



Uso de wikis semánticas

Araceli Oleaga Fons

2n cicle d'Enginyeria en Informàtica

Felipe Geva Urbano

Entrega final
11 Junio 2013



Esta obra está sujeta a una licencia de [Reconocimiento-NoComercial-CompartirIgual 3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Uso de wikis semánticas
Nombre del autor:	Araceli Oleaga Fons
Nombre del consultor:	Felipe Geva Urbano
Fecha de entrega (mm/aaaa):	06/2013
Área del trabajo final:	XML y Web Semántica
Titulación:	2º ciclo de Ingeniería Informática
Resumen del trabajo (máximo 250 palabras):	
<p>Este trabajo está centrado en el uso de wikis semánticas y más concretamente en la creación de una ontología y su posterior inclusión en la wiki semántica elegida.</p> <p>En primer lugar se ofrece la justificación del proyecto, los objetivos establecidos y el plan de proyecto establecido.</p> <p>A continuación, para empezar a asentar conocimientos, se hace una introducción a la web semántica, su estructura, componentes, ventajas e inconvenientes.</p> <p>Continuando y adentrándonos más en el corazón del proyecto, se explica lo que es una ontología y se realiza una investigación sobre los lenguajes necesarios para desarrollar una, como son XML, RDF y OWL, de los cuales se ofrece una descripción básica y con algún ejemplo para su comprensión.</p> <p>Después, se avanza con la explicación del concepto de las wikis semánticas y se concreta más información sobre Semantic MediaWiki que será la wiki elegida para realizar el caso práctico.</p> <p>Por último, existe un apartado que describe el proceso seguido para la aplicación del caso práctico donde se ofrece una descripción del método seguido para crear una ontología sobre series de televisión, la instalación de Semantic Mediawiki y la inclusión de la ontología en la wiki.</p> <p>Para finalizar se ha creado un glosario con los términos más importantes que aparecen en la memoria.</p>	
Abstract (in English, 250 words or less):	
This work focuses on the use of semantic wikis and specifically in the	

creation of an ontology and its subsequent inclusion in the chosen semantic wiki.

First, it provides the justification for the project, the objectives and the project plan set.

Then, to start settling knowledge, it provides an introduction to the semantic web, its structure, components, advantages and disadvantages.

Continuing and moving ahead in the heart of the project, it explains what is an ontology and shows the research on the languages needed to develop, such as XML, RDF and OWL, which provides a basic overview and some examples for its understanding.

Then, it proceeds with the explanation of the concept of semantic wikis and more specific information about Semantic MediaWiki that will be chosen for the case study.

Finally, there is a section that describes the process for the implementation of the case study which is a description of the method used to create a television series ontology, Semantic MediaWiki installation and inclusion of the ontology in the wiki.

At the end, there is a glossary of key terms that appear along the way.

Palabras clave (entre 4 y 8):

XML, RDF, OWL, Ontología, Web semántica, Wiki semántica

Índice

1. Introducción.....	3
1.1 Contexto y justificación del Proyecto.....	3
1.2 Objetivos del Proyecto.....	4
1.3 Enfoque del Proyecto.....	4
1.4 Planificación del proyecto.....	6
1.4.1 Diagrama de Gantt.....	8
1.5 Productos obtenidos.....	9
1.6 Breve descripción de los otros capítulos de la memoria.....	10
2. Web Semántica.....	11
2.1 Introducción.....	11
2.2 ¿Qué es?.....	11
2.3 Componentes.....	11
2.4 Estructura.....	12
2.5 Ventajas.....	13
2.6 Inconvenientes y desventajas.....	14
3. Ontologías.....	14
3.1 Qué son.....	14
3.2 Lenguajes de representación.....	16
3.2.1 XML.....	16
3.2.2 RDF.....	20
3.2.3 OWL.....	25
3.3 Herramientas de creación, edición y gestión.....	28
3.3.1 WORDNET.....	28
3.3.2 PROTÉGÉ.....	31
3.3.3 SPARQL.....	31
4. Wiki Semántica.....	33
4.1 Qué es.....	33
4.2 Semantic MediaWiki.....	34
4.2.1 Definición.....	34
4.2.2 Características.....	34
4.2.3 Funcionalidades.....	35
4.2.4 Cómo almacena la información.....	36
4.2.5 Cómo se añade información semántica.....	36
4.2.6 Cómo importar ontologías.....	36
5. Aplicación práctica.....	37
5.1 Diseño de una ontología.....	37
5.1.2 De qué trata.....	37
5.1.3 Pasos que se han seguido.....	37
5.1.5 Estructura jerárquica de clases y propiedades.....	41
5.2 Aplicación sobre wiki semántica.....	43
5.3 Problemas encontrados.....	59
5.3.1. Carga del archivo XML/RDF.....	59
5.3.2 Error en el código del Endpoint del SPARQL.....	60
5.3.3 CSS Wiki.....	61
6. Conclusiones.....	62
7. Glosario.....	63
8. Anexos.....	65
9. Bibliografía.....	66

Índice de Ilustraciones

Ilustración 1: Diagrama de Gantt - Entrega Pac 2.....	8
---	---

Ilustración 2: Diagrama de Gantt - Entrega Pac 3.....	9
Ilustración 3: Diagrama de Gantt - Entrega final.....	9
Ilustración 4: Web Semántica - Estructura web semántica.....	13
Ilustración 5: RDF - Grafo RDF.....	21
Ilustración 6: RDF - Ejemplo completo RDF.....	25
Ilustración 7: WORDNET - Ejemplo Wordnet.....	30
Ilustración 8: SPARQL - Ejemplo SPARQL.....	32
Ilustración 9: Aplicación práctica - Wordnet, TV Show y Coordinates.....	38
Ilustración 10: Aplicación práctica - Wordnet, TV Show e Hypernoms.....	39
Ilustración 11: Aplicación práctica - Wordnet, TV Show y Hyponyms.....	39
Ilustración 12: Aplicación práctica - Jerarquía de clases Series TV.....	41
Ilustración 13: Aplicación práctica - Propiedades de Series TV.....	42
Ilustración 14: Aplicación práctica - Propiedades de datos de Series TV.....	43
Ilustración 15: Wiki Semántica - Página principal después de instalación.....	44
Ilustración 16: Wiki Semántica - Páginas especiales para SMW después de instalación.....	44
Ilustración 17: Wiki Semántica - Páginas especiales para RDFIO después de instalación.....	46
Ilustración 18: Wiki Semántica - Importación RDF (Inicio) Ontología Series TV.....	46
Ilustración 19: Wiki Semántica - Importación RDF (final) Ontología Series TV.....	47
Ilustración 20: Wiki Semántica - Listado categorías Series TV MediaWiki.....	47
Ilustración 21: Wiki Semántica - Página categoría Capítulo Series TV.....	48
Ilustración 22: Wiki Semántica - Listado de propiedades MediaWiki.....	48
Ilustración 23: Wiki Semántica - Propiedad CapítuloEsDeLaSerie de SeriesTV MediaWiki.....	49
Ilustración 24: Wiki Semántica - Ejemplo opción buscar por atributo en SMW.....	49
Ilustración 25: Wiki Semántica - Ejemplo opción explorar artículos en SMW(1).....	50
Ilustración 26: Wiki Semántica - Ejemplo opción explorar artículos en SMW(2).....	50
Ilustración 27: Wiki Semántica - Ejemplo opción explorar artículos en SMW(3).....	51
Ilustración 28: Wiki Semántica - Ejemplo opción búsqueda semántica en SMW(1).....	51
Ilustración 29: Wiki Semántica - Ejemplo opción explorar artículos en SMW(2).....	52
Ilustración 30: Wiki Semántica - Ejemplo opción explorar artículos en SMW(3).....	52
Ilustración 31: Wiki Semántica - Ejemplo página nueva con datos Series TV.....	53
Ilustración 32: Wiki Semántica - Página principal final.....	54
Ilustración 33: Wiki Semántica - Ejemplo resultados semánticos 1.....	55
Ilustración 34: Wiki Semántica - Ejemplo resultados semánticos 2.....	56
Ilustración 35: Wiki Semántica - Ejemplo resultados semánticos 3.....	57
Ilustración 36: Wiki Semántica - Ejemplo resultados semánticos 4.....	58
Ilustración 37: Wiki Semántica - Ejemplo uso SPARQL para Series TV.....	59
Ilustración 38: Wiki Semántica - Error SPARQL.....	60

Índice de tablas

Tabla 1: XML - Ejemplo documento XML.....	18
Tabla 2: XML - Ejemplo DTD.....	20
Tabla 3: RDF - Ejemplo tipos de datos.....	22
Tabla 4: RDF - Ejemplo literales.....	23
Tabla 5: OWL - Construcciones en OWL Lite.....	26
Tabla 6: OWL - Construcciones en OWL DL.....	27
Tabla 7: WORDNET - Relaciones semánticas.....	29
Tabla 8: Wiki Semántica - Ejemplo código búsqueda semántica.....	53
Tabla 9: Wiki Semántica - Ejemplo código resultados semánticos 2.....	56
Tabla 10: Wiki Semántica - Ejemplo código resultados semánticos 3.....	57
Tabla 11: Wiki Semántica - Ejemplo código resultados semánticos 4.....	58

1. Introducción

1.1 Contexto y justificación del Proyecto

El campo de la Web Semántica es un campo relativamente nuevo que está teniendo un gran impacto tanto en la práctica como en la investigación. La filosofía de la Web Semántica es añadir conocimiento a la web. Si se consiguiese este objetivo, búsquedas como la siguiente se podrían realizar automáticamente por nuestro ordenador:

"Buscamos un viaje que incluya transporte y alojamiento para irme de vacaciones en algún lugar de la naturaleza exuberante, poca gente y muy buen clima durante el mes de agosto. El alojamiento deberá disponer de facilidades para los niños"

Aunque pueda parecer ciencia ficción, se están haciendo muchos avances relacionados con este tema. Una de las piezas clave del éxito de la Web Semántica son las ontologías. Las ontologías permiten representar información relacionada con uno o más dominios (las podríamos ver como un esquema conceptual). La idea general de la web semántica es que si una página web describe su información utilizando ontologías, preguntas de tipo semántico (como la de arriba) se podrían resolver.

En concreto, el proyecto trata sobre el uso de wikis semánticas. Las wikis son unas herramientas que han demostrado su utilidad en el campo de la web 2.0. Son probablemente la fuente de información más extensa y actualizada que existe. Sin embargo, su información está codificada para ser consumida por humanos. Esto hace que sea especialmente difícil para los programas informáticos interpretar esta información eficientemente. Para solucionar esta limitación aparecen las wikis semánticas.

Una wiki semántica es una wiki que utiliza una ontología (o esquema conceptual) para etiquetar sus contenidos. De esta manera se puede representar información de dominio que sea interpretable tanto por humanos como programas informáticos.

El objetivo de este proyecto es desarrollar una ontología y aplicarla mediante el uso de una wiki semántica (Semantic MediaWiki). Profundizar en las características de esta wiki, su funcionalidad, como almacena la información, como se añade información semántica, como importar ontologías, etc.

Después, desarrollar una ontología y desplegarla, validando la utilidad de la herramienta, sus puntos fuertes y sus posibles debilidades.

1.2 Objetivos del Proyecto

Los objetivos concretos que se quieren conseguir con la temática de este proyecto son los siguientes:

1. Estudiar los conceptos básicos de la web semántica.
2. Conocer que es una ontología.
3. Conocer algunos de los lenguajes de representación de ontologías más utilizados (OWL, RDF...)
4. Conocer algunas de las herramientas de edición y gestión de ontologías más conocidas.
5. Aprender a utilizar ontologías para hacer aplicaciones semánticamente más ricas.
6. Realización de casos prácticos de estas tecnologías, implementando una aplicación web sencilla.

Además, se deberá estudiar en profundidad lo que es una wiki semántica. En particular se estudiará la Semantic MediaWiki y se deberá conocer cómo funciona, qué características tiene, cómo almacena información, cómo se añade información semántica, cómo importar ontologías.

Una vez conseguido el párrafo anterior, se deberá desarrollar una ontología y desplegarla en el entorno anterior para comprobar su funcionamiento y sacar conclusiones al respecto.

El tema de la ontología es libre por lo que se deberá dedicar un tiempo a investigar sobre qué tema realizarla.

1.3 Enfoque del Proyecto

Partiendo de los objetivos del proyecto y los plazos establecidos se han creado una serie de tareas a llevar a cabo que están estructuradas en dos entregas como se puede ver a continuación:

Tarea		Precedencia
1	Preparación PAC 2	
	1.1. Investigar sobre web semántica: recopilar datos y lectura de los mismos.	
	1.2. Redacción capítulo 3 de la memoria referente a web semántica.	1.1.
	1.3. Investigación sobre las ontologías.	
	1.3.1. Recopilación información sobre ontologías.	
	1.3.2. Lectura exhaustiva de la información recopilada.	1.3.1.
	1.4. Decidir qué ontología va a ser diseñada.	

	1.5. Parte de la aplicación práctica.	1.3., 1.4.
	1.5.1. Decidir qué lenguaje utilizar para la implementación de la ontología.	
	1.5.2. Decidir qué herramienta de edición utilizar para la implementación.	
	1.5.3. Desarrollo de la ontología.	1.5.1., 1.5.2.
2	Enviar borrador PAC 2 para una posible corrección.	1
3	Entrega PAC 2.	2
4	Redacción PAC 3.	
	4.1. Redacción capítulo 4.	3
	4.2. Investigar sobre las wikis semánticas: recopilar datos y lectura de los mismos.	
	4.3. Redacción capítulo 5.1 sobre qué es una wiki semántica.	4.2.
	4.4. Investigación Semantic MediaWiki	4.3.
	4.4.1. Recopilación información sobre Semantic MediaWiki.	
	4.4.2. Lectura exhaustiva de la información recopilada.	4.4.1.
	4.4.3. Redacción apartado 5.2 sobre Semantic MediaWiki.	4.4.2
	4.4.4. Buscar documentación sobre instalación de Semantic MediaWiki.	
	4.4.5. Instalación Semantic MediaWiki.	
	4.4.6. Familiarizarse con la wiki semántica instalada y realizar alguna prueba.	4.4.5.
	4.5. Parte de la aplicación práctica.	4.4., 1.5.3.
	4.5.1. Aplicación de la ontología desarrollada sobre la wiki semántica estudiada.	
	4.5.2. Redacción capítulo 6.	4.5.1.
5	Enviar borrador PAC 3 para una posible corrección.	4
6	Entrega PAC 3.	5
7	Entrega Final	6
	7.1. Conclusiones: redacción de las conclusiones del proyecto.	
	7.2. Revisión final: lectura en profundidad del proyecto para su revisión y posibles mejoras.	7.1.
	7.3. Síntesis: selección de las partes del proyecto que serán de utilidad para la realización de la presentación.	7.2.
	7.4. Presentación: desarrollo de una presentación de 20 diapositivas hechas con Power Point con una síntesis del proyecto.	7.3.

8	Entrega Final: memoria y presentación.	7.4.
9	Debate: entre los días 25 y 28 de junio, el tribunal planteará preguntas y situaciones al alumno mediante correo electrónico.	8

Entonces, el enfoque principal del proyecto está basado en la investigación y recopilación de la información necesaria para desarrollar tanto los apartados teóricos como prácticos.

1.4 Planificación del proyecto

Según el plan docente se tienen fijadas las siguientes fechas:

Hitos	Fecha
Inicio PAC 2	12/03/2013
Entrega PAC 2	15/04/2013
Inicio PAC 3	16/04/2013
Entrega PAC 3	21/05/2013
Inicio Entrega Final	22/05/2013
Entrega Entrega Final	11/06/2013
Inicio Debate Virtual	25/06/2013
Fin Debate Virtual	28/05/2013

A partir de estas fechas se dispone de los días siguientes para cada entrega:

- PAC 2: 35 días.
- PAC 3: 36 días.
- Entrega Final: 21 días.
- Debate Virtual: 4 días.

Teniendo en cuenta la disponibilidad de la que dispongo a lo largo de la semana, en un día entre semana podría dedicar alrededor de 2 horas y un día de fin de semana alrededor de 4 horas al proyecto dependiendo. Se podría dar el caso de poder dedicar más tiempo entre semana y menos en fin de semana que se iría compensando entre sí.

