i08/Tudhope/>
- **W3C (2001).** *Review of RDF Thesaurus Work: A review and discussion of RDF schemas for thesauri. Public DRAFT for discussion.* Editores: B. Matthews y A. Miles. Recuperado el 5 de mayo de 2008, de <http://www.w3.org/2001/sw/Europe/reports/thes/8.2/>
- **W3C (2003).** *OWL Web Ontology Language XML Presentation Syntax. W3C Note 11 June 2003.* Autores: M. Hori, J. Euzenat y P.F. Patel-Schneider. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/owl-xmlsyntax/>
- **W3C (2004a1).** *OWL Web Ontology Language Guide. W3C Recommendation 10 Feb 2004.* Editores: M.K. Smith, C.Welty y D.L. McGuinness. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/owl-guide/>
- **W3C (2004a2).** *OWL Web Ontology Language Overview. W3C Recommendation 10 Feb 2004.* Editores: D.L. McGuinness y F. van Harmelen. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/owl-features/>
- **W3C (2004a3).** *OWL Web Ontology Language Reference. W3C Recommendation 10 Feb 2004.* Editores: M. Dean y G. Schreiber. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/owl-ref/>

- **W3C (2004a4).** *OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation 10 Feb 2004.* Editores: P.F. Patel-Schneider, P. Hayes e I. Horrocks. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/owl-semantics/>
- **W3C (2004a5).** *OWL Web Ontology Language Test Cases. W3C Recommendation 10 Feb 2004.* Editores: J.J. Carroll, J. De Roo. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/owl-test/>
- **W3C (2004a6).** *OWL Web Ontology Language Use Cases and Requirements. W3C Recommendation 10 Feb 2004.* Editor: J. Heflin. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/webont-req/>
- **W3C (2004b1).** *RDF Primer. W3C Recommendation 10 February 2004.* Editores: F. Manola y E. Miller. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/rdf-primer/>
- **W3C (2004b2).** *Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation 10 February 2004.* Editores: G. Klyne y J.J. Carroll. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/rdf-concepts/>
- **W3C (2004b3).** *RDF/XML Syntax Specification (Revised). W3C Recommendation 10 February 2004.* Editor: D. Beckett. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/rdf-syntax-grammar/>
- **W3C (2004b4).** *RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation 10 February 2004.* Editores: D. Brickley y R.V. Guha. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/rdf-schema/>
- **W3C (2004b5).** *RDF Semantics. W3C Recommendation 10 February 2004.* Editor: P. Hayes. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/rdf-mt/>

- **W3C (2004b6).** *RDF Test Cases. W3C Recommendation 10 February 2004.*
Editores: J. Grant, D. Beckett. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/rdf-testcases/>
- **W3C (2004c1).** *XML Schema Part 0: Primer Second Edition. W3C Recommendation 28 October 2004.* Editores: D.C. Fallside y P. Walmsley. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/xmlschema-0/>
- **W3C (2004c2).** *XML Schema Part 2: Datatypes Second Edition. W3C Recommendation 28 October 2004.* Editores: P.V. Biron y A. Malhotra. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/xmlschema-2/>
- **W3C (2004d).** *Web Services Architecture.* Editores: D. Booth, H. Haas, F. McCabe, E. Newcomer, M. Champion, C. Ferris, y D. Orchard. Recuperado el 19 de septiembre de 2008, de <http://www.w3.org/TR/ws-arch>
- **W3C (2005a1).** *SKOS Core Guide. 2nd W3C Public Working Draft 2 November 2005.* Editores: A. Miles y D. Brickley. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/2005/WD-swbp-skos-core-guide-20051102/>
- **W3C (2005a2).** *SKOS Core Vocabulary Specification. 2nd W3C Public Working Draft 2 November 2005.* Editores: A. Miles y D. Brickley. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/2005/WD-swbp-skos-core-spec-20051102/>
- **W3C (2005a3).** *Quick Guide to Publishing a Thesaurus on the Semantic Web. W3C Working Draft 17 May 2005.* Editor: A. Miles. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/2005/WD-swbp-thesaurus-pubguide-20050517/>

- **W3C (2006a1).** *Extensible Markup Language (XML) 1.0 (Fourth Edition)*. W3C Recommendation 16 August 2006, edited in place 29 September 2006. Editores: T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler y F. Yergeau. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/2006/REC-xml-20060816/>
- **W3C (2006a2).** *Namespaces in XML 1.0 (Second Edition)*. W3C Recommendation 16 August 2006. Editores: T. Bray, D. Hollander, A. Layman y R. Tobin. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/REC-xml-names/>
- **W3C (2007).** *SKOS Use Cases and Requirements*. W3C Working Draft 16 May 2007. Editores: A. Isaac, J. Phipps y D. Rubin. Recuperado el 15 de junio de 2008, de <http://www.w3.org/TR/2007/WD-skos-ucr-20070516/>
- **W3C (2008a1).** *SKOS Reference*. W3C Working Draft 29 August 2008. Editores: A. Miles y S. Bechhofer. Recuperado el 1 de septiembre de 2008, de <http://www.w3.org/TR/2008/WD-skos-reference-20080829/>
- **W3C (2008a2).** *SKOS Primer*. W3C Working Draft 29 August 2008. Editores: A. Isaac y E. Summers. Recuperado el 1 de septiembre de 2008, de <http://www.w3.org/TR/2008/WD-skos-primer-20080829/>
- **W3C (2008a3).** *SKOS Simple Knowledge Organization System eXtension for Labels (XL) RDF Schema*. 20th August 2008 "Last Call" Edition. Editores: A. Miles y S. Bechhofer. Recuperado el 29 de septiembre de 2008, de <http://www.w3.org/TR/2008/WD-skos-reference-20080829/skos-xl.html>
- **W3C (2008b1).** *Web Content Accessibility Guidelines (WCAG) 2.0*. W3C Proposed Recommendation 03 November 2008. Editores B. Caldwell, M. Cooper, Loretta Guarino y Gregg Vanderheiden. Recuperado el 11 de noviembre de 2008, de <http://www.w3.org/TR/WCAG20/>
- **W3C (2008c).** *SPARQL Query Language for RDF*. W3C Recommendation 15 January 2008. Editores E. Prud'hommeaux y A. Seaborne. Recuperado el 3 de enero de 2009, de <http://www.w3.org/TR/rdf-sparql-query/>

- **Welling, L. & Thompson, L. (2005).** *Desarrollo web con PHP y MySQL*. Madrid: Anaya Multimedia.
- **Wersig, G. (1971).** *Das Krankenhaus-Informationssystem (KIS) : Überlegungen Zu Strukturen Und Realisierungsmöglichkeiten Integrierter Krankenhaus-Informationssysteme*. München-Pullach: Verlag Dokumentation.
- **Zakas, N.C. (2006).** *Javascript para desarrolladores Web*. Madrid: Anaya Multimedia.
- **Zang, J. & Dimitrof, A. (2005).** The impact of metadata implementation on webpage visibility in search engine results (Part II). *Information Processing and Management*, 41, 691-715.

Esta tesis ha sido elaborada en su integridad con software libre. El sistema operativo sobre el que se han utilizado las diferentes aplicaciones ha sido la distribución de Linux Ubuntu (versión 8.10). Los diferentes documentos e ilustraciones se han confeccionado mediante el paquete ofimático OpenOffice (versión 3.0). Para el retoque de ilustraciones se ha utilizado Gimp (versión 2.6). Los diagramas de casos de uso y clases se han desarrollado con Umbrello (versión 2.1.3.)

