

# Proyecto PFC-UOC-OTG

Memoria del Proyecto: 14/06/2011

Autor: Jesús Antonio Párraga Úbeda

Consultor responsable: Felipe Geva Urbano

## Tabla de contenido

1.	RESUMEN EJECUTIVO .....	5
2.	OBJETIVOS DEL PROYECTO.....	5
3.	RELACIÓN DE TAREAS (EDT) .....	6
4.	DEFINICIÓN DE RIESGOS .....	11
5.	DEFINICIÓN DE HITOS .....	12
6.	PLANIFICACIÓN.....	12
7.	GESTION DE INCIDENCIAS .....	13
8.	HERRAMIENTAS A UTILIZAR .....	14
9.	WEB SEMÁNTICA.....	15
9.1.	CONCEPTO DE WEB SEMÁNTICA .....	15
9.2.	MEJORAS APORTADAS POR LA WEB SEMÁNTICA.....	16
10.	LENGUAJES Y TÉCNICAS DE LA WEB SEMANTICA .....	17
10.1.	URI/IRI .....	17
10.2.	XML.....	17
10.3.	XML Schema .....	17
10.4.	ONTOLOGÍAS .....	17
10.5.	LENGUAJES .....	20
10.6.	LÓGICA E INFERENCIA .....	23
10.7.	AGENTES.....	23
11.	DESARROLLO DE LA WEB SEMÁNTICA .....	23
12.	ESTADO DEL ARTE: ONTOLOGIAS TURISMO .....	24
13.	DESARROLLO DE LA ONTOLOGÍA .....	28
14.	INTRODUCCIÓN AL PROTOTIPO .....	33
15.	DATOS TÉCNICOS DEL PROTOTIPO.....	33
16.	DISEÑO DE LA APLICACIÓN .....	34
3.1	VISTA .....	34
3.3	CONTROL .....	39
3.3	MODELO .....	40
3.4	ACCESO INSTANCIAS .....	42
3.5	ESTRUCTURA DE CLASES.....	44
17.	CONFIGURACIÓN Y ENTREGA PROTOTIPO.....	45
18.	FUTUROS DESARROLLOS .....	47
19.	ENTREGA FINAL .....	48

20.	BIBLIOGRAFÍA.....	48
21.	DOCUMENTACIÓN RELACIONADA.....	52

<b>CONTROL DE VERSIONES</b>			
<b>Versión</b>	<b>Descripción</b>	<b>Fecha</b>	<b>Elaboración/Revisión</b>
1.00	Elaboración memoria final.	14/06/2011	Jesús Antonio Párraga Úbeda / Felipe Geva Urbano

## 1. RESUMEN EJECUTIVO

Este proyecto surge como desarrollo del proyecto final de carrera de la Ingeniería Informática de la UOC dentro del área XML y Web Semántica.

Inicialmente se han propuesto varias opciones a desarrollar, agrupadas en dos tipologías: creación de ontologías y gestión de ontologías.

De las opciones proporcionadas por los profesores encargados del proyecto se ha seleccionado el del primer grupo la creación de ontologías turísticas geográficas. La selección se ha realizado por las posibles aplicaciones en el entorno laboral y por el grado de conocimiento inicial.

Dada la naturaleza del proyecto se considera que el alumno cubrirá el total de las responsabilidades asociadas a los roles necesario, excepto el de usuario que será cubierto por el consultor responsable de la asignatura.

Hasta la fecha, la información proporcionada por el usuario ha sido la contenida en el plan docente, del cual se ha obtenido la información que se presenta en este documento, completada por las respuestas a las dudas que se han presentado al usuario a través del foro de la asignatura.

El objetivo principal es la obtención de una ontología que defina y represente información relacionada con el ámbito turístico de tal manera que deberá permitir etiquetar ficheros de información sobre puntos de interés turístico. Pueden considerarse puntos de interés, por ejemplo, monumentos, iglesias, museos, etc. De cada punto de interés podrán registrarse una serie de características. Además, esta ontología deberá permitir establecer la latitud y longitud de la información asociada a cada fichero, su posición geográfica. En la definición del alcance del proyecto quedará más delimitado y totalmente especificada en el análisis de los requisitos.