Partiendo de esta información y de las tareas del apartado anterior obtenemos la siguiente planificación:

Tareas		Fechas	Tiempo dedicado
1	Preparación PAC 2	12/03/2013	
	1.1. Investigar sobre web semántica: recopilar datos y lectura de los mismos.	12/03-13/03	3 horas
	1.2. Redacción capítulo 3 de la memoria referente a web semántica.	13/03-14/03	3 horas
	1.3. Investigación sobre las ontologías.	15/03	
	1.3.1. Recopilación información sobre ontologías.	15/03-16/03	4 horas
	1.3.2. Lectura exhaustiva de la información recopilada.	16/03-17/03	4 horas
	1.4. Decidir qué ontología va a ser diseñada.	17/03	2 horas
	1.5. Parte de la aplicación práctica.	18/03	
	1.5.1. Decidir qué lenguaje utilizar para la implementación de la ontología.	18/03	2 horas
	1.5.2. Decidir qué herramienta de edición utilizar para la implementación.	19/03	2 horas
	1.5.3. Desarrollo de la ontología.	20/03-09/04	54 horas
2	Crear y enviar borrador PAC 2 para posibles cambios y mejoras.	10/04-14/04	18 horas
3	Entrega PAC 2.	15/04/2013	
4	Redacción PAC 3.	16/04/2013	
	4.1. Redacción apartado 4.	16/04-20/04	10 horas
	4.2. Investigar sobre las wikis semánticas: recopilar datos y lectura de los mismos.	20/04-21/04	4 horas
	4.3. Redacción capítulo 5.1 sobre qué es una wiki semántica.	21/04-22/04	4 horas
	4.4. Investigación Semantic MediaWiki	23/04	
	4.4.1. Recopilación información sobre Semantic MediaWiki.	23/04-24/04	4 horas
	4.4.2. Lectura exhaustiva de la información recopilada.	25/04-26/04	4 horas
	4.4.3. Redacción apartado 5.2 sobre Semantic MediaWiki.	27/04-29/04	10 horas
	4.4.4. Buscar documentación sobre instalación de Semantic MediaWiki.	30/04	2 horas
	4.4.5. Instalación Semantic MediaWiki.	01/05	2 horas
	4.4.6. Familiarizarse con la wiki semántica instalada y realizar alguna prueba.	02/05-03/05	4 horas
	4.5. Parte de la aplicación práctica.	04/05	
	4.5.1. Aplicación de la ontología	04/05-06/05	10 horas

	desarrollada sobre la wiki semántica estudiada.		
	4.5.2. Redacción capítulo 6.	07/05-13/05	18 horas
5	Crear y enviar borrador PAC 3 para posibles cambios y mejoras.	14/05-20/05	18 horas
6	Entrega PAC 3.	21/05/2013	
7	Inicio Entrega Final	22/05/2013	
	7.1. Conclusiones: redacción de las conclusiones del proyecto.	22/05-26/05	14 horas
	7.2. Revisión final: lectura en profundidad del proyecto para su revisión y posibles mejoras.	27/05-29/05	6 horas
	7.3. Síntesis: selección de las partes del proyecto que serán de utilidad para la realización de la presentación.	30/05-01/06	8 horas
	7.4. Presentación: desarrollo de una presentación de 20 diapositivas hechas con Power Point con una síntesis del proyecto.	01/06-10/06	26 horas
8	Fin Entrega Final: memoria y presentación.	11/06/2013	
9	Debate: entre los días 25 y 28 de junio, el tribunal planteará preguntas y situaciones al alumno mediante correo electrónico.	25/06-28/06	4 días

1.4.1 Diagrama de Gantt

El diagrama de Gantt generado a partir de la planificación anterior es lo que se muestra más abajo. Debido a que son varias tareas y no cabe todo en la misma impresión de pantalla, se ha decidido mostrarlo por entregas:

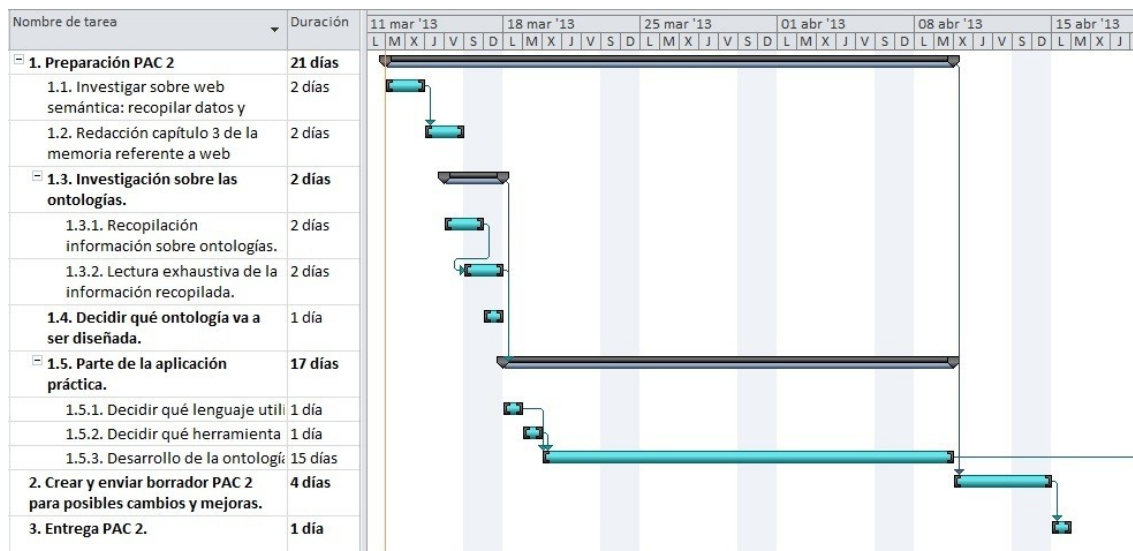


Ilustración 1: Diagrama de Gantt - Entrega Pac 2

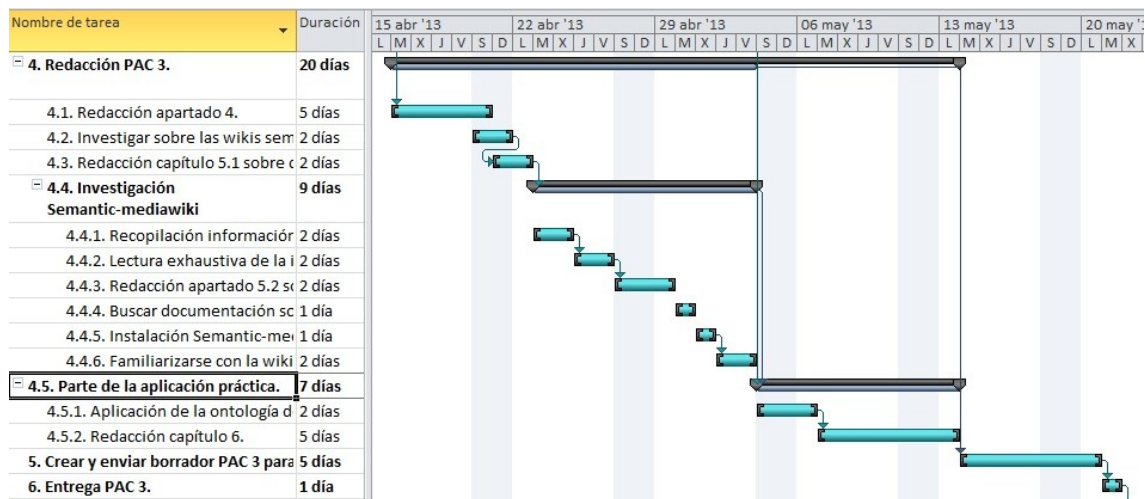


Ilustración 2: Diagrama de Gantt - Entrega Pac 3

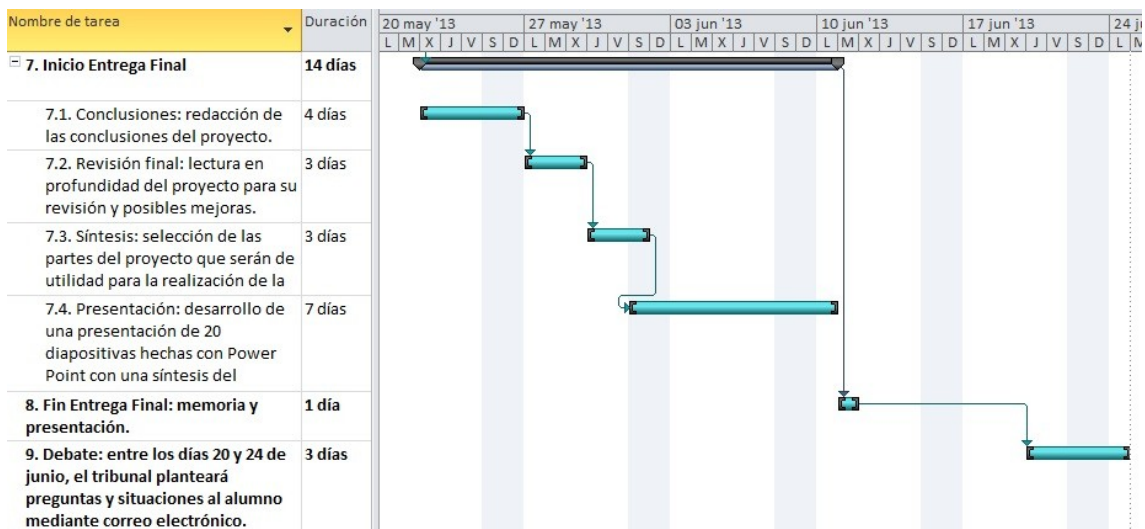


Ilustración 3: Diagrama de Gantt - Entrega final

1.5 Productos obtenidos

Como resultado del desarrollo del proyecto los productos obtenidos serán los siguientes:

- *Plan de proyecto*: Documento con toda la planificación del proyecto. Contiene la subdivisión de las tareas, fechas de entrega, diagramas, etc.
- *Memoria del proyecto*: Se trata del presente documento. Contiene los objetivos planteados, la descripción del trabajo realizado, los resultados y las conclusiones.
- *Presentación del proyecto*: Presentación de diapositivas del contenido resumido del trabajo realizado.

- Productos del caso práctico: Ficheros obtenidos con el software Protégé-2000 además de los que se hayan tenido que generar para la wiki semántica.

1.6 Breve descripción de los otros capítulos de la memoria

La memoria estará estructurada de la siguiente manera:

- Introducción: Se explica brevemente el contenido del proyecto, objetivos, planificación, etc.
- Web Semántica: Se introduce este concepto y se da una visión general de su contenido.
- Ontologías: Se describirá lo que son las ontologías, sus componentes, lenguajes y sistemas de edición existentes.
- Wiki Semántica: Al igual que con la Web Semántica se realizará una introducción al concepto y se describirá el contenido.
- Aplicación práctica: Se dividirá en dos apartados, uno para la ontología desarrollada y otro para la wiki semántica.
- Conclusiones
- Glosario
- Anexos
- Bibliografía

2. Web Semántica

2.1 Introducción

El concepto de web semántica surge a finales de los años 90, gracias a Tim Berners Lee, por la imposibilidad que encontró en introducir información semántica en la World Wide Web, que también fue de su creación, con la intención de no perder el concepto.

La web semántica se basa en la idea de añadir metadatos, que son datos sobre otros datos, semánticos y ontológicos, que veremos en capítulos posteriores, en el contenido de las páginas web. Es decir, el fin es introducir información útil en la web para que máquinas o aplicaciones informáticas puedan hacer uso de la misma de manera correcta al igual que hacen las personas.

2.2 ¿Qué es?

Internet, como lo conocemos, está compuesto por miles de páginas web que en su mayoría están descritas en HTML, un lenguaje de marcado, que es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación. Principalmente este lenguaje se usa para adecuar el aspecto visual de un documento en incluir objetos en el texto como imágenes, esquemas de diálogo, etc. Lo que no puede ofrecer HTML es la categorización de los elementos que lo forman ni relacionar datos entre sí para darles significado.

La web semántica supliría esta deficiencia. Tiene a su disposición componentes tecnológicos de descripción de contenidos que combinados con un lenguaje de marcado (diseñado para describir los datos) ofrecen descripciones explícitas de los recursos de la web mediante un etiquetado específico que es accesible por gestores de contenido que lo saben interpretar, tratar y procesar.

2.3 Componentes

Los principales componentes son los metalenguajes, que son lenguajes que se usan para hablar de otro lenguaje, y los estándares de representación como XML, XML Schema, RDF, RDF Schema y OWL, así como el lenguaje SPARQL para la consulta de datos RDF.

A continuación se ofrece una breve descripción de cada uno de estos componentes y su relación extraída de la web *OWL Web Ontology Language*:

- **XML** aporta la sintaxis superficial para los documentos estructurados, pero sin dotarles de ninguna restricción sobre el significado.

- **XML Schema** es un lenguaje para definir la estructura de los documentos XML.
- **RDF** es un modelo de datos para los recursos y las relaciones que se puedan establecer entre ellos. Aporta una semántica básica para este modelo de datos que puede representarse mediante XML.
- **RDF Schema** es un vocabulario para describir las propiedades y las clases de los recursos RDF, con una semántica para establecer jerarquías de generalización entre dichas propiedades y clases.
- **OWL** es un lenguaje para definir ontologías mediante la descripción detallada de propiedades y clases.
- **SPARQL** es un lenguaje de consulta de conjuntos de datos RDF. Además en dicha especificación también se incluye un formato XML que detalla el modo en el que se estructuran los resultados obtenidos.

2.4 Estructura

La web semántica se podría estructurar en capas de tecnologías web y estándares siguiendo los principios definidos por W3C. A continuación se ofrece una descripción:

- En las capas más bajas encontramos *Unicode* y *URI* que aseguran que se utilizarán conjuntos de caracteres internacionales y proporcionan los medios para identificar objetos.
- La capa XML que incluye los *Namespaces*, que son contenedores abstractos en el que un grupo de uno o más identificadores únicos pueden existir, y las definiciones de esquemas, asegura la integración de las definiciones que se hagan con otros estándares basados en XML.
- La capa RDF permite la creación de axiomas sobre los objetos y la definición de vocabularios que se podrán identificar mediante una URI. En esta capa se pueden asignar tipos a los recursos y a los enlaces.
- La capa OWL, RDFS, SKOS que correspondería con las ontologías, soporta la evolución de los vocabularios definidos en la capa inferior ya que permite la definición de relaciones entre diferentes conceptos.
- Y por último, las capas superiores, Lógica y Confianza, están en vías de estudio. La primera permite la definición de reglas y la segunda si se confiará en el resultado de la ejecución de las reglas anteriormente definidas.

A continuación se muestra una imagen de la estructura antes comentada:

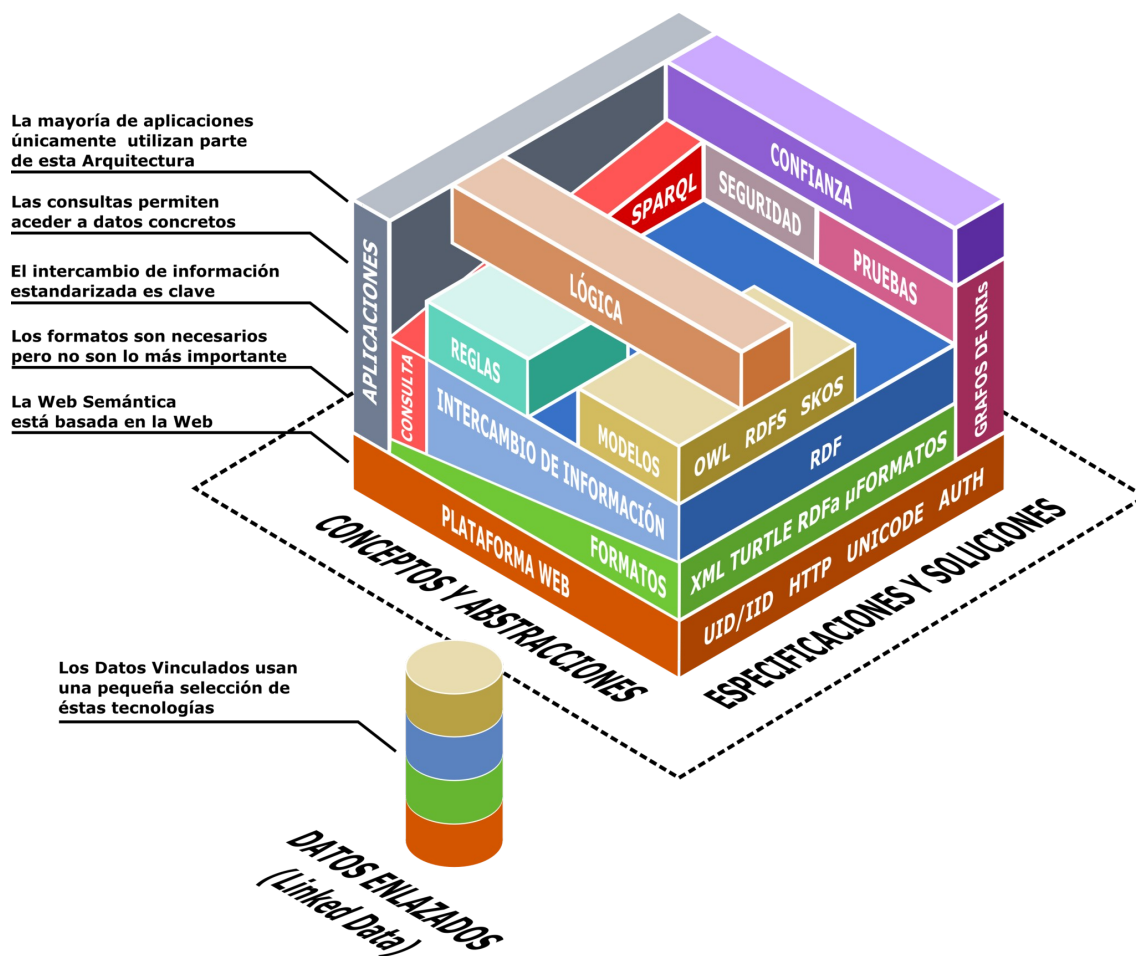


Ilustración 4: Web Semántica - Estructura web semántica

2.5 Ventajas

La usabilidad y aprovechamiento de la web y sus recursos interconectados puede aumentar con la web semántica gracias a:

- Los documentos etiquetados con información semántica. Se pretende que esta información sea interpretada por el ordenador con una capacidad comparable a la del lector humano.
- Vocabularios comunes de metadatos (Ontología) y mapas entre vocabularios que permitan a quienes elaboran los documentos disponer de nociones claras sobre cómo deben etiquetarlos para que los agentes automáticos puedan usar la información contenida en los metadatos.
- Agentes automáticos que realicen tareas para los usuarios de estos metadatos de la Web Semántica.
- Servicios Web (a menudo con agentes propios) que provean de información a los agentes.

2.6 Inconvenientes y desventajas

Entre los más conocidos se encuentran las dificultades tecnológicas y la falta de interés de los propietarios de las páginas web.

Tecnológicamente existen herramientas para expresar la información en el formato que la web semántica necesita. La más especializada es OWL. Lo que no existe es una herramienta que traduzca automáticamente la información de la web en marcas OWL. Esto implica que los creadores de las páginas web hagan a mano este proceso para que pueda ser interpretado.

El segundo problema que comentábamos es el de la pasividad de los propietarios web. El modelo de negocio actual de las páginas web proviene de la publicidad, que se cobra al entrar un usuario y no una máquina.

3. Ontologías

3.1 Qué son

El concepto de ontología ha sido usado durante mucho tiempo en filosofía, específicamente como una rama del campo de la metafísica que estudia lo que hay. Las preguntas como: ¿Existe Dios? ¿Existen los pensamientos como tales?, son denominadas preguntas de ontología. Además, también se estudia la manera en que se relacionan las entidades que existen, por ejemplo, la relación entre un universal (verde) y un particular que "lo tiene" (esta pera).

En los últimos años, este concepto, se ha introducido en el campo de la informática con un sentido muy parecido. En este área, una ontología, define los conceptos que hay en un dominio específico, describe las relaciones entre los mismos y sus propiedades. En general, se utilizan para describir y representar un área de conocimiento, siendo ésta, por ejemplo, la medicina, espectáculos, historia, etc.

La información que contienen las ontologías puede ser entendida por los ordenadores. Son usadas por personas, bases de datos y aplicaciones que necesitan compartir información de un dominio en concreto, como los antes mencionados.

Podemos descomponerla en los siguientes aspectos:

- *Conceptos*, también denominados *clases*, descritos explícita y formalmente.
- *Propiedades*, que pertenecen a los conceptos, para describir sus características y atributos.
- *Restricciones* de las propiedades.

Una ontología contiene una estructura jerárquica de clases o conceptos. Cada clase podría tener más subclases y a su vez, estas subclases, tener

otras subclases, que representan aspectos más específicos de su superclase. También se le puede llamar taxonomía.

Las ventajas de usar una ontología son las siguientes:

- Facilita la comprensión de las ideas clave de un dominio concreto de aplicación.
- Aporta los términos que podemos emplear cuando creamos los documentos RDF sobre un dominio.
- Permite reutilizar conocimiento existente sobre el dominio.
- Compartir el conocimiento entre las personas y las máquinas.

Existe un método, creado por Natalya F. Noy y Deborah L. McGuinness de la Universidad de Standford, que ayuda en el proceso de desarrollo de una ontología que se lista a continuación:

1. Determinar el dominio y alcance de la ontología.

Para tal fin se debe poder responder a estas preguntas:

- ¿Cual es el dominio que la ontología cubrirá?
- ¿Para qué se usará la ontología?
- ¿Qué tipo de preguntas debe proveer respuestas la información de la ontología?
- ¿Quién usará y mantendrá la ontología?

2. Considerar reutilizar ontologías existentes.

Reutilizar una ontología existente es una buena elección, pero sólo cuando ésta es adecuada para nuestro propósito. Sin embargo este es un punto muy importante a considerar, sobre todo si nuestra aplicación tiene que interactuar con otras que ya están usando una ontología.

3. Enumerar términos importantes en la ontología.

Este paso es un requisito necesario antes de definir las clases y la jerarquía de clases. Los autores sugieren que se construya una lista de términos del dominio sin preocuparse del solapamiento de los conceptos que estos representan, de las relaciones entre los términos o de cualquier propiedad que los conceptos puedan tener. El objetivo es asegurarse de que todos los términos importantes están incluidos y así facilitar la definición de las clases y de la jerarquía de clases.

4. Definir las clases y la jerarquía de clases.

Se proponen tres enfoques:

- *Top-down*, consiste en definir las clases mas generales y a partir de ellas las más especializadas.

- *Bottom-up*, consiste en hacer lo contrario al anterior. Empezar por las clases más especializadas hasta llegar a las más generales.
- Combinar los otros dos métodos, cambiando de uno al otro a medida que se definen las clases y su jerarquía.

Hay que tener en cuenta que una jerarquía de clases es una relación "is-a", en la que la relación entre subclases es siempre transitiva.

5. Definir las propiedades de las clases.

En este punto los autores sugieren que se consideren los siguientes tipos de propiedades:

- Intrínsecas, son aquéllas inherentes o esenciales de una clase.
- Extrínsecas, son aquéllas que son externas o no esenciales de una clase.
- Partes, elementos que son partes del objeto que representa la clase.
- Relaciones con otros individuos, con otras clases.

6. Definir las restricciones de las propiedades.

Por ejemplo, el tipo de valor que puede admitir, su cardinalidad, su dominio, su rango, etc.

7. Crear las instancias.

Este paso es el último que realizamos, y puede ser opcional, ya que el documento que contiene la definición ontológica no necesita tener incluidas instancias.

3.2 Lenguajes de representación

A continuación se describen algunos de los lenguajes existentes de los que puede estar formada una ontología. Empezaremos con XML puesto que es la base de todos los lenguajes ontológicos y seguiremos con los demás que son ampliaciones y mejoras de sus predecesores.

3.2.1 XML

XML atiende a **eXtensible Markup Language**. Es un lenguaje de marcado que permite definir la gramática de lenguajes específicos para estructurar documentos grandes. Fue desarrollado por el W3C (World Wide Web Consortium) a partir del lenguaje SGML.

El uso que se da a este lenguaje es muy amplio, desde bases de datos e integración de sistemas con distintas plataformas hasta editores de texto y hojas de cálculo.

La información en un documento XML está estructurada, esto quiere decir que está compuesta de partes bien definidas y que estas partes a su vez están compuestas de otras partes. Se podría decir que la información al quedar de forma estructurada tendría forma de árbol. Por ejemplo, un coche está formado por un chasis y cuatro ruedas, y éstas, a su vez, por llantas. Pues bien, estas partes, en un documento XML, son llamadas *elementos* y están señalizadas por *etiquetas*.

Una *etiqueta* consiste en una marca hecha en el documento que señala una porción de éste como un elemento. Crea un pedazo de información con un sentido claro y definido. Las etiquetas tienen la forma `<nombre>`, donde *nombre* es el nombre del elemento que se está señalando.

A continuación se describe más explícitamente el contenido de un documento XML:

- **Prólogo**

Aunque no es obligatorio, los documentos XML pueden empezar con unas líneas que describen la versión XML, el tipo de documento y otras cosas.

El prólogo de un documento XML contiene:

- Una declaración XML: es la sentencia que declara al documento como un documento XML.
- Una declaración de tipo de documento: enlaza el documento con su DTD (definición de tipo de documento), o el DTD puede estar incluido en la propia declaración o ambas cosas al mismo tiempo.
- Uno o más comentarios e instrucciones de procesamiento.

Un ejemplo de prólogo sería:

```
<?xml version="1.0" encoding="UTF-8"?>
```

- **Cuerpo**

A diferencia del prólogo, el cuerpo no es opcional en un documento XML, el cuerpo debe contener sólo un elemento raíz, característica indispensable también para que el documento esté bien formado. Sin embargo, es necesaria la adquisición de datos para su buen funcionamiento.

A continuación se muestra un ejemplo corto de un documento XML con su prólogo y cuerpo:

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">

<Edit_Mensaje>
  <Mensaje>
    <Remitente>
      <Nombre>Nombre del remitente</Nombre>
      <Mail>Correo del remitente </Mail>
    </Remitente>
    <Destinatario>
      <Nombre>Nombre del destinatario</Nombre>
      <Mail>Correo del destinatario</Mail>
    </Destinatario>
    <Texto>
      <Asunto>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Asunto>
      <Parrafo>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Parrafo>
    </Texto>
  </Mensaje>
</Edit_Mensaje>

```

Tabla 1: XML - Ejemplo documento XML

El cuerpo de un documento XML puede estar formado a su vez por los siguientes elementos:

- **Elementos**

Los elementos XML pueden tener contenido (más elementos, caracteres o ambos), o bien ser elementos vacíos.

- **Atributos**

Los elementos pueden tener atributos, que son una manera de incorporar características o propiedades a los elementos de un documento. Deben ir entre comillas.

Por ejemplo, un elemento «felino» puede tener un atributo «Silvestre» y un atributo «tipo», con valores «no tiene bigotes» y «callejero» respectivamente. Y se expresa como se ve a continuación:

```
<Felino Silvestre="no tiene bigotes" tipo="callejero">A Silvestre le pareció ver a un lindo canario...</Estudiante>
```

- **Entidades predefinidas**

Estas entidades se utilizan para representar caracteres especiales para que no sean interpretados como marcado en el procesador XML.

Por ejemplo tenemos la siguiente entidad predefinida:

& carácter: &.

- **Espacios de nombres**

Los espacios de nombres XML permiten separar semánticamente los elementos que forman un documento XML.

Documentos XML bien formados y control de errores

Los documentos denominados como «bien formados» (del inglés *well formed*) son aquellos que cumplen con todas las definiciones básicas de formato y pueden, por lo tanto, analizarse correctamente por cualquier analizador sintáctico (*parser*) que cumpla con la norma. Se separa esto del concepto de validez que se explica más adelante.

Estos documentos siguen las siguientes pautas:

- Los documentos han de seguir una estructura estrictamente jerárquica en lo que respecta a las etiquetas que delimitan sus elementos. Una etiqueta debe estar correctamente incluida en otra, es decir, las etiquetas deben estar correctamente anidadas. Los elementos con contenido deben estar correctamente cerrados.
- Los documentos XML sólo permiten un elemento raíz del que todos los demás sean parte.
- Los valores de tipo atributo en XML siempre deben estar encerrados entre comillas simples o dobles.
- El XML es sensible a mayúsculas y minúsculas. Existe un conjunto de caracteres llamados espacios en blanco (espacios, tabuladores, retornos de carro, saltos de línea) que los procesadores XML tratan de forma diferente en el marcado XML.
- Es necesario asignar nombres a las estructuras, tipos de elementos, entidades, elementos particulares, etc. En XML los nombres tienen alguna característica en común.
- Las construcciones como etiquetas, referencias de entidad y declaraciones se denominan marcas; son partes del documento que el procesador XML espera entender. El resto del documento entre marcas son los datos «entendibles» por las personas.

Validez

Como se comentaba más arriba, que un documento esté «bien formado» se refiere a su estructura sintáctica básica, es decir, que se componga de elementos, atributos y comentarios como XML especifica que se escriban. La validez de un documento viene definida por la especificación de la relación entre los distintos elementos en el documento y que un documento específico debe cumplir.

Esta relación entre elementos se especifica en un documento externo o definición (expresada como DTD—*Document Type Definition*, 'Definición de Tipo de Documento'— o como XSchema). Crear una definición equivale a crear un nuevo lenguaje de marcado, para una aplicación específica.

Document Type Definition

Un DTD (en español "definición de tipo de documento") define los tipos de elementos, atributos y entidades permitidas, y puede expresar algunas limitaciones para combinarlos. Los documentos XML que se ajustan a su DTD son denominados válidos.

Aquí está el ejemplo de código del DTD del documento «Edit_Mensaje.dtd» presentado como ejemplo más arriba:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Este es el DTD de Edit_Mensaje -->

<!ELEMENT Mensaje (Remitente, Destinatario, Texto)*>
<!ELEMENT Remitente (Nombre, Mail)>
<!ELEMENT Nombre (#PCDATA)>
<!ELEMENT Mail (#PCDATA)>
<!ELEMENT Destinatario (Nombre, Mail)>
<!ELEMENT Nombre (#PCDATA)>
<!ELEMENT Mail (#PCDATA)>
<!ELEMENT Texto (Asunto, Parrafo)>
<!ELEMENT Asunto (#PCDATA)>
<!ELEMENT Parrafo (#PCDATA)>
```

Tabla 2: XML - Ejemplo DTD

XML Schema (XSD)

Un XSD es algo similar a un DTD. Define qué elementos puede contener un documento XML, cómo están organizados y qué atributos y de qué tipo pueden tener sus elementos.

Ventajas de los XSD frente a los DTD

Las ventajas son las siguientes:

- Usan sintaxis de XML, al contrario que los DTD.
- Permiten especificar los tipos de datos.
- Son extensibles.

Herramientas para trabajar con XML

Cualquier procesador de texto que sea capaz de producir archivos .txt es capaz de generar XML. Existen otros entornos de desarrollo más preparados como Eclipse o Netbeans, entre otros, que facilita la creación y lectura puesto que reconoce el formato y ayuda a que esté bien formado.

3.2.2 RDF

RDF cuyas siglas en inglés significan **R**esource **D**escription **F**ramework es una recomendación de W3C diseñado originalmente como un modelo de datos de metadatos. Hoy en día se usa como un lenguaje para la representación de la información de los recursos de la web,

concretamente, define sintaxis y modelos de datos para la representación semántica de los datos.

Motivaciones

Las motivaciones por las que se llevó a cabo el desarrollo de este lenguaje fueron:

1. Crear metadatos sobre la información de la web.
2. Aplicaciones que requieren modelos de información abiertos.
3. Permitir que la información sea procesada fuera del entorno en el que fue creada.
4. Integración entre aplicaciones de manera que sean capaces de intercambiar información para crear nueva.
5. Proceso automatizado de la información por agentes de software.

Metas

Debido a las motivaciones mencionadas, se establecieron unas metas para llevar a cabo su desarrollo:

1. Modelo de datos simple.
2. Semánticas formales e inferencia probable.
3. Basarse en vocabulario URI.
4. Basarse en sintaxis XML.
5. Poder usar XML schema datatypes.
6. Cualquier persona pueda realizar declaraciones sobre cualquier fuente.

Conceptos

Los conceptos principales que se usan en RDF son los siguientes:

- **Modelo de datos tipo grafo**

La estructura de cualquier expresión en RDF es una colección de tripletes, consistiendo cada uno de ellos en un sujeto, predicado (también llamado propiedad) y objeto. Este conjunto es denominado Grafo RDF. Se puede representar como en la siguiente figura:

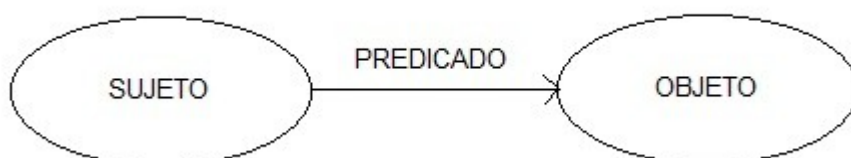


Ilustración 5: RDF - Grafo RDF

Cada triplete representa una declaración de una relación entre cosas delimitadas por los nodos que los conectan. Los nodos suelen ser los

sujetos y los objetos. La dirección del arco siempre apunta hacia los objetos.

El significado de este grafo en RDF es la conjunción (AND lógica) de las declaraciones correspondientes a todos los tripletes existentes para el mismo.

- **Vocabulario basado en URI e identificación de nodos**

1. Los nodos antes mencionados pueden ser una URI, un literal o estar en blanco (no es ni un literal ni una referencia URI y puede ser usado en varias declaraciones RDF).
2. Las propiedades son referencias URI.
3. Una referencia URI o literal que se usa en un nodo identifica lo que el nodo representa.
4. Una referencia URI usada como predicado identifica una relación entre las cosas representadas en los nodos que conecta.
5. Un predicado también puede ser un nodo en el grafo a pesar de tener una referencia URI.

- **Tipos de datos**

Los tipos de datos (Datatypes) se utilizan para representar valores como enteros, n^o en coma flotante y fechas.

Podrían describirse tres tipos de representaciones diferentes: un espacio léxico, de valor y un mapeo de léxico-a-valor. Por ejemplo:

Espacio de valor	{T,F}
Espacio léxico	{"0","1","true","false"}
Mapeo léxico-a-valor	{<"true",T>,<"1",T>,<"0",F>,<"false",F>}

Tabla 3: RDF - Ejemplo tipos de datos

RDF predefine sólo un tipo de dato, *rdf:XMLLiteral*, utilizado para incrustar XML en RDF, por lo que no hay un concepto de tipo número o fechas que se pueda construir o cualquier otro tipo de valor común. RDF utiliza tipos de datos definidos de forma separada e identificados con referencias URI. Por este motivo se prevé usar ampliamente los tipos de datos predefinidos de XML Schema.

- **Literales**

Se usan para identificar valores como números y fechas como una representación léxica. También pueden representarse mediante referencias URI pero se recomienda el uso intuitivo de literales normales.

Un literal puede ser el objeto de una declaración RDF, pero nunca el sujeto o predicado.

Pueden ser de dos tipos:

1. *Literal plano*, que es un string combinado, opcionalmente, con algún lenguaje de tags.
2. *Literal escrito*, que es un string combinado con un tipo de datos URI.

Ejemplos:

Literal escrito	Mapeo léxico-a-valor	Valor
<xsd:boolean, "true">	<"true",T>	T
<xsd:boolean, "1">	<"1",T>	T
<xsd:boolean, "false">	<"false",F>	F
<xsd:boolean, "0">	<"0",F>	F

Tabla 4: RDF - Ejemplo literales

- **Sintaxis XML**

Se da una referencia en el apartado 3.2.1.

- **Expresión de hechos simples**

La relación entre dos cosas se puede representar o describir con hechos simples. Estos hechos pueden ser representados como un triplete RDF en el cual el predicado da nombre a la relación y el sujeto y el objeto denotan las dos cosas. En una base de datos relacional puede realizarse una representación similar en una de las filas de una tabla. La tabla tiene dos columnas que corresponden con el sujeto y el objeto del triplete y el predicado correspondería con el nombre de la tabla.

- **Vinculación**

Una expresión RDF A se dice que implica una expresión RDF B si toda posible disposición de cosas en el mundo que hacen A verdadero también hacen B verdadero. Partiendo de esta base, si la verdad de A se puede asumir o demostrar entonces la verdad de B puede ser inferida.