La gestión se llevará a cabo siguiendo la metodología que el alumno cursó en la asignatura Metodología y gestión de proyectos informáticos.

No se presentarán datos económicos del proyecto debido a su naturaleza académica.

Se considera iniciado el proyecto con la fecha de inicio del curso, el día 03/03/2011, y deberá finalizar con la última entrega el día 14/06/2011.

En este documentos se presentarán las tareas a realizar, los riesgos, la definición del alcance y la planificación del proyecto. Debe tenerse en cuenta que la información a la hora de realizar este informe es escasa, los conocimientos del alumno relacionado con la tecnología necesaria son mínimos y deberá cubrir todos los roles que sean necesarios y la jornada de trabajo será variable y dependerá de factores externos.

## 2. OBJETIVOS DEL PROYECTO

El principal objetivo del proyecto es la definición de una ontología para el etiquetado de la información de puntos de interés turístico. Esta definición deberá cubrir un ámbito lo mayor posible incluyendo información de los puntos de interés turístico incluido su localización geográfica (latitud y longitud).

Este objetivo puede dividirse en sub-objetivos que se intentarán cumplir en su totalidad:

1. Estudio de los conceptos relacionados con la web-semántica y las ontologías.

2. Estudio de las ontologías de información turística existentes. Estudiar si es posible la reutilización de alguna de las ontologías existentes para nuestro caso.
3. Estudio del ámbito a representar y creación de la ontología para dicho ámbito.
4. Instanciación de la ontología creada. Estas instancias de la ontología deben describir la información contenida en un fichero o conjunto de ficheros con información de puntos de interés turístico.
5. Creación de un prototipo para evaluar la ontología. Éste prototipo debe discriminar los ficheros o instancias según los criterios de búsqueda del usuario. Para que esto sea posible el prototipo debe permitir interactuar al usuario haciéndole preguntas sobre el tipo de información que le interesa: tipos de rutas, tipos de puntos de interés (museos, iglesias, ruinas, etc.), la zona geográfica, etc. Una vez obtenida la información del usuario el prototipo buscará en las instancias de la ontología (que se encontrarán preferentemente en una base de datos XML) y presentará al usuario una página web con enlaces a los recursos que puedan serle de interés.

La consecución de estos objetivos de implementación llevan asociados otros objetivos docentes y que también requerirán la realización de tareas:

- Estudiar los conceptos básicos de la web semántica.
- Conocer que es una ontología.
- Conocer algunos de los lenguajes de representación de ontologías más utilizados (OWL, RDF).
- Conocer alguno de los lenguajes de consulta de ontologías, como SPARQL, y de definición de reglas, como SWRL.
- Conocer algunas herramientas de edición y gestión de ontologías más conocidas.
- Conocer como acceder desde un lenguaje de programación al modelo definido, usando las API disponibles que facilitan su uso.
- Aprender a utilizar ontologías para realizar aplicaciones semánticamente más ricas.
- Realizar casos prácticos con estas ontologías, implementando una aplicación web más sencilla.

### **3. RELACIÓN DE TAREAS (EDT)**

#### **1. Definición del plan de proyecto**

Desarrollo del documento del plan de proyecto que debe incluir los objetivos del proyecto, las tareas a realizar y los principales hitos con la planificación.

Entrada: Plan docente, aportaciones en el foro del aula.

Salida: Documento con el plan de trabajo.

Dependencias: no existen dependencias.

Duración estimada: 11 jornadas.

## 2. Estudio teoría: web semántica y ontologías

Estudio de las definiciones, usos y estado del arte de la web semántica y las ontologías (metodologías y modo de trabajo).

Entrada: Documentación aportada por el consultor y búsquedas realizadas por el estudiante.

Salida: Documento con el resumen de teoría.

Dependencias: no existen dependencias.

Duración estimada: 5 jornadas.

## 3. Estudio lenguaje RDF y OWL

Estudio de los lenguajes y de las herramientas de diseño y desarrollo existentes.