RDF Schema

RDF Schema es una extensión semántica de RDF. Proporciona elementos básicos para la descripción de vocabularios. La última versión de la recomendación se publicó en 2004 por W3C.

Las características principales de este lenguaje son:

- Los archivos de RDF Schema (RDFS) es un archivo RDF, es decir, tienen la misma sintaxis y la misma estructura.
- Se basa en XML.
- Es extensible.

RDFS tiene las siguientes clases diferenciadas por los siguientes tipos:

- Básicas
 - *rdfs:Class* permite declarar recursos como clases para otros recursos
 - *rdfs:Resource* es la clase a la que pertenecen todos los recursos.
 - *rdfs:Literal* es la clase de todos los valores literales, cadenas y enteros
 - *rdfs:Datatype* es la clase que abarca todos los tipos de datos definidos en el modelo RDF.
- Definen relaciones
 - *rdfs:subClassOf* es una instancia de *rdf:Property* que permite definir jerarquías. Relaciona una clase con sus superclases.
 - *rdfs:supPropertyOf* es una instancia de *rdf:Property* que permite definir jerarquías de propiedades.
- Restricciones de propiedades
 - *rdfs:domain* es una instancia de *rdf:Property* que especifica el dominio de una propiedad P, *rdfs:property*. Es decir, la clase de los recursos que aparecen como sujetos en los tripletes donde P es predicado.
 - *rdfs:range* es una instancia de *rdf:Property* que especifica el rango de una propiedad P, *rdfs:range*. Es decir, la clase de los recursos que aparecen como objetos en los tripletes donde P es predicado.
- Otras
 - *rdfs:ConstraintResource* es la clase que agrupa a todas las restricciones.
 - *rdfs:ConstraintProperties* es la clase que contiene todas las propiedades que definen restricciones. Sus instancias son *rdfs:range* y *rdfs:domain*.
 - *rdfs:seeAlso* es una instancia de *rdf:Property* que relaciona un recurso con otro que proporciona información adicional sobre el primero.
 - *rdfs:isDefinedBy* es una instancia de *rdf:Property* para relacionar un recurso con los lugares donde se encuentra el recurso sujeto.
 - *rdfs:label* es una instancia de *rdf:Property* que se usa para proporcionar una versión claramente entendible del nombre de un recurso.
 - *rdfs:comment* es una instancia de *rdf:Property* que se usa para proporcionar una descripción de un recurso.

A continuación se muestra un ejemplo de una declaración de tripletes, su grafo y su documento RDF:

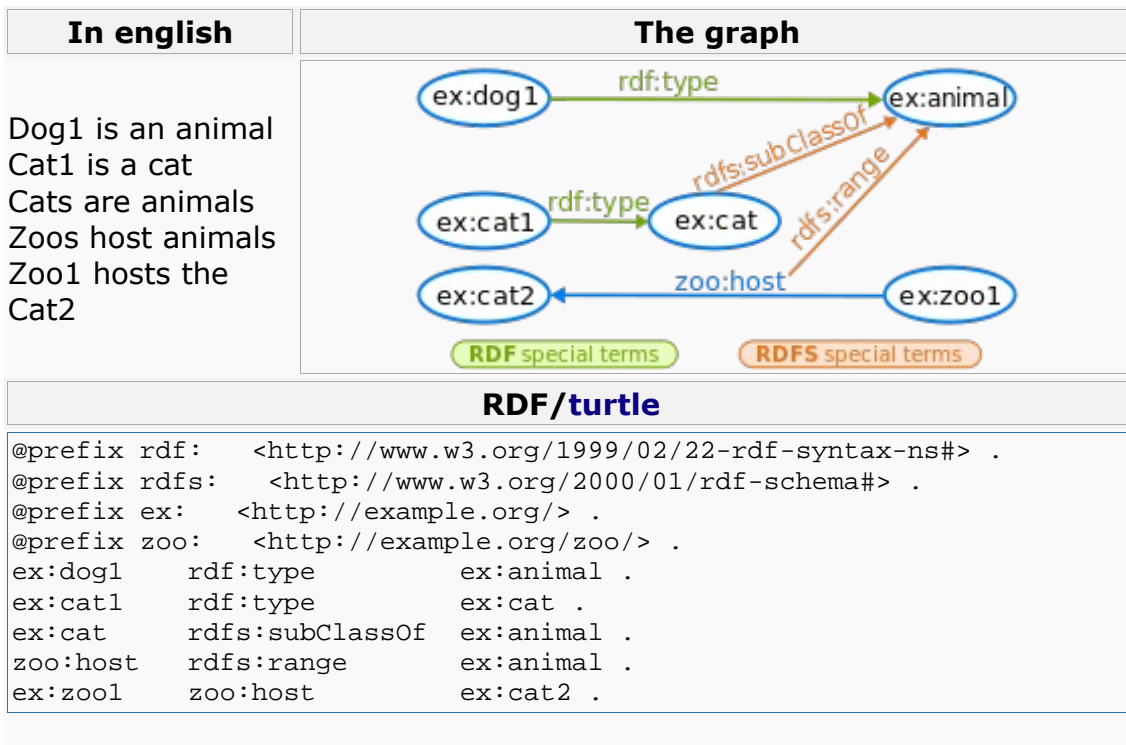


Ilustración 6: RDF - Ejemplo completo RDF

3.2.3 OWL

Las siglas OWL atienden a Web Ontology Language (Lenguaje Ontológico para Web). Es una recomendación de la W3C, concretamente la que se lanzó en 2004 pues después de ésta ha salido una nueva recomendación en 2012 de una nueva versión de OWL, OWL 2. Este proyecto se ha centrado en la recomendación de 2004 por lo que será la que se describa.

OWL fue diseñado para aplicaciones que necesitan procesar el contenido de la información en lugar de sólo presentarla para personas. Facilita la interpretación de contenido web por máquinas mejor que otros lenguajes anteriormente mencionados. Lo consigue proporcionando más vocabulario junto con semántica formal.

Tiene tres sublenguajes que han sido diseñados específicamente para las necesidades de comunidades de desarrolladores. Son los siguientes:

OWL Lite

Se utiliza para realizar clasificaciones jerárquicas y restricciones simples. Por ejemplo, ofrece restricciones de cardinalidad pero únicamente valores de 0 y 1.

Estas son las construcciones que se pueden hacer en este sublenguaje:

<p>Característica de RDF Schema:</p> <ul style="list-style-type: none"> • <i>Class (Thing, Nothing)</i> • <i>rdfs:subClassOf</i> • <i>rdf:Property</i> • <i>rdfs:subPropertyOf</i> • <i>rdfs:domain</i> • <i>rdfs:range</i> • <i>Individual</i> 	<p>(In)Equivalencias:</p> <ul style="list-style-type: none"> • <i>equivalentClass</i> • <i>equivalentProperty</i> • <i>sameAs</i> • <i>differentFrom</i> • <i>AllDifferent</i> • <i>distinctMembers</i>
<p>Características de propiedades:</p> <ul style="list-style-type: none"> • <i>ObjectProperty</i> • <i>DatatypeProperty</i> • <i>inverseOf</i> • <i>TransitiveProperty</i> • <i>SymmetricProperty</i> • <i>FunctionalProperty</i> • <i>InverseFunctionalProperty</i> 	<p>Restricciones de propiedades:</p> <ul style="list-style-type: none"> • <i>Restriction</i> • <i>onProperty</i> • <i>allValuesFrom</i> • <i>someValuesFrom</i>
<p>Cardinalidad restringida:</p> <ul style="list-style-type: none"> • <i>minCardinality</i> (sólo 0 or 1) • <i>maxCardinality</i> (sólo 0 or 1) • <i>cardinality</i> (sólo 0 or 1) 	<p>Información de cabecera:</p> <ul style="list-style-type: none"> • <i>Ontology</i> • <i>imports</i>
<p>Intersección de clases:</p> <ul style="list-style-type: none"> • <i>intersectionOf</i> 	<p>Versionado:</p> <ul style="list-style-type: none"> • <i>versionInfo</i> • <i>priorVersion</i> • <i>backwardCompatibleWith</i> • <i>incompatibleWith</i> • <i>DeprecatedClass</i> • <i>DeprecatedProperty</i>
<p>Anotaciones en propiedades:</p> <ul style="list-style-type: none"> • <i>rdfs:label</i> • <i>rdfs:comment</i> • <i>rdfs:seeAlso</i> • <i>rdfs:isDefinedBy</i> • <i>AnnotationProperty</i> • <i>OntologyProperty</i> 	<p>Tipos de datos</p> <ul style="list-style-type: none"> • <i>xsd datatypes</i>

Tabla 5: OWL - Construcciones en OWL Lite

OWL DL

Se utiliza para desarrollos que requieren máxima expresividad conservando la completitud computacional y decibilidad. Se incluyen todas las

construcciones del lenguaje pero sólo se puede utilizar bajo ciertas restricciones. Por ejemplo, una clase no puede ser una instancia de otra clase. OWL DL es llamado así por su correspondencia con la descripción lógica, un campo de investigación que ha estudiado las lógicas que constituyen la base formal de OWL.

Además de las construcciones que se pueden realizar en OWL Lite también se añaden las siguientes:

<p>Axiomas clase:</p> <ul style="list-style-type: none"> • <i>onOf, dataRange</i> • <i>disjoiningWith</i> • <i>equivalentClass</i>(para expresiones de clase) • <i>rdfs:subClassOf</i> (para expresiones de clase) 	<p>Combinaciones Booleanas de expresiones de clase:</p> <ul style="list-style-type: none"> • <i>unionOf</i> • <i>complementOf</i> • <i>intersectionOf</i>
<p>Cardinalidad arbitraria:</p> <ul style="list-style-type: none"> • <i>minCardinality</i> • <i>maxCardinality</i> • <i>cardinality</i> 	<p>Información de relleno:</p> <ul style="list-style-type: none"> • <i>hasValue</i>

Tabla 6: OWL - Construcciones en OWL DL

OWL Full

Se utiliza para desarrollos que requieren la máxima expresividad y la libertad sintáctica de RDF sin garantías computacionales. Por ejemplo, en OWL Full una clase puede ser tratada simultáneamente como una colección de individuos y como individuo por derecho propio. OWL Full permite que una ontología aumente el significado del vocabulario predefinido (RDF o OWL). Es poco probable que cualquier software de razonamiento sea capaz de soportar el razonamiento completo para cada característica de OWL Full.

Las construcciones que se pueden hacer con este sublenguaje son las mismas que con OWL DL.

Nota: Para una descripción detallada de las construcciones dirigirse a <http://www.w3.org/TR/owl-features/>.

Cada uno de estos sublenguajes es una extensión de su predecesor, tanto para lo que puede ser expresado legalmente como para lo que puede suponerse de forma válida. A continuación se muestran una serie de afirmaciones sobre estos lenguajes cuyos inversos no son válidos:

- Toda ontología legal en OWL Lite es una ontología legal en OWL DL.
- Toda ontología legal en OWL DL es una ontología legal en OWL Full.
- Toda suposición válida en OWL Lite lo es para OWL DL.

- Toda suposición válida para OWL DL lo es para OWL Full.

La elección de un lenguaje u otro se basará en lo que el desarrollador desee conseguir, según la expresividad que se quiera alcanzar, el uso de las funcionalidades RDF completas, entre otros detalles. OWL Full se puede ver como una extensión de RDF mientras que OWL Lite y DL son una restricción de RDF.

3.3 Herramientas de creación, edición y gestión

A continuación se presentan algunas de las herramientas que se pueden utilizar para crear, editar y gestionar ontologías.

3.3.1 WORDNET

Wordnet es una base de datos léxica que contiene las palabras de la lengua inglesa. Se encarga de agrupar las palabras en conjuntos de sinónimos llamados *synsets*, proporcionar definiciones cortas y generales y almacenar las relaciones semánticas entre los conjuntos de sinónimos. Tiene un propósito doble, combinar lo que sería un diccionario y un tesoro para que su uso sea más intuitivo y soportar análisis automáticos de texto y aplicaciones de Inteligencia Artificial.

Su desarrollo comenzó en 1985 por el Cognitive Science Laboratory de la Universidad de Princeton dirigido por el profesor de psicología George A. Miller. A lo largo de los años ha ido creciendo en desarrolladores y financiamiento.

Wordnet distingue entre sustantivos, verbos, adjetivos y adverbios puesto que siguen diferentes reglas gramaticales. Cada *synset* contiene un grupo de palabras que son sinónimos o *collocations* (es una secuencia de palabras que unidas toman un significado específico). Puede ser que diferentes significados de una misma palabra esté en distintos *synsets*. Se definen *glosses* como ejemplo para entender el significado de los *synset*. Por ejemplo:

"good, right, ripe – (most suitable or right for a particular purpose; "a good time to plant tomatoes"; "the right time to act"; "the time is ripe for great sociological changes") "

La mayoría de los *synsets* están conectados a otros *synsets* mediante numerosas relaciones semánticas. Estas relaciones varían según el tipo de palabra, y se incluyen:

<p>Sustantivos</p> <ul style="list-style-type: none"> • Hypernyms: Y es un hypernym de X si cada X es un (del tipo de) Y (canino es un hypernym de perro). • Hyponyms: Y es un hyponym de X si cada Y es un (del tipo de) X (perro es un hyponym de canino). • Coordinate terms: Y es un coordinate term de X si X y Y comparten un hypernym (lobo es un coordinate term de perro y perro es un coordinate term de lobo). • Holonym: Y es un holonym de X si X es parte de Y (edificio es un holonym de ventana). • Meronym: Y es un meronym de X si Y es una parte de X (ventana es una parte de edificio).
<p>Verbos</p> <ul style="list-style-type: none"> • Hypernym: el verbo Y es un hypernym del verbo X si la actividad X es un (del tipo de) Y (percibir es un hypernym de escuchar). • Troponym: el verbo Y es un troponym del verbo X si la actividad Y está realizando X de alguna manera (susurrar es un troponym de hablar). • Entailment: el verbo Y es entailment de X si haciendo X se debe estar haciendo Y (dormir es un entailment de roncar). • coordinate terms: son los verbos que comparte un hypernym común (susurrar y gritar).
<p>Adjetivos</p> <ul style="list-style-type: none"> • Sustantivos relacionados • Similar a • Participios de verbos
<p>Adverbios</p> <ul style="list-style-type: none"> • La raíz de los adjetivos

Tabla 7: WORDNET - Relaciones semánticas

Mientras que las relaciones semánticas se aplican a todos los miembros de un synset porque comparten significado, no todos son mutuamente sinónimos, las palabras también pueden estar conectadas a otras palabras a través de relaciones léxicas, incluyendo antónimos.

Los sustantivos y los verbos están organizados en jerarquías. Por ejemplo, el primer significado de la palabra perro tendrá la siguiente jerarquía hypernym; las palabras del mismo nivel son sinónimos unas de otras. Cada conjunto de sinónimos tiene un único indexador y comparte sus propiedades, como una definición gloss (o diccionario).

```
dog, domestic dog, Canis familiaris
=> canine, canid
=> carnivore
=> placental, placental mammal, eutherian, eutherian mammal
=> mammal
=> vertebrate, craniate
=> chordate
=> animal, animate being, beast, brute, creature,
fauna
=> ...
```

Ilustración 7: WORDNET - Ejemplo Wordnet

En el caso de los adjetivos la organización es diferente, la jerarquía y los conceptos léxicos no se aplican de la misma manera que para los sustantivos y los verbos. El grafo de los sustantivos es mucho más profundo, los verbos tienen una estructura más densa y los adjetivos están organizados en varios clusters diferentes. Los adverbios están organizados en términos de los adjetivos de los que derivan y por tanto heredan su estructura.

WordNet como una ontología

La relación hiperónimo/hipónimo entre los synsets podría ser interpretada como una relación de especialización entre categorías conceptuales, es decir, podría ser usado como una ontología.

Antes de utilizarlo como tal, deberían corregirse bastantes inconsistencias semánticas básicas como la existencia de especializaciones comunes para categorías exclusivas y redundancias en la jerarquía de especialización. Estas correcciones ya se han tenido en cuenta para su uso como ontología y se han desarrollado y documentado.

Problemas y limitaciones

WordNet no provee información como la etimología, pronunciación y la forma de los verbos irregulares y contiene sólo información limitada sobre su uso. A pesar de que contiene un rango amplio de palabras, no cubre vocabulario de un dominio específico. Si una aplicación desea usar WordNet debe hacerlo en dominios específicos cubiertos por el mismo de lo contrario no podrán usarlo.

Aplicaciones

Se ha usado para los siguientes propósitos:

- Desambiguación del significado de palabras.
- Recuperación de información.
- Clasificación automática de texto.
- Resumen automático de texto.
- Traducción automática.
- Generación de crucigramas.

3.3.2 PROTÉGÉ

Protégé es un entorno gráfico para la creación y edición de ontologías y bases de conocimiento.

La última versión, la 4.2, ya contiene ciertos plug-ins que las anteriores versiones no tenían. Es extensible e independiente de la plataforma en la que se trate puesto que está desarrollado en Java. Además, hace un fuerte uso de *Swing* para crear una interfaz de usuario compleja. Puede funcionar con java versión 6 y superior.

Como se describía anteriormente esta herramienta posee un entorno gráfico que es intuitivo y fácil de usar y tiene una arquitectura extensible a base de plug-ins.

Los proyectos creados mediante Protégé se pueden registrar de manera que generen una base de datos. Este software puede ser compatible con cualquier base de datos que utilice un controlador JDBC 1.0. Además, también se pueden exportar en formato RDF Schema o en OWL.

Permite la definición de clases, subclases, propiedades de las clases, rangos, restricciones, etc.

Otras características que ofrece este software son las siguientes:

- Crear instancias de las clases, llamadas individuales, que forman una base de datos que respeta la estructura de la ontología.
- Creación de formularios para facilitar la interacción de los usuarios con la base de datos.
- Creación de consultas automatizadas de la base de datos.

Los desarrolladores de este software ofrecen tutoriales en su web como manuales de usuario, ejemplos paso a paso, etc., que ayudan a comprender el funcionamiento del mismo.

3.3.3 SPARQL

SPARQL cuyo acrónimo es **P**rotocol **a**nd **R**DF **Q**uery **L**anguage, es un lenguaje de consultas para RDF. Es decir, es un lenguaje de consultas para bases de datos que recupera y manipula información guardada en formato RDF.

Fue estandarizado por el grupo DAWG del W3C y se considera una de las tecnologías clave de la web semántica. La última versión, la 1.1, es de Marzo de 2013.

Las consultas de SPARQL consisten en patrones triples, conjunciones, disjunciones y patrones opcionales.

Para poder realizar consultas usando SPARQL se han desarrollado unas interfaces que pueden ser usadas desde programas de aplicación o

directamente por personas y se denominan "**SPARQL endpoints**". Su función es aceptar y validar las consulta y devolver el resultado de las mismas. Para su uso por personas, un SPARQL endpoint puede ser una aplicación instalada de forma local o una aplicación basada en Web que podemos utilizar desde nuestro navegador, como la que ofrece DBpedia. Para uso desde programas, disponemos de un conjunto de API destinados a tal efecto.

Tiene cuatro variaciones de consultas para diferentes fines:

- **SELECT**: se usa para extraer valores sin tratar de un endpoint. Los resultados se devuelven en formato tabla.
- **CONSTRUCT**: se usa para extraer información de un endpoint y transformar los resultados en un RDF válido.
- **ASK**: usado para obtener un resultado Verdadero/Falso para una consulta a un endpoint.
- **DESCRIBE**: usado para extraer un grafo RDF del endpoint. La información que se muestre dependerá del endpoint y lo que el desarrollador haya considerado como información útil.

Cada una de estas consultas tiene un bloque WHERE para realizar restricciones. En el caso de la consulta DESCRIBE es opcional.

Lo siguiente es un ejemplo de consulta en el que la pregunta que se hace es: "¿Cuáles son las capitales de los países de África?":

```
PREFIX abc: <http://example.com/exampleOntology#>
SELECT ?capital ?country
WHERE {
  ?x abc:cityname ?capital ;
     abc:isCapitalOf ?y .
  ?y abc:countryname ?country ;
     abc:isInContinent abc:Africa .
}
```

Ilustración 8: SPARQL - Ejemplo SPARQL

Las variables se indican con "?" o "\$" delante. Enlaces para ?capital y ?country serán devueltos como información.

El procesador de consultas de SPARQL buscará sets de tripletes que casen con los patrones marcados en la consulta, enlazando las variables de la consulta con las partes correspondientes de cada triplete.

Para realizar las consultas más concisas, SPARQL permite la definición de prefijos y URIs base de manera similar a Turtle. Por ejemplo, en la consulta anterior, el prefijo "abc" se atribuye a "http://example.com/exampleOntology#".

4. Wiki Semántica

4.1 Qué es

Una wiki semántica es como una wiki cuya estructura representa el conocimiento de manera que se facilite la inferencia a partir de ese conocimiento dentro de sus páginas.

Las wikis semánticas aparecieron por primera vez tanto conceptualmente como en software en la década de 2000. El software que lo soporta se implementó de manera más seria a partir de 2005.

Así como una wiki normal contiene texto estructurado e hipervínculos sin tipo, la información de la wiki semántica puede ser consultada como una base de datos ya que se encuentra dentro de las páginas y las relaciones entre las mismas.

El modelo de conocimiento que se encuentra en una wiki semántica suele estar disponible en un lenguaje formal para que de esta manera las máquinas puedan procesarlo en un modelo entidad-relación o base de datos relacional.

La notación formal puede ser incluida en las propias páginas por los usuarios o puede ser derivada de las páginas o nombres de página o el método enlazado. Esto permite a las máquinas calcular nuevos hechos a partir de lo representado en el modelo de conocimiento.

Para entender mejor este nuevo concepto se plantea el siguiente ejemplo:

Se tiene una wiki semántica dedicada a los vehículos. Por ejemplo, el tipo más básico de los datos de una página sería que un coche es un tipo de automóvil (como lo podría ser una camioneta, autobús, furgoneta, etc.), lo que es conocido como una relación de herencia. La página para un coche contendría, además de información estándar, algunos datos semánticos legibles o al menos intuitivos por máquinas. La wiki sería capaz de generar automáticamente una lista de automóviles, simplemente listando todas las páginas que se han etiquetado como tipo de automóvil. Otras etiquetas semánticas en la página del coche podrían indicar otros datos acerca de los coches, incluyendo sus colores y nº de puertas posibles, cilindradas...

A lo largo del tiempo se han desarrollado diferentes wikis semánticas. Éstas son algunas de ellas:

- *Semantic Media Wiki*: se trata de una extensión de MediaWiki. MediaWiki es un motor de wikis programado en PHP. Este motor es usado por la Wikipedia. Semantic MediaWiki extiende el lenguaje de anotaciones de MediaWiki para expresar las relaciones entre los enlaces (páginas) y los atributos.
- *OntoWiki*: es una aplicación de wiki semántica libre, de código abierto, destinado a servir como un editor de ontologías y un sistema

de adquisición de conocimientos. Está escrita en PHP y basada en formularios en vez de en sintaxis por lo que trata de ocultar gran parte de la complejidad de los formalismos de representación del conocimiento de los usuarios.

- *JSPWiki*: es un motor de wiki basado en tecnología Java Server, permite ser extendido por plugins, crear formularios que pueden ser usados en cualquier página wiki.
- *IkeWiki*: es una wiki semántica desarrollada completamente en Java. Utiliza OWL como lenguaje de representación de ontologías y SPARQL como lenguaje de consultas.
- *SweetWiki*: es una web semántica diseñada alrededor de una ontología que describe la estructura de la wiki. Se usó OWL Lite para definir los conceptos, propiedades y relaciones utilizadas en la wiki. Los metadatos se encuentran embebidos en las páginas de la wiki. Utiliza SPARQL como lenguaje de consulta junto al motor de inferencia CORESE. Está desarrollada en Java.

4.2 Semantic MediaWiki

4.2.1 Definición

Semantic MediaWiki (SMW) es una extensión de MediaWiki que permite buscar, organizar, crear tags, listar, evaluar y compartir el contenido de la wiki. Como se decía antes, una wiki tradicional contiene textos que un ordenador no puede entender ni evaluar. Lo que ofrece SMW es poder crear anotaciones semánticas que permite a la wiki funcionar como una base de datos colaborativa.

Fue creada en 2005 y actualmente tiene diez desarrolladores y se usa en cientos de sitios.

Además, se han creado un gran número de extensiones que permiten ampliar la habilidad de editar, mostrar y buscar entre la información guardada en SMW.

4.2.2 Características

Estas son las características que ofrece:

- **Listas generadas automáticamente.** Las wikis tienden a tener muchas listas agregadas. Estas listas están destinadas a ser erróneas puesto que deben ser actualizadas manualmente. En SMW estas listas se generan automáticamente, están siempre actualizadas y pueden ser fácilmente modificadas para obtener más información.

- **Visualización de la información.** Existen varios formatos posibles para visualizar la información gracias a extensiones como Semantic Result Formats y Semantic Maps. Éstas, por ejemplo permiten que los datos se vean en calendarios, timelines, grafos y mapas. Además de estas extensiones existen otras que ofrecen otros formatos.
- **Estructura de datos mejorada.** En MediWiki se tiende a hacer bastante uso de categorías para estructurar datos. En SMW las categorías, en según qué contexto, pueden ser reemplazadas por valores semánticos reduciendo la necesidad de tener un sistema complejo de clasificación. Además, si se usan "semantic templates" para almacenar marcas semánticas la wiki puede ganar una estructura de datos sólida.
- **Búsqueda de información.** Usuarios pueden buscar información específica creando sus propias queries usando extensiones específicas como Halo o Semantic Drilldown.
- **Reutilización externa.** Una vez la información está creada en la SMW no tiene porqué quedarse en la wiki, puede ser fácilmente exportable vía formatos como CSV, JSON y RDF. Esto permite a la SMW servir como fuente de información a otras aplicaciones o servir como base de datos relacional en el caso de uso en empresas.
- **Integración y mezcla de datos.** La información contenida en una instalación de SMW puede ser importada, exportada o transferida, mediante las extensiones adecuadas, y no tiene porqué estar aislada. Se puede integrar con información externa y relacionarse con otra información semántica en la wiki.

4.2.3 Funcionalidades

Algunas de las funcionalidades que ofrece SMW son:

- **Interfaces de navegación**

SMW provee varias interfaces simples de navegación de datos contenidos en la wiki. Adicionalmente hay dos maneras de navegar a través de los datos de la wiki: usando una o más extensiones que ofrecen la navegación a través de los datos de SMW y usar una herramienta externa que se popula con los datos de la wiki vía RDF.

Por ejemplo, se utiliza el Factbox para sumarizar la información semántica introducida en una página, la navegación semántica que ofrece una interfaz simple de búsqueda de datos semánticos, búsquedas por propiedades, tipos y valores.

- **Búsqueda semántica**

Semantic MediaWiki incluye un lenguaje de consultas fácil de usar que posibilita al usuario acceder al conocimiento de la wiki.

- **Editar**

Facilidad para introducir anotaciones semánticas al editar una página de la misma manera que en MediaWiki.

- **Web Semántica**

A pesar de que Semantic MediaWiki está diseñado para usarse sin ningún conocimiento técnico adicional, está estrechamente relacionada con tecnología de Web Semántica. De esta manera permite a la wiki compartir su conocimiento con aplicaciones externas codificándolo en el formato estándar OWL/RDF.

4.2.4 Cómo almacena la información

Semantic MediaWiki almacena sus datos a través de diez tablas adicionales a las que usa MediaWiki (que normalmente es una base de datos MySQL). SMV puede almacenar sus datos adicionalmente en un almacén de tripletes RDF, como 4store y Virtuoso, aunque en este tipo de casos la base de datos estándar también se usa.

4.2.5 Cómo se añade información semántica

Es posible importar y reusar vocabulario que pertenezca a documentos de la Web Semántica o a cualquier estándar asociando los elementos del vocabulario con términos de la wiki.

4.2.6 Cómo importar ontologías

Según la versión que se tenga de SMW existirá una página que facilita la importación aunque no se recomienda porque es una versión beta. También se ofrece la posibilidad de realizarlo mediante un script siguiendo ciertos pasos.

Por último también existen diversas extensiones que facilitan este proceso con varias funcionalidades extra según la base de datos que se utilice y la manera de guardar los datos. Normalmente un documento OWL/RDF es usado.

5. Aplicación práctica

5.1 Diseño de una ontología

A continuación se podrán leer una serie de capítulos que explican cómo se ha llevado a cabo el diseño de una ontología utilizando el entorno Protégé versión 4.2.

5.1.2 De qué trata

La ontología que se ha elegido trata sobre series de televisión, en particular, la información que contendrá o a la que se ha orientado es a las series estadounidenses.

Algunas de las preguntas a las que debería responder serían del tipo:

- ¿De qué año es la serie?
- ¿Qué personajes tiene la serie?
- ¿En qué lugar transcurre la serie?
- ¿Quién la dirige?
- ¿Quién la produce?
- ¿Quién la interpreta?
- ¿Qué intérpretes interpretan a los personajes?
- ¿Qué temporadas tiene la serie?
- ¿Qué capítulos tiene la serie?
- ¿Qué capítulos tiene la temporada?

5.1.3 Pasos que se han seguido

En primer lugar, para empezar desde cero la ontología se siguieron los pasos que se comenta en el documento de Natalya F. Noy y Deborah L. McGuinness, "*Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología*" como se describe a continuación:

1. Se determinó el dominio y alcance de la ontología, como bien se dice anteriormente a las series de televisión estadounidenses. Además también se pensó en las preguntas a las que debería responder que también se pueden consultar en el punto anterior.
2. Se pensó en si cabría la posibilidad de reutilizar alguna ontología existente. Por ejemplo, para la pregunta *¿En qué lugar transcurre la serie?*, quizá, se podría haber hecho uso de alguna ontología relacionada con países y ciudades y más concretamente de Estados Unidos. Al pensarlo más detenidamente se desestimó puesto que muchas de las series ocurren en lugares ficticios que seguramente no aparecerían y que además quizá la ontología a reutilizar fuese más de lo que se necesitaba. En un principio se pensó que existiría una clase para este concepto en un primer momento como CIUDAD pero

finalmente quedaba más claro y cercano a lo que se buscaba como LUGAR.

Para el concepto general de la ontología, de dominio series de televisión se buscaron ontologías parecidas. Se encontró en Semantic Web (<http://semanticweb.org/wiki/Ontology>) una ontología relacionada con la música, que describía los conceptos relacionados con la música como artistas, albums, tracks, etc. Se miró por si la estructura podía ayudar puesto que se semejaba un poco a la finalidad de las series aunque tampoco del todo.

Se hizo una búsqueda más profunda en Semantic Web mediante el buscador Swoogle. Se encontró alguna relacionada con el cine como: <http://labotalc.loria.fr/~kasimir/downloads/owl/cinema.owl> aunque sirvió para hacerse una idea en cuanto al uso de propiedades no se reutilizó pues se alejaba del tema en cuestión ya que se centraba en algunos actores y películas concretos.

3. Se enumeraron una serie de términos importantes relativos al tema pensado, que más tarde se convertirían en las clases, como: serie, capítulos, títulos, duración, género, tipo, actores, temporadas, productora, ciudad, año...

Para este punto también se hizo uso de una herramienta que se ha explicado en capítulos anteriores, WordNet, para tratar de encontrar términos apropiados y relaciones adecuadas. Se hizo una búsqueda en inglés de los términos anteriores, se empezó por "TV Show" y sus "Coordinate Terms" obteniendo lo siguiente:

```
1 sense of tv show

Sense 1
television program, TV program, television show, TV show -- (a program broadcast by television)
-> broadcast, program, programme -- (a radio or television show; "did you see his program last night?")
=> news program, news show, news -- (a program devoted to news; "we watch the 7 o'clock news every night")
=> rerun -- (a program that is broadcast again; "she likes to watch 'I love Lucy' reruns")
=> talk show, chat show -- (a program during which well-known people discuss a topic or answer questions telephoned in by the audience; "in England they call a talk show a chat show")
=> television program, TV program, television show, TV show -- (a program broadcast by television)
=> game show, giveaway -- (a television or radio program in which contestants compete for awards)
=> serial, series -- (a serialized set of programs; "a comedy series"; "the Masterworks concert series")
=> episode, installment, instalment -- (a part of a broadcast serial)
=> sustaining program -- (a program without a commercial sponsor)
```

Ilustración 9: Aplicación práctica – Wordnet, TV Show y Coordinates

En esta búsqueda se vio que como término coordinado se tenía "serial , series" que es lo más parecido a la posible traducción que se

querría utilizar de TV Show, Serie, para la jerarquía de clases. También se vio que existía otro término adecuado para la ontología, "episode, installment, instalment", que serviría como capítulo. A parte de esto no se encontró nada más útil.

A continuación se hizo una búsqueda del mismo término para sus "Hypernyms" obteniendo lo siguiente:

1 sense of tv show

Sense 1

television program, TV program, television show, **TV show** -- (a program broadcast by television)
=> broadcast, program, programme -- (a radio or television show; "did you see his program last night?")
=> show -- (a social event involving a public performance or entertainment; "they wanted to see some of the shows on Broadway")
=> social event -- (an event characteristic of persons forming groups)
=> event -- (something that happens at a given place and time)
=> psychological feature -- (a feature of the mental life of a living organism)
=> abstraction -- (a general concept formed by extracting common features from specific examples)
=> abstract entity -- (an entity that exists only abstractly)
=> entity -- (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Ilustración 10: Aplicación práctica – Wordnet, TV Show e Hypernyms

Para las superclases el resultado se alejaba bastante del sentido de lo que se pretendía desarrollar. Finalmente, para este mismo término, se hizo la consulta para los Hyponyms:

1 sense of tv show

Sense 1

television program, TV program, television show, **TV show** -- (a program broadcast by television)
=> colorcast, colourcast -- (a television program that is broadcast in color)
=> pilot program, pilot film, pilot -- (a program exemplifying a contemplated series; intended to attract sponsors)
=> situation comedy, sitcom -- (a humorous television program based on situations that could arise in everyday life)
=> special -- (a television production that features a particular person or work or topic; "the last of a series of BBC specials on Iran is being shown tonight")

Ilustración 11: Aplicación práctica – Wordnet, TV Show y Hyponyms

Para las subclases tampoco se obtuvo un resultado adecuado. Se hizo otra búsqueda con los sinónimos de TV Show, pero siguió sin ser satisfactorio.

Finalmente, se decidió realizar la ontología libre, en base a la intuición, sobre el tema en cuestión, relativamente conocido, para centrarla más a la visión inicial.

Para el término género, se habían declarado las subclases tipo: aventura, acción, drama, etc. Y a medida que se fue adelantando en

el desarrollo estaba más claro que esas subclases debían desaparecer puesto que no coincidían con el propósito de la ontología.

El año se ha decidido crearlo como clase y no como un atributo temporal de la clase SERIE porque se quiere dar importancia al año de inicio de la serie y poder buscar por este concepto.