Entrada: Documentación aportada por el consultor y búsquedas realizadas por el estudiante.

Salida: Documento con el resumen de teoría.

Dependencias: estudio de la teoría web-semántica y ontologías.

Duración estimada: 6 jornadas.

## 4. Instalación de las herramientas de desarrollo

Búsqueda, descarga e instalación de las herramientas de desarrollo necesarias. Además de la familiarización con las mismas.

Entrada: Documentación aportada por el consultor y búsquedas realizadas por el estudiante.

Salida: Instalaciones realizadas.

Dependencias: estudio de la teoría web-semántica y ontologías y de los lenguajes a usar.

Duración estimada: 1 jornadas.

## 5. Recogida y análisis de requisitos de la ontología a definir

Será necesario definir exactamente cuál es el ámbito de la ontología, el campo a definir es muy amplio por lo que solamente se podrá definir una parte, también será posible ampliar la definición que se desarrollo re-usando ontologías preexistentes.

Entrada: Plan de trabajo, resúmenes de teoría (recomendable).

Salida: notas con los requisitos recogidos.

Dependencias: Plan de trabajo realizado, recomendable estudio teoría.

Duración estimada: 3 jornadas.

## 6. Especificación de requisitos de la ontología a definir

Deberán quedar perfectamente registrados por escrito y aprobados por el usuario, que podrá realizar las puntualizaciones oportunas.

Entrada: Plan de trabajo, notas de la recogida de requisitos.

Salida: Documento con la especificación del análisis.

Dependencias: Plan de trabajo y recogida de análisis.

Duración estimada: 2 jornadas.

## 7. Búsqueda de Ontologías relacionadas

Una vez ha quedado bien definido lo que se espera de la ontología a desarrollar se podrá realizar la búsqueda de ontologías relacionadas con el turismo. Deberá realizarse un estudio de las distintas ontologías para que en caso de ser posibles realizar una reutilización de aquella o aquellas que nos sean más útiles.

Entrada: Plan de trabajo, definición requisitos.

Salida: Análisis de las distintas ontologías existentes y posibles usos.

Dependencias: especificación de requisitos.

Duración estimada: 2 jornadas.

## 8. Diseño del modelo conceptual

Con los requisitos, ámbito de la ontología deseada y posibles ontologías a reutilizar mediante los conocimientos adquiridos en las tareas de estudio se podrá realizar la definición o diseño de la ontología deseada.

Entrada: Plan de trabajo, definición de requisitos.

Salida: Diseño de la ontología.

Dependencias: Definición de los requisitos de la ontología.

Duración: 10 días.

## 9. Validación del modelo conceptual

Una vez definida la ontología debe validarse que cumple con todos los requisitos, que cubre todo el ámbito deseado y que será usable.

Entrada: Definición de la ontología y definición de los requisitos.

Salida: Ontología validada.

Dependencias: Definición de la ontología.

Duración. 2 días.

## 10. Creación de instancias concretas de la ontología creada

Deben crearse una serie de instancias de la ontología para que se pueda probar su usabilidad según los objetivos marcados.

Entrada: Definición de la ontología y definición de los requisitos.

Salida: conjunto de instancias de la ontología.

Dependencias: Definición de la ontología validada.

Duración: 3 días.

## 11. Creación del prototipo

La realización del prototipo deberá realizarse dividiendo el trabajo en varias sub-tareas debido a la diversidad de las actividades que será necesario realizar.

- 11.1. Recogida y análisis de requisitos
- 11.2. Diseño e implementación de la web
- 11.3. Estudio de las API disponibles para desarrollar el prototipo
- 11.4. Instalación de un servidor y entorno de desarrollo
- 11.5. Búsqueda y estudio de BBDD para XML
- 11.6. Instalación de la BBDD
- 11.7. Diseño e implementación de la instancia de BBDD
- 11.8. Diseño e implementación código servidor interactuar con web de usuario
- 11.9. Diseño e implementación código servidor API para ontologías
- 11.10. Diseño e implementación código servidor API para BBDD

Entrada: requisitos del prototipo.