4. Se decidió realizar una combinación entre *Top-down* y *Bottom-up* para realizar la jerarquía de clases.
5. Se definieron las propiedades de las clases que se muestran más abajo.
6. Se definieron las facetas de los slots, el tipo de valor que tendrían, valores admitidos y la cardinalidad.
7. Se crearon instancias de cada clase y se nutrió con cada una de las propiedades.

Una vez se tenía desarrollada la ontología en el Protégé e introducido una serie de instancias o individuales se han hecho una serie de consultas sencillas con SPARQL, que ya viene integrado en esta versión del entorno, y los resultados eran los esperados.

5.1.5 Estructura jerárquica de clases y propiedades

A continuación se presenta una imagen de la jerarquía de clases de los conceptos antes mencionados:

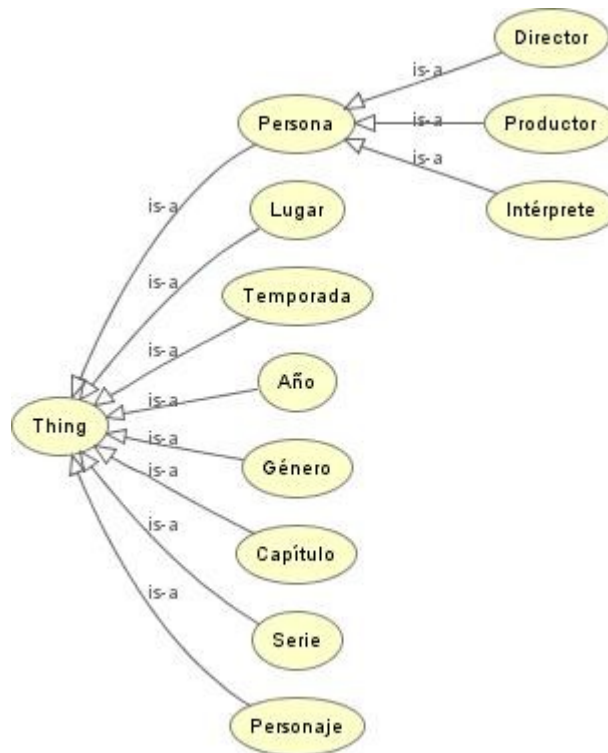


Ilustración 12: Aplicación práctica – Jerarquía de clases Series TV

Para poder obtener este grafo ha sido necesario instalar el plug-in *graphviz* y cambiar el path en el protege para que se nutriera del mismo y no del que viene por defecto.

Se han declarado las siguientes propiedades de objeto:

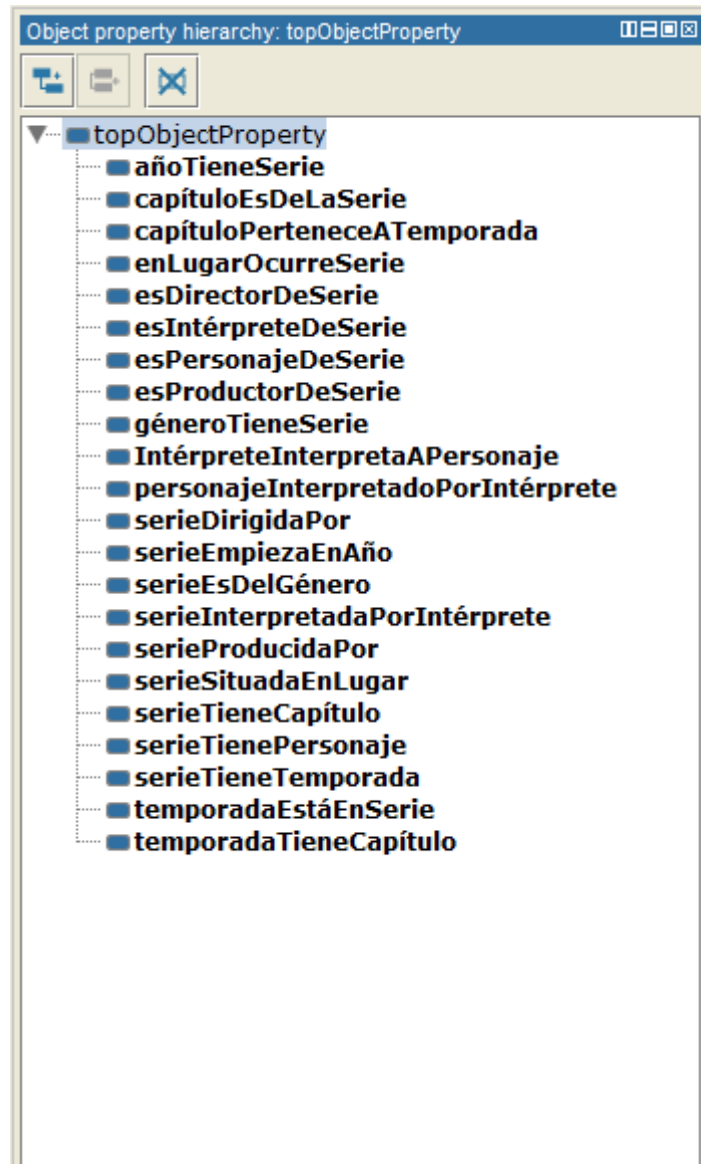


Ilustración 13: Aplicación práctica – Propiedades de Series TV

Y por último se han declarado las siguientes propiedades de datos:

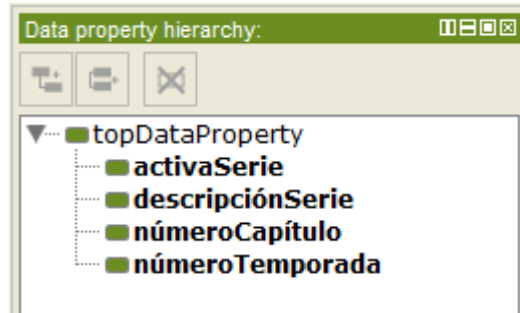


Ilustración 14: Aplicación práctica – Propiedades de datos de Series TV

- activaSerie: es un booleano que indica si la serie se sigue produciendo o no.
- descripciónSerie: es un string que proporciona una sinopsis de la serie.
- númeroCapítulo: es un entero que indica de qué número de capítulo se trata.
- númeroTemporada: es un entero que indica de qué temporada se trata.

5.2 Aplicación sobre wiki semántica

Para realizar la carga de la ontología en una wiki semántica se ha elegido la Semantic MediaWiki más arriba descrita puesto que parece bastante completa, consistente y conocida. Además hay muchísima documentación referente a la instalación y uso, tanto de administrador como usuario.

Para llevar a cabo este proceso se han seguido los siguientes pasos:

1. Instalación de MediaWiki.

Como se ha explicado, Semantic MediaWiki es una extensión de MediaWiki por lo que es necesario instalarla en primer lugar siguiendo los pasos del documento: <http://www.mediawiki.org/wiki/Installation>.

Durante la instalación se permite la creación de un usuario con rol de administrador y dar nombre a la Wiki, en este caso se le ha puesto "Series TV".

Las principales especificaciones que pide la versión instalada de MediaWiki, que es la 1.19, son :

- PHP 5.2+ (except for version 5.3.1)
- MySQL 5.0.2+ (except for versions 5.1.49 to 5.1.52)
- Apache 2

Después de la instalación se ve la wiki como se muestra en el pantallazo:



Ilustración 15: Wiki Semántica - Página principal después de instalación

2. Instalación de Semantic MediaWiki.

Siguiendo el siguiente manual: <http://semantic-mediawiki.org/wiki/Help:Installation>, se consigue instalar la extensión de MediaWiki, Semantic MediaWiki.

Se añaden algunas páginas especiales para el uso semántico de la web:



Ilustración 16: Wiki Semántica - Páginas especiales para SMW después de instalación

3. Instalación de la extensión RDFIO.

Después de investigar un poco entre la documentación existente se vieron varias alternativas para poder importar una ontología:

1. **Página especial integrada en la Semantic MediaWiki.** Tal y como se describe en este documento: [44](http://semantic-</div><div data-bbox=)

mediawiki.org/wiki/Help:Ontology_import, existe una página especial para realizar la importación de documentos en OWL-DL en RDF-serialization estrictamente. Además, también se exige que si se quiere inferencia se debe declarar explícitamente para que se importe con unas normas. Las relaciones de tipo inverso, complemento y unión tampoco serán importadas.

Esta técnica, según se especifica en el documento no funciona muy bien y está obsoleta y sólo en versiones antiguas de Semantic MediaWiki.

2. **Uso de scripts para actualizar los datos de la wiki.** En el siguiente documento: http://semantic-mediawiki.org/wiki/Help:Data_import_with_a_script se describe, con un ejemplo, cómo subir mediante librerías en PHP o Python con pyWikipediaBot framework y un script ontologías de un tamaño relativo.
3. **Haciendo uso de extensiones creadas a propósito.** En la página: http://semantic-mediawiki.org/wiki/Help:SMW_extensions se describen prácticamente todas las extensiones que tiene SMW. Entre ellas, se han mirado especialmente las que usaban almacenes de tripletes RDF.

Finalmente se ha optado por la opción 3, se ha instalado la extensión RDFIO para poder importar el documento creado a partir de la ontología desarrollada en el Protégé. La razón es que la primera opción está obsoleta y la segunda, después de comparar, parecía más complicada.

RDFIO extiende la funcionalidad de Semantic MediaWiki de importar y exportar RDF. Técnicamente RDFIO implementa el almacén de tripletes PHP/MySQL ofrecido por la librería ARC2.

También ofrece un endpoint SPARQL para poder realizar consultas con la URI original del archivo importado.

También, se ha elegido porque trabaja con PHP y MySQL que son las especificaciones de la wiki a pesar de que todavía se encuentre en proceso de desarrollo. Es por eso que puede aparecer un error pertinente al endpoint del SPARQL en la parte superior de la pantalla. Puesto que Semantic MediaWiki ya ofrece un lenguaje de consultas bastante interesante también se ha decidido investigarlo.

Para instalar la extensión se han seguido los pasos del documento: <http://www.mediawiki.org/wiki/Extension:RDFIO>.

Una vez instalada la wiki ha quedado con las siguientes páginas especiales:

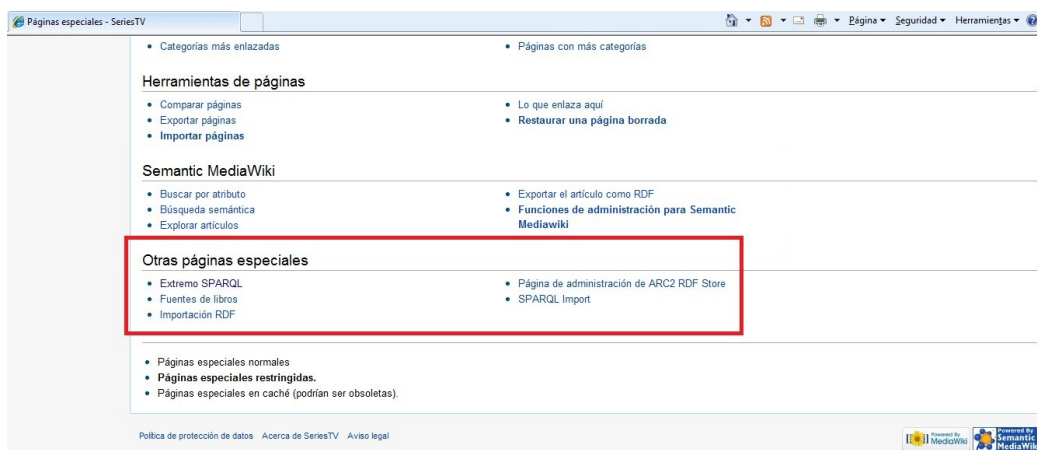


Ilustración 17: Wiki Semántica - Páginas especiales para RDFIO después de instalación

4. Resultado de la carga del archivo XML/RDF.

Para realizar la carga del archivo se ha utilizado la funcionalidad de la extensión RDFIO, Importación RDF, para el archivo de la ontología XML/RDF que se puede ver adjunto en los anexos.

Después de la carga se muestra lo siguiente:

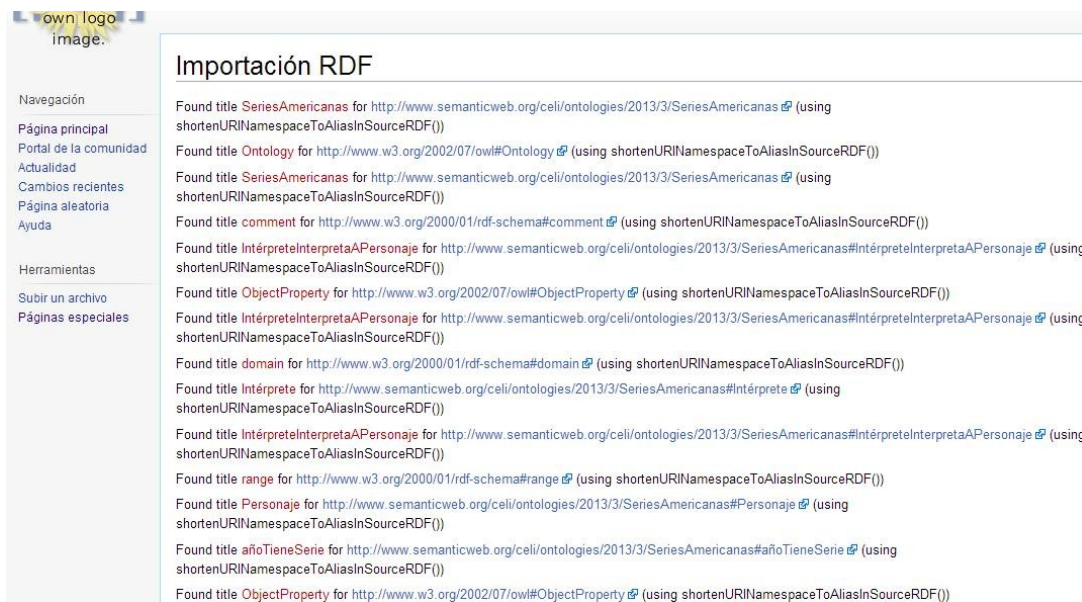


Ilustración 18: Wiki Semántica - Importación RDF (Inicio) Ontología Series TV

Y finaliza así:



Ilustración 19: Wiki Semántica - Importación RDF (final) Ontología Series TV

5. Comprobar datos cargados.

Para poder ver los datos cargados junto con la ontología existen varias maneras:

1. Acceder desde las opciones que ofrece MediaWiki.

Accediendo a las páginas especiales de Categorías y Propiedades pueden verse las clases y las propiedades de objeto. Se destacan estas dos aunque existen más como la de listar los tipos de datos.

Listado de categorías:

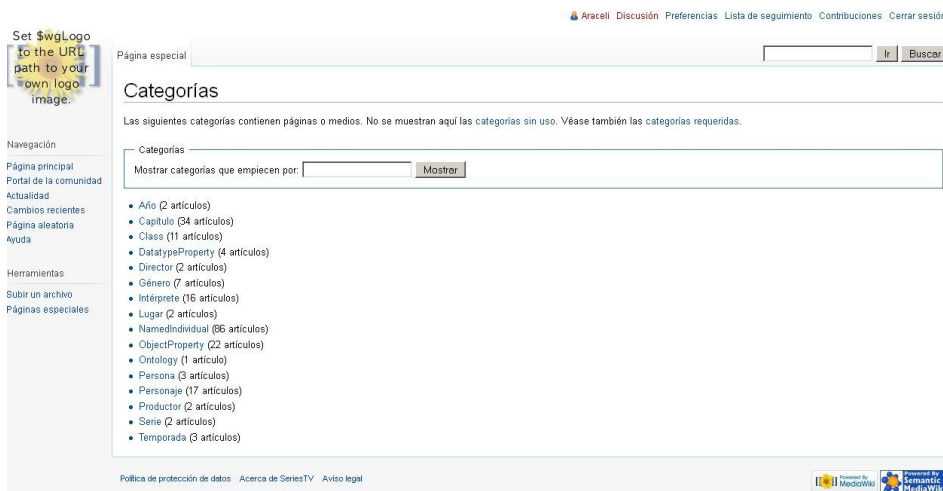


Ilustración 20: Wiki Semántica - Listado categorías Series TV MediaWiki

Si accedemos a alguna de las categorías nos ofrece todo lo que está relacionado con esta categoría, por ejemplo:

Ilustración 21: Wiki Semántica - Página categoría Capítulo Series TV

Listado de propiedades:

Ilustración 22: Wiki Semántica - Listado de propiedades MediaWiki

Si accedemos a alguna de ellas también nos enseñará todo los artículos que están relacionados por esa propiedad como se muestra en la imagen más abajo, con la particularidad de que ya ofrece funcionalidades de SMW a través del circulito y la

lupa que aparecen a la derecha de cada elemento y que ya se explica en el siguiente punto:

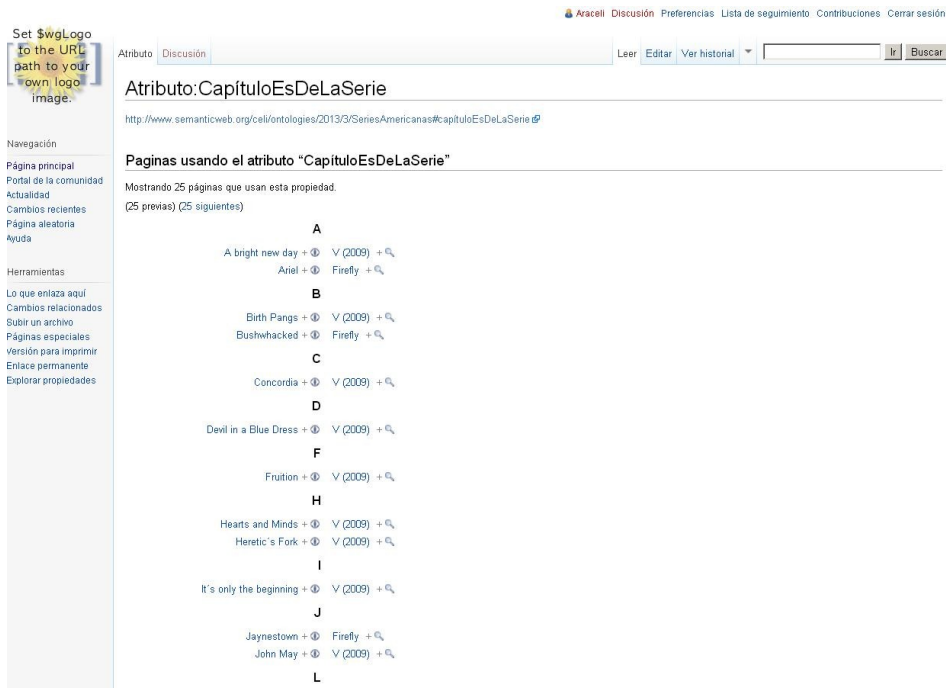


Ilustración 23: Wiki Semántica - Propiedad CapítuloEsDeLaSerie de SeriesTV MediaWiki

2. Acceder desde las opciones que ofrece SemanticMediaWiki.

Las opciones que ofrece SMW son parecidas a las de la MediaWiki pero están más orientadas a las anotaciones semánticas y por lo tanto el resultado es más completo.