Salida: implementación del prototipo.

Dependencias: plan de trabajo.

Duración: 34 días.

## 12. Pruebas prototipo

Una vez se ha implementado el prototipo deberán realizarse una serie de pruebas con instancias de la ontología creadas con este propósito. Debe quedar probado que los resultados son los esperados y que no se producen errores.

Entrada: prototipo.

Salida: informe pruebas del prototipo.

Dependencias: prototipo realizado.

Duración: 5 jornadas.

## 13. Validación prototipo

Con el prototipo construido y probado el usuario debe validarlo y darle el visto bueno.

Entradas: prototipo e informe de pruebas.

Salidas: prototipo validado.

Dependencias: prototipo debe estar desarrollado y probado.

Duración: 2 jornadas.

#### 14. Documentación (Memoria)

Realización de la memoria del proyecto. Adicionalmente debe indicarse los objetivos conseguidos, los aspectos formales, la bibliografía utilizada, los recursos requeridos y las conclusiones y aportaciones personales.

Entradas: Toda la documentación que se haya generado, así como las implementaciones.

Salidas: Memoria del proyecto.

Dependencias: La ontología y el prototipo deben estar probados y validados.

Duración: 5 jornadas.

#### 15. Elaboración de la presentación

Realización de una presentación en PowerPoint con un resumen que abarcará el trabajo realizado y el resultado obtenido con el mismo, tanto productos como de conocimientos adquiridos.

Entradas: Toda la documentación que se haya generado, así como las implementaciones.

Salidas: Presentación virtual.

Dependencias: La ontología y el prototipo deben estar probados y validados.

Duración: 5 jornadas.

#### 16. Entrega

Entrega del proyecto junto con la memoria y la presentación virtual.

Entradas: Toda la documentación que se haya generado, así como las implementaciones.

Salidas: Entrega en el aula.

Dependencias: La ontología y el prototipo deben estar probados y validados junto con la documentación (memoria y presentación).

Duración: 5 jornada.

Todas las tareas a realizar las deberá realizar el alumno por lo que no se ha indicado el responsable de cada tarea.

## 4. DEFINICIÓN DE RIESGOS

Identificación y definición de los posibles riesgos que pueden aparecer a lo largo de la vida del proyecto:

### R1. Desconocimiento de los fundamentos y las tecnologías a usar.

El alumno desconoce prácticamente todo lo relacionado con el proyecto, tanto lo relacionado con la teoría como con la práctica. Esto puede llevar a que las tareas se prolonguen más de lo previsto, y por tanto a la no consecución de todos los objetivos.

Deberá realizarse una planificación lo más flexible posible y volver a planificar en caso de ser necesario. Se deberá realizar un sobreesfuerzo. Algunas tareas podrían no realizarse.

### R2. Desconocimiento del detalle del alcance del proyecto

Con la información que se dispone en este momento y junto con R1 es difícil realizar una planificación muy realista. Se corre el riesgo de que no se puedan realizar todos los objetivos planificados y a tener que realizar distintas planificaciones. Se deberá realizar un sobreesfuerzo. Algunas tareas podrían no realizarse.

### R3. Los roles serán cubiertos prácticamente por completo por una persona

Al llevarse a cabo las actividades asociadas a todos los roles por una misma persona no podrá realizarse una planificación con tareas solapadas. Esta circunstancia incrementa el grado de criticidad del resto de riesgos.

### R4. Grado de dependencia del tiempo disponible

El tiempo disponible dependerá de otras actividades llevadas a cabo por el recurso (alumno) como es el trabajo, otra asignatura en curso y la vida privada. Esto implica que no siempre se podrá disponer del tiempo previsto en la planificación. Se podrá solucionar realizando un sobreesfuerzo y en su caso volviendo a planificar.

### R5. Jornadas de duración variable

El tiempo disponible será variable, las jornadas que se han indicado en la planificación no serán de duración homogénea. Esto hace que la planificación sea más difícil de realizar. En caso de que la planificación no sea lo suficientemente flexible será necesario realizar sobreesfuerzos y varias planificaciones.

## 5. DEFINICIÓN DE HITOS

A continuación se indican los hitos principales del proyecto junto a la fecha en que deberían cumplirse según la planificación realizada.

	Hito	Descripción	Fecha
H1	Plan de trabajo (PEC1)	Entrega del plan de trabajo.	16/03/2011
H2	Conocimientos iniciales	Adquisición de los conocimientos relacionados con el proyecto junto a los necesarios para realizar la ontología.	31/03/2011
H3	Ontología (PEC2)	Ontología definida, creada y validada.	19/04/2011
H4	Prototipo (PEC3)	Prototipo implementado y probado.	23/05/2011
H5	Memoria	Memoria realizada.	04/06/2011
H6	Presentación	Presentación realizada.	09/06/2011
H7	Entrega	Entrega realizada.	14/06/2011

La entrega de la PEC1 coincide con la entrega del Plan de trabajo.

La PEC2 puede coincidir con la adquisición de los conocimientos necesarios y la definición y desarrollo de la ontología.

La PEC3 puede incluir el desarrollo del prototipo.

La Entrega final puede contener las pruebas y validación del prototipo, la realización de la documentación y la entrega.

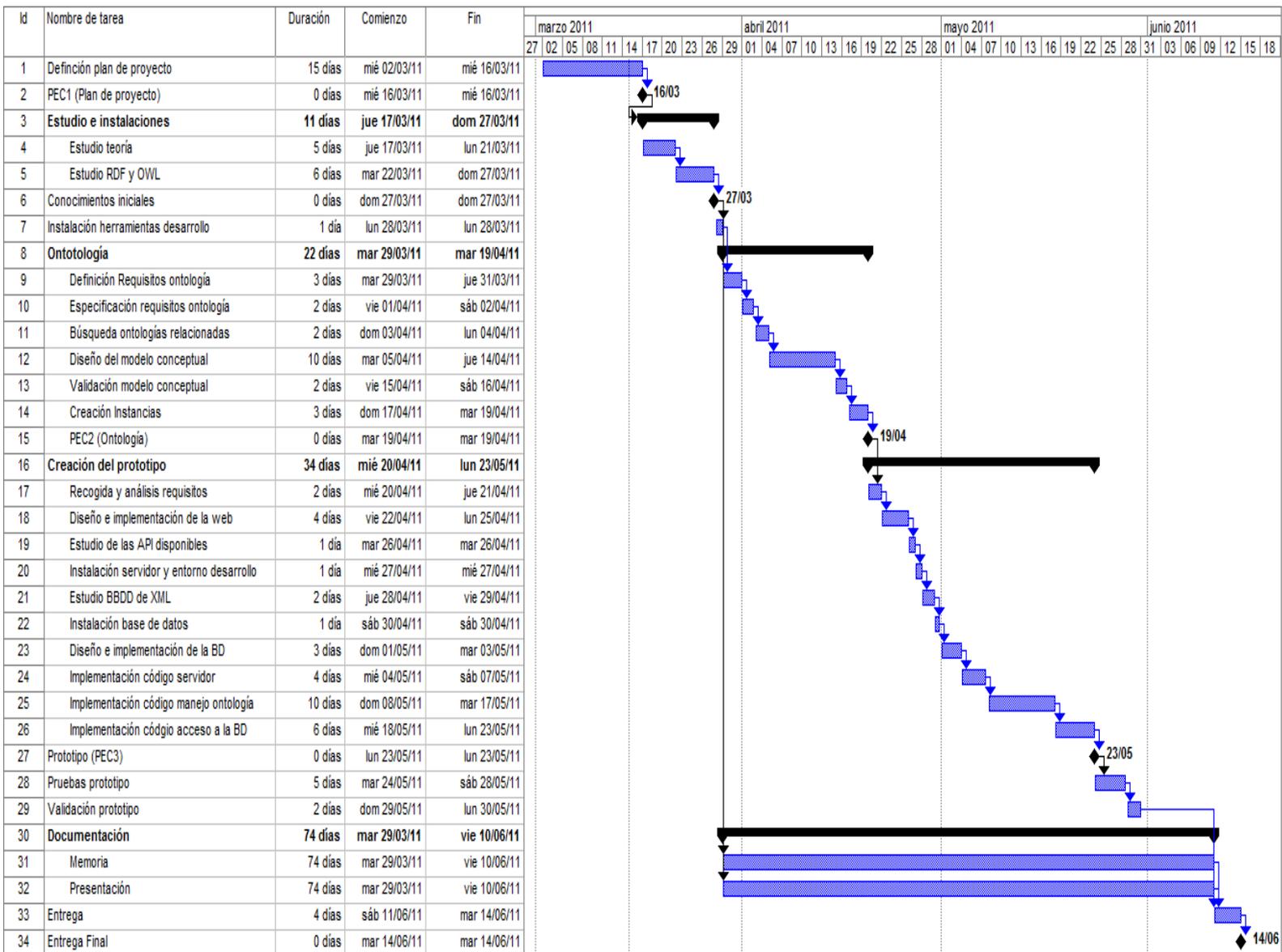
## 6. PLANIFICACIÓN

Dados los riesgos detallados una de las decisiones que se han tomado para la planificación es que se trabajarán siete días en semana para corregir posibles desviaciones.

Todas las tareas consecutivas se han tratado como dependientes, esto es así porque solo se dispone de un recursos, por lo que no tendría sentido diseñar una planificación con tareas en paralelo.

Una posible mejora a esta planificación, y que se puede realizar en caso de que se produzca alguno de los riesgos, es realizar una planificación donde la realización de la documentación se realiza en paralelo al resto de tareas, incrementando así el tiempo necesario del resto de tareas pero con menos tiempo requerido para la documentación.

A continuación se muestra el diagrama de Gantt con la planificación detallada:



## 7. GESTION DE INCIDENCIAS

A lo largo del desarrollo del proyecto se han producido una serie de incidencias. Gracias a la gestión realizada ha sido posible conseguir una entrega final en fecha.