Se puede **buscar por atributo** (propiedades de tipo y de objeto) y un valor dados, como por ejemplo:

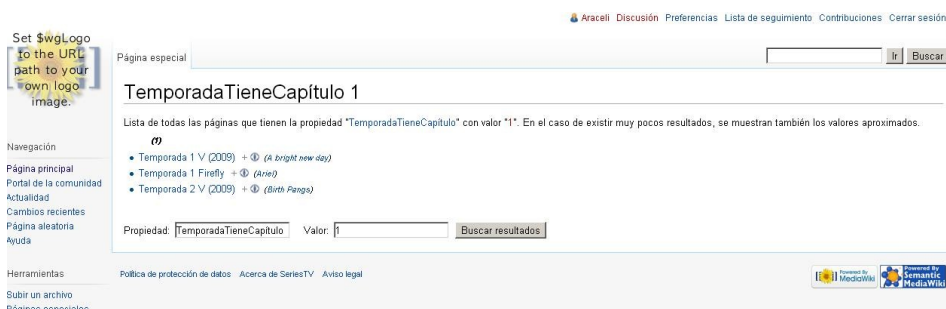


Ilustración 24: Wiki Semántica - Ejemplo opción buscar por atributo en SMW

Si en las cajas que aparecen como Propiedad y Valor (en la parte inferior) se ingresa el nombre de una propiedad y el valor que se está buscando, como por ejemplo, en este caso se buscan temporadas que tengan el capítulo 1, y en el resultado

nos aparecen todas las temporadas que tengan el capítulo 1, la serie a la que pertenece y el nombre del capítulo y que están relacionados por esta propiedad.

La búsqueda por atributo también se puede realizar, una vez hecha cualquier otra consulta y si aparece, a través de la lupa que aparece al lado del término.

Se puede observar al lado del nombre de la temporada que aparece un pequeño símbolo redondo, como un ojo de gato, que indica que se puede hacer una exploración de ese artículo como se muestra:

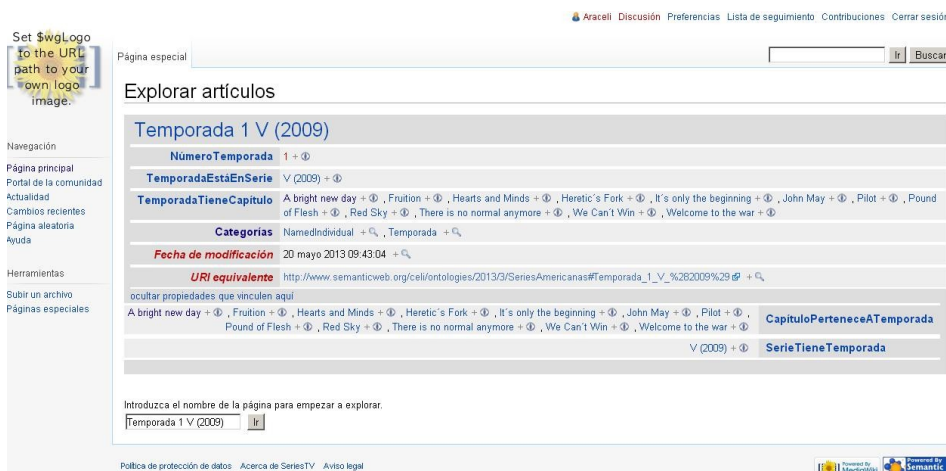


Ilustración 25: Wiki Semántica - Ejemplo opción explorar artículos en SMW(1)

También se puede **buscar directamente por artículos**, es decir, lo que diríamos *Individuales o Instancias* que se han introducido para la ontología como se muestra en la imagen:



Ilustración 26: Wiki Semántica - Ejemplo opción explorar artículos en SMW(2)

Y al clicar sobre buscar nos mostrará la siguiente imagen:

The screenshot shows a detailed view of the 'Firefly' article in a Semantic Wiki. The page is titled 'Explorar artículos' and lists various properties of the series. The properties include:

- ActivaSerie:** False + ⌵
- DescripciónSerie:** Five hundred years in the future, a renegade crew aboard a small spacecraft tries to survive as they travel the unknown parts of the galaxy and evade warring factions as well as authority agents out to get them. + ⌵
- SerieDirigidaPor:** Joss Whedon + ⌵
- SerieEmpiezaEnAño:** 2002 + ⌵
- SerieEsDelGénero:** Aventura + ⌵, Ciencia Ficción + ⌵, Drama + ⌵
- SerieInterpretadaPorInterprete:** Adam Baldwin + ⌵, Alan Tudyk + ⌵, Gina Torres + ⌵, Jewel Staitte + ⌵, Morena Baccarin + ⌵, Nathan Fillion + ⌵, Ron Glass + ⌵, Sean Maher + ⌵, Summer Glau + ⌵
- SerieProducidaPor:** Joss Whedon + ⌵
- SerieSituadaEnLugar:** Firefly + ⌵
- SerieTieneCapitulo:** Ariel + ⌵, Bushwhacked + ⌵, Jaynestown + ⌵, Objects in Space + ⌵, Our Mrs. Reynolds + ⌵, Out of Gas + ⌵, Safe + ⌵, Serenity (1) + ⌵, Serenity (2) + ⌵, Shindig + ⌵, The Train Job + ⌵, War Stories + ⌵
- SerieTienePersonaje:** Dr. Simon Tam + ⌵, Inara + ⌵, Jayne Corr + ⌵, Kaylee Frye + ⌵, Malcolm Mal Reynolds + ⌵, Ryver Tam + ⌵, The Shepherd + ⌵, Wash + ⌵, Zoe + ⌵
- SerieTieneTemporada:** Temporada 1 Firefly + ⌵
- Categorías:** NamedIndividual + ⌵, Serie + ⌵
- Fecha de modificación:** 20 mayo 2013 09:43:03 + ⌵
- URI equivalente:** http://www.semanticweb.org/cel/ontologies/2013/3/SeriesAmericanas#Firefly + ⌵

Below the properties, there is a section for 'ocultar propiedades que vinculen aquí' and a list of related articles with their properties.

Ilustración 27: Wiki Semántica - Ejemplo opción explorar artículos en SMW(3)

En este punto ya se muestran todas las propiedades relacionadas con este artículo y a su vez las propiedades que pueden vincular a ese artículo.

Por último también se ofrece la **búsqueda semántica** mediante su propio lenguaje de consultas tal y como se muestra en la siguiente imagen:

The screenshot shows the 'Búsqueda semántica' page in a Semantic Wiki. The page has a search bar with the query: '[[Category:Personaje]] [[!esPersonajeDeSerie:Firefly]]'. Below the search bar, there are several options for formatting and classification. The 'Formatar como' section has a dropdown menu set to 'Tabla ancha (por defecto)'. The 'Clasificación' section has a text input field with the value '[Agregar condición de orden]'. The 'Otras opciones' section has several input fields and checkboxes for search parameters:

- limit:** La cantidad máxima de resultados a devolver
- offset:** El desplazamiento del primer resultado
- link:** all (Mostrar los valores como vínculos)
- sort:** Propiedad a partir de la que ordenar la petición
- order:** Orden de ordenación de la consulta
- headers:** Muestra los nombres de encabezados/propiedades
- mainlabel:** La etiqueta a asignar al nombre de la página principal
- intro:** El texto a mostrar antes de los resultados de la consulta, si los hubiere
- outro:** El texto a mostrar después de los resultados de la consulta, si los hubiere
- searchlabel:** … siguientes resultados
- default:** El texto a mostrar si no hay resultados de consulta
- class:** sortable wikitables smwtable (Una clase CSS adicional a establecer para la tabla)

At the bottom of the page, there is a 'Buscar resultados' button and a link to 'Ocultar preguntas | Mostrar código embebido | Consultando ayuda'.

Ilustración 28: Wiki Semántica - Ejemplo opción búsqueda semántica en SMW(1)

En esta consulta se piden los personajes en la serie Firefly cuyo resultado es:

The screenshot shows the configuration options for a Semantic MediaWiki search query. The options are:

- mainlabel:** La etiqueta a asignar al nombre de la página principal
- intro:** El texto a mostrar antes de los resultados de la consulta, si los hubiere
- outro:** El texto a mostrar después de los resultados de la consulta, si los hubiere
- searchlabel:** … siguientes resultados
- default:** El texto a mostrar si no hay resultados de consulta
- class:** sortable wikitable smwtable

Below the configuration, the search results are displayed. The search query is `[[es:PersonajeDeSerie::Firefly]]`. The results are shown in a table with the following entries:

Dr. Simon Tam
Inara
Jayne Corr
Kaylee Frye
Malcolm Mal Reynolds
Ryker Tam
The Shepherd
Wash
Zoe

Ilustración 29: Wiki Semántica - Ejemplo opción explorar artículos en SMW(2)

La búsqueda semántica ofrece muchísimas posibilidades de búsqueda como de muestra de resultados. Para más información sobre el lenguaje de consultas que usa se ha buscado en:

http://semantic-mediawiki.org/wiki/Help:Semantic_search.

Una de las posibilidades y bastante interesante que ofrece es ver el código embebido para mostrar la consulta antes hecha:

The screenshot shows the embedded code for the search query. The code is:

```
{#ask: [[Category:Personaje]] [[es:PersonajeDeSerie::Firefly]]
|?#
|format=broadtable
|link=all
|headers=show
|searchlabel=_siguientes resultados
```

Below the code, the search results are displayed. The search query is `[[es:PersonajeDeSerie::Firefly]]`. The results are shown in a table with the following entries:

Dr. Simon Tam
Inara
Jayne Corr
Kaylee Frye
Malcolm Mal Reynolds
Ryker Tam
The Shepherd
Wash
Zoe

Ilustración 30: Wiki Semántica - Ejemplo opción explorar artículos en SMW(3)

Se puede copiar :

```
{{#ask: [[Category:Personaje]]
[[esPersonajeDeSerie::Firefly]]
|?#
|format=broadtable
|link=all
|headers=show
|searchlabel=... siguientes resultados
|class=sortable wikitable smwtable
}}
```

Tabla 8: Wiki Semántica - Ejemplo código búsqueda semántica

Y pegar en una página nueva para crear un artículo para la wiki:

Araceli | Discusión | Preferencias | Lista de seguimiento | Contribuciones | Cerrar sesión

Página | Discusión | Leer | Editar | Ver historial | Ir | Buscar

Personajes de Firefly

Esta página muestra los personajes de la serie Firefly mediante una consulta realizada con la Búsqueda Semántica.

Dr. Simon Tam
Inara
Jayne Cobb
Kaylee Frye
Malcolm Mal Reynolds
Ryker Tam
The Shepherd
Wash
Zoe

Esta página fue modificada por última vez el 21 may 2013, a las 08:47.
Esta página ha sido visitada 5 veces.
Política de protección de datos | Acerca de SeriesTV | Aviso legal

Powered by MediaWiki | Semantic MediaWiki

Ilustración 31: Wiki Semántica - Ejemplo página nueva con datos Series TV

6. Página principal de la Wiki Semántica, Series TV.

Se ha modificado la página principal para que quede de la siguiente manera y sea más fácil acceder a las funcionalidades requeridas para este proyecto:

Set \$wgLogo to the URL path to your own logo image.

Navegación

- Página principal
- Portal de la comunidad
- Actualidad
- Cambios recientes
- Página aleatoria
- Ayuda

Herramientas

- Lo que enlaza aquí
- Cambios relacionados
- Páginas especiales
- Versión para imprimir
- Enlace permanente
- Explorar propiedades

Página [Discusión](#) [Leer](#) [Editar](#) [Ver historial](#)

Página principal

MediaWiki ha sido instalado con éxito.

Consulta la [Guía de usuario](#) para obtener información sobre el uso del software wiki.

Contenido

- 1 Empezando
- 2 MediaWiki
- 3 Ontología Series TV (Semantic MediaWiki)
- 4 Páginas de ejemplo de Series TV
- 5 Importar RDF

Empezando [editar]

- [Lista de ajustes de configuración](#)
- [FAQ de MediaWiki](#)
- [Lista de correo de anuncios de distribución de MediaWiki](#)

MediaWiki [editar]

- [Categorías](#)
- [Propiedades](#)

Ontología Series TV (Semantic MediaWiki) [editar]

- [Buscar por atributo](#)
- [Búsqueda semántica](#)
- [Explorar artículos](#)

Páginas de ejemplo de Series TV [editar]

- [Personajes de Firefly](#)
- [Capitulos Disponibles](#)
- [Personajes Interpretados por Morena Baccarin](#)

Importar RDF [editar]

- [ARC2Admin](#)
- [RDFImport](#)
- [SPARQLEndpoint](#)

Ilustración 32: Wiki Semántica - Página principal final

El primer bloque, "Empezando", es lo que aporta MediaWiki cuando es instalada. Ofrece manuales de configuración así como un FAQ.

El segundo bloque, "MediaWiki", se ha creado para tener a mano páginas especiales de MediaWiki que pueden ser útiles a la hora de ver la información contenida y cargada. Su funcionamiento se ha explicado en el apartado anterior.

El tercer bloque, "Ontología Series TV (Semantic MediaWiki)", contiene páginas especiales de Semantic MediaWiki para poder navegar por la información semántica. Su funcionamiento se ha explicado en el apartado anterior.

El cuarto bloque, "Páginas de ejemplo de Series TV", es una página creada a partir del lenguaje de consultas propio de la wiki para mostrar información de la ontología y que se ha mostrado en el apartado anterior. A la página "Personajes de Firefly" se le ha añadido un link a otra página "Capítulos de Firefly" en la que se pueden ver los capítulos de esta serie como bien dice el título.

Personajes de Firefly

Esta página muestra los personajes de la serie Firefly mediante una consulta realizada con la Búsqueda Semántica.

Dr. Simon Tam
Inara
Jayne Corr
Kaylee Frye
Malcolm Mal Reynolds
Ryver Tam
The Shepherd
Wash
Zoe

[Ver capítulos de la serie](#)

Ilustración 33: Wiki Semántica - Ejemplo resultados semánticos 1

Clicando sobre el recuadro rojo se podrá ver lo siguiente:

Capítulos de Firefly

A continuación se muestran los capítulos de la serie Firefly

Capítulos de Firefly ↕	NúmeroCapítulo ↕
The Train Job	1
Objects in Space	10
Serenity (1)	11
Serenity (2)	12
Bushwhacked	2
Our Mrs. Reynolds	3
Jaynestown	4
Out of Gas	5
Shindig	6
Safe	7
Ariel	8
War Stories	9

Ilustración 34: Wiki Semántica - Ejemplo resultados semánticos 2

Cuyo código es el siguiente:

```
{{#ask: [[Category:Capítulo]] [[CapítuloEsDeLaSerie::Firefly]]
|?NúmeroCapítulo
|format=table
|link=subject
|sort=NúmeroCapítulo
|order=ASC
|headers=plain
|mainlabel=Capítulos de Firefly
|intro=A continuación se muestran los capítulos de la serie Firefly
|searchlabel=... siguientes resultados
|class=sortable wikitable smwtable
}}
```

Tabla 9: Wiki Semántica - Ejemplo código resultados semánticos 2

Aquí se puede ver que la información se muestra de manera diferente, es una tabla, está ordenado por el número del capítulo en orden ascendente. Además se pueden incluir texto antes de mostrar los datos obtenidos de la consulta mediante la variable "intro". También se puede elegir si las páginas existentes quieren ser referenciadas mediante un link o ninguna o sólo el sujeto de la consulta.

Otra página de ejemplo es "Capítulos disponibles". Es un listado de todos los capítulos existentes en la wiki. Esta lista sería fácil de

mantener puesto que la consulta ataca a los datos semánticas de los datos y siempre cogerá los datos nuevos. A continuación se muestra parte del resultado:

Capítulos Disponibles

Información de los capítulos disponibles:

Capítulo	Serie	Número de Capítulo	Temporada
The Train Job	Firefly	1	Temporada 1 Firefly
Objects in Space	Firefly	10	Temporada 1 Firefly
Serenity (1)	Firefly	11	Temporada 1 Firefly
Serenity (2)	Firefly	12	Temporada 1 Firefly
Bushwhacked	Firefly	2	Temporada 1 Firefly
Our Mrs. Reynolds	Firefly	3	Temporada 1 Firefly
Jaynestown	Firefly	4	Temporada 1 Firefly
Out of Gas	Firefly	5	Temporada 1 Firefly
Shindig	Firefly	6	Temporada 1 Firefly
Safe	Firefly	7	Temporada 1 Firefly
Ariel	Firefly	8	Temporada 1 Firefly
War Stories	Firefly	9	Temporada 1 Firefly
Pilot	V (2009)	1	Temporada 1 V (2009)
Hearts and Minds	V (2009)	10	Temporada 1 V (2009)
Fruition	V (2009)	11	Temporada 1 V (2009)
Red Sky	V (2009)	12	Temporada 1 V (2009)
There is no normal anymore	V (2009)	2	Temporada 1 V (2009)

Ilustración 35: Wiki Semántica - Ejemplo resultados semánticos 3

El código de esta página es el que sigue:

```

{{#ask: [[Category:Capítulo]]
|?CapítuloEsDeLaSerie=Serie
|?NúmeroCapítulo=Número de Capítulo
|?CapítuloPerteneceATemporada=Temporada
|format=table
|link=all
|sort=CapítuloEsDeLaSerie,CapítuloPerteneceATemporada,NúmeroCapítulo
|headers=show
|mainlabel=Capítulo
|intro=Información de los capítulos disponibles:
|searchlabel=\x26hellip; siguientes resultados
|class=sortable wikipable smwtable
}}
```

Tabla 10: Wiki Semántica - Ejemplo código resultados semánticos 3

Aquí se puede ver en las líneas 2,3, 4 y 9 que se puede nombrar la cabecera de cada columna de datos obtenidos como se desee. Además se puede ordenar por varias propiedades mediante la variable "sort".

Por último se ha creado la página "Personajes de Morena Baccarin" en el que se muestran los personajes interpretados por esta actriz y las series a las que pertenecen:



Ilustración 36: Wiki Semántica - Ejemplo resultados semánticos 4

Cuyo código es:

```
{ {#ask: [[Category:Personaje]] [[PersonajeInterpretadoPorIntérprete: :Morena Baccarin]]
|?EsPersonajeDeSerie=
|format=ul
|link=all
|headers=show
|intro=Personajes interpretados por Morena Baccarin y la serie a la que pertenece:
|searchlabel=\\x26hellip; siguientes resultados
|class=sortable wikitable smwtable
}}
```

Tabla 11: Wiki Semántica - Ejemplo código resultados semánticos 4

Aquí se puede ver que la información está formateada de manera diferente, está itemizada.

El quinto bloque, "Importar RDF", se han añadido las páginas especiales que ofrece la extensión RDFIO. Son las siguientes:

- ARC2Admin: es la página de administración del almacén de RDF implementado con la librería ARC2. Se utiliza la primera vez que se instala para cargar la estructura en la base de datos.
- RDFImport: es la página que se utiliza para cargar los archivos XML/RDF. Se ha mostrado en apartados anteriores.
- SPARQLEndpoint: desde esta página se pueden realizar consultas a la información mediante SPARQL. Para poder hacer las consultas sobre al información deseada, en este caso, la de la ontología es necesario conocer la URI ORIGINAL que se puede encontrar explorando los artículos e introducirla como prefijo y elegir el checkbox "Query by equivalent Uris" y

trabajaremos sobre la información deseada. Un ejemplo sobre la ontología se muestra en la siguiente imagen:

The screenshot shows a web interface for executing SPARQL queries. The query entered is:

```

PREFIX w : <http://www.semanticweb.org/celi/ontologies/2013/3/SeriesAmericanas#>
.
SELECT ?s ?o
WHERE ( ?s w:esPersonajeDeSerie ?o )
LIMIT 25
    
```

The results are displayed in a table with two columns: 's' (Series) and 'o' (Character). The results list various series and their corresponding characters, such as 'http://araceli.p.ht/w/index.php/Especial:URIResolver/Jayne_Corr' and 'http://araceli.p.ht/w/index.php/Especial:URIResolver/Firefly'.

Ilustración 37: Wiki Semántica - Ejemplo uso SPARQL para Series TV

Para poder ver todo el trabajo realizado se puede acceder vía web a la instalación realizada en: http://araceli.p.ht/w/index.php/P%C3%A1gina_principal

5.3 Problemas encontrados

En este apartado se explicarán algunos de los problemas encontrados durante la carga de la ontología en la wiki semántica.

5.3.1. Carga del archivo XML/RDF

Se buscó un hosting para poder alojar la wiki y que fuese pública. Después de instalar MediaWiki, Semantic MediaWiki y la extensión RDFIO se realizó el primer intento de carga del documento completo. Debido a que generaba más entradas en la bbdd de las que permite el hosting no se cargó todo lo necesario o por lo menos dio la impresión de que no se había cargado nada. Se probó un par de veces más sin éxito.

Una de las soluciones que se intentó fue cargar el documento por partes y parecía que sí funcionaba. Una vez se navegó por la información importada realmente se vió que se creó un popurrí de ontología, las uris se repetían,

las clases aparecían guardadas de una manera pero también de otra por lo que no se tomó como correcto.

Realmente en la primera carga del documento entero sí se cargaron datos y parece que de manera correcta pero al cortarse la carga empezó a ir mal. Sumándole que se hizo otra vez la carga por partes del documento acabó por ser un desastre.

La solución a este problema fue volverlo a instalar todo de manera local con todo lo que ello conlleva (Apache, PHP, MySQL, MediaWiki, SemanticMediaWiki, RDFIO) y probar de cargar el documento completo. Y funcionó. A continuación se hizo una copia de la base de datos, se creó una nueva en el hosting en la cual se cargó esta copia y mediante el localsettings.php se apuntó a esta nueva base de datos. Se probó si funcionaba bien y efectivamente la carga había ido correctamente tal y como se puede ver en la web.

5.3.2 Error en el código del Endpoint del SPARQL

Debido a que la extensión RDFIO está en constante desarrollo, la versión descargada e instalada en la wiki contiene un pequeño error de programación que realmente no afecta al funcionamiento pero sí salta el warning en la cabecera de la página.

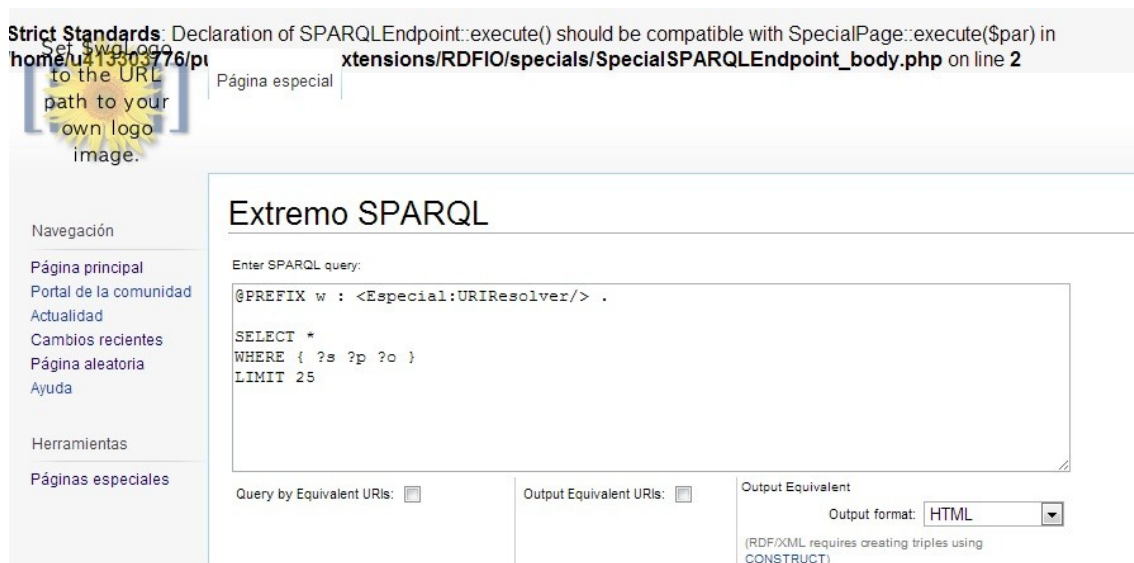


Ilustración 38: Wiki Semántica - Error SPARQL

Es posible que salga no sólo en la página para el SPARQL si no en otros momentos también.

5.3.3 CSS Wiki

Se ha detectado que en algunas ocasiones el navegador CHROME no coge bien las CSS de la wiki. Parece que es debido a que la url termina con símbolos raros "&*" (http://araceli.p.ht/w/load.php?debug=false&lang=es&modules=mediawiki.legacy.commonPrint%2Cshared%7Cskins.vector&only=styles&skin=vector&*). Esto a veces le va bien y otras no aunque normalmente consigue cargarlo correctamente.

6. Conclusiones

Una de las lecciones aprendidas al realizar este proyecto, cuyo tema central es el uso de una wiki semántica mediante una ontología, es todo el proceso de investigación que requiere previamente. Incluso antes de llevar a cabo la investigación sobre el tema en concreto, es necesario investigar lo que hay que investigar.

Esta tarea se consiguió al elaborar el plan de proyecto al que además se le añade la complicación de administrar el tiempo asignado. Gracias a este proceso, finalmente, se consigue averiguar qué temas son importantes revisar y realizar una estimación del tiempo necesario para cada uno de ellos y así llevar a cabo el proyecto de manera correcta.

Una vez realizado todo el trabajo se concluye que para llegar a conocer y comprender las wikis semánticas y las ontologías es necesario conocer otros conceptos e incluso lenguajes para crear ontologías.

Sobre el desarrollo de las ontologías se ha conocido que existen diferentes lenguajes, casi todos ellos basados en XML, para elaborarlas. Uno más primitivo llamado RDF y el más moderno y completo llamado OWL basado en RDF. Las ontologías ayudan a mantener estructurada y entendible la información deseada. Es una lección que se aprende de este sistema, a crear una estructura para los datos adecuada para distintos sistemas.

En cuanto a las wikis semánticas es increíble el hecho de que el propio agente que la trate pueda sacar la información deseada de la manera preferida sin que los datos estén estructurados exactamente para esos tipos de consultas. Esto se produce gracias a la información semántica establecida en el contenido de la wiki que hace posible la inferencia de los datos.

Uno de los problemas que se ha visto, en general, sobre la web semántica es que todo es relativamente nuevo y poco conocido. No hay un método fácil para introducir datos semánticos en la web y esto es un gran obstáculo para desarrolladores e incluso propietarios de webs que por pereza o desconocimiento no dan el paso de investigación que requiere este mundo.

En cuanto al producto obtenido a raíz de este trabajo para el cual se ha seguido el plan de proyecto establecido se cree haber llegado a la realización completa del mismo. Si bien, quizá, en alguna entrega ha sido necesario hacer uso de varios días extra (previstos en el plan) para terminarla. Esto es debido a que la falta de conocimiento ha ralentizado el proceso de creación de la ontología y la memoria.

Como línea de trabajo futuro se podría plantear mejorar la visualización de la información en la wiki. Crear índices más vistosos con la información enlazada sin que sea necesario realizar consultas a propósito en los diferentes lenguajes o buscadores que ofrecen tanto la wiki como las extensiones instaladas.

7. Glosario

Las descripciones de los términos que se muestran a continuación se encuentran ordenados según la aparición aproximada que tienen en el documento:

Web Semántica: es una web que contiene información útil para que máquinas o aplicaciones informáticas puedan hacer uso de la misma de manera correcta de la misma manera que haría una persona.

Metadatos: son datos que describen otros datos.

Etiqueta: consiste en una marca hecha en el documento que señala la porción de éste como un elemento.

Lenguaje de marcado: es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación.

HTML: HyperText Markup Language (lenguaje de marcado hipertextual). Es un lenguaje de marcado utilizado para la elaboración de páginas web que se usa para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes.

Metalinguajes: es un lenguaje que se usa para hablar acerca de otro lenguaje.

SGML: Standard Generalized Markup Language (Estándar de Lenguaje de Marcado Generalizado). Consiste en un sistema para la organización y etiquetado de documentos.

XML: eXtensible Markup Language (lenguaje de marcado extensible). Es un lenguaje de marcado que permite definir la gramática de lenguajes específicos para estructurar documentos grandes.

XML Schema: es un lenguaje para definir la estructura de los documentos XML.

DTD: Document Type Definition (Definición de Tipo de Documento). Define los tipos de elementos, atributos y entidades permitidas de un documento XML y puede expresar algunas limitaciones para combinarlos.

RDF: Resource Description Framework (Marco de descripción de recursos) .Es un modelo de datos para los recursos y relaciones que se pueden establecer entre ellos. Aporta una semántica básica para este modelo de datos que puede representarse mediante XML.

RDF Schema: es un vocabulario para describir las propiedades y las clases de los recursos RDF con una semántica para establecer jerarquías de generalización entre dichas clases y propiedades.

OWL: Web Ontology Language (Lenguaje Ontológico para Web). Es un lenguaje para definir ontologías mediante la descripción detallada de clases y propiedades.

URI: Uniform Resource Identifier (Identificador Uniforme de Recursos). Es una cadena de caracteres corta que identifica inequívocamente un recurso (servicio, página, documento, etc.).

Namespace: Espacio de nombres. Es un conjunto de nombres en el cual todos los nombres son únicos.

Ontología: define los conceptos que hay en un dominio específico, describe las relaciones entre los mismos y sus propiedades.

Taxonomía: es la ciencia de la clasificación. Habitualmente se emplea el término para designar a la taxonomía biológica, la ciencia de ordenar la diversidad biológica en taxones anidados unos dentro de otros, ordenados de forma jerárquica, formando un sistema de clasificación.

Instancia: se refiere a a una realización específica de una clase o prototipo determinados.

WORDNET: base de datos léxica que contiene palabras de la lengua inglesa.

Protégé: es un entorno gráfico para la creación y edición de ontologías y bases de conocimiento.

SPARQL: Protocol and RDF Query Language (Lenguaje de protocolo y consulta de RDF) es un lenguaje de consulta de conjuntos de datos RDF.

Wiki: es un sitio web cuyas páginas pueden ser editadas por múltiples voluntarios a través del navegador web.

Wiki semántica: es una wiki que tiene un modelo de conocimiento subyacente descrito en sus páginas.

MediaWiki: es un software para wikis libre programado en PHP. Es usado por Wikipedia y otros proyectos de la Fundación Wikimedia.

Semantic MediaWiki: es una extensión de MediaWiki que permite buscar, organizar, crear tags, listar, evaluar y compartir el contenido de la wiki.

RDFIO: es una extensión de Semantic MediaWiki que extiende su funcionalidad de importar y exportar RDF.

8. Anexos

1. Archivo XML/RDF de la ontología Series TV exportada desde el Protégé en formato XML/RDF llamado "Series TV - XML/RDF.owl".
2. Archivo OWL de la ontología Series TV implementada en el Protégé llamada "Series TV - OWL.owl"

9. Bibliografía

http://es.wikipedia.org/wiki/Web_sem%C3%A1ntica

<http://es.wikipedia.org/wiki/Metadato>

[http://es.wikipedia.org/wiki/Etiqueta_\(metadato\)](http://es.wikipedia.org/wiki/Etiqueta_(metadato))

<http://es.wikipedia.org/wiki/Folcsonom%C3%ADa>

http://es.wikipedia.org/wiki/Lenguaje_de_marcado

<http://es.wikipedia.org/wiki/Metalenguajes>

<http://www.w3.org/TR/owl-features/>

<http://es.wikipedia.org/wiki/Namespase>

<http://zarza.usal.es/~fgarcia/doctorado/iuce/WSemantica.pdf>

<http://arantxa.ii.uam.es/~castells/publications/castells-uclm03.pdf>

https://es.wikipedia.org/wiki/Ontolog%C3%ADa#cite_note-SEP-1

http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html

http://es.wikipedia.org/wiki/Extensible_Markup_Language

<http://en.wikipedia.org/wiki/XML>

<http://www.rdfabout.com/quickintro.xpd>

<http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>

http://es.wikipedia.org/wiki/Resource_Description_Framework

http://en.wikipedia.org/wiki/Resource_Description_Framework

http://en.wikipedia.org/wiki/RDF_Schema

http://es.wikipedia.org/wiki/RDF_Schema

<http://www.w3.org/TR/owl-features/>

<http://protege.stanford.edu/doc/users.html>

[http://en.wikipedia.org/wiki/Prot%C3%A9g%C3%A9_\(software\)](http://en.wikipedia.org/wiki/Prot%C3%A9g%C3%A9_(software))

<http://wordnet.princeton.edu/>

<http://es.wikipedia.org/wiki/WordNet>

<http://en.wikipedia.org/wiki/SPARQL>

http://es.wikipedia.org/wiki/Wiki_sem%C3%A1ntica

http://es.wikipedia.org/wiki/Modelo_de_conocimiento

<http://www.mediawiki.org/wiki/MediaWiki>

<http://www.mediawiki.org/wiki/Installation>

http://semantic-mediawiki.org/wiki/Help:Introduction_to_Semantic_MediaWiki

http://en.wikipedia.org/wiki/Semantic_MediaWiki

<http://semantic-mediawiki.org/>

<http://semantic-mediawiki.org/wiki/FAQ>

<http://semantic-mediawiki.org/wiki/Help:Installation>

http://semantic-mediawiki.org/wiki/Help:Ontology_import

http://semantic-mediawiki.org/wiki/Help:Import_vocabulary

<http://www.mediawiki.org/wiki/Extension:RDFIO>

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/385/1/36605tfc.pdf>

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/493/1/27706tfc.pdf>

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/8138/1/jcabezascTFC0611.pdf>

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/10741/9/sfernandezveTFC1211.pdf>

<http://openaccess.uoc.edu/webapps/o2/bitstream/10609/955/1/36607tfc.pdf>