### Incidencia 01

Riesgos asociados: R4 y R5

**I.01** Durante el desarrollo de las tareas incluidas en el hito H3 (coincidente con la PEC2) el recurso disponible dispuso de menos tiempo del planificado en cada jornada.

**C.01** La causa de la incidencia ha sido principalmente la obligatoriedad de realizar un curso por parte del recurso, durante tres semanas. A esto hay que incluir la inexperiencia de dicho recurso en la elaboración de definiciones conceptuales.

**S.01** La principal solución ha sido concentrar el tiempo dedicado a las tareas en el fin de semana aumentando el tiempo de dedicación en estos periodos, tal y como se había

previsto en la gestión de riesgos. Puesto que el peso del desarrollo era durante un periodo concentrado el feed-back que se requiere para estas tareas no ha podido ser el mismo que el recibido en tareas que se han desarrollado en intervalos homogéneos de tiempo.

**R.01** El riesgo que se ha producido ha sido un retraso en la entrega de la PEC2 de varias jornadas.

### **Incidencia 02**

Riesgos asociados: R4 y R5

**I.02** Por retrasos en la entrega del hito H3 el recurso disponible no ha podido dedicar todo el tiempo planificado para la elaboración de las tareas necesarias para la finalización del hito H4.

**C.01** La causa de la incidencia ha sido principalmente la necesidad de emplear tiempo y recursos dedicados a éstas tareas.

**S.01** La principal solución como estaba establecido en la gestión de riesgos ha sido la ampliación de horas dedicadas en jornadas de fin de semana y el uso de días de vacaciones.

**R.01** El riesgo que se ha producido ha sido un retraso de una jornada en la entrega de la PEC3.

No se han producido más incidencias reseñables y las existentes no han afectado a la entrega final, por lo que no se incluye más documentación al respecto.

## **8. HERRAMIENTAS A UTILIZAR**

Para el desarrollo del proyecto será necesario el uso de una serie de herramientas: se usarán para la definición de la ontología Protegé y OWL Viz que permite visualizarla, para validarla se usará Racer. El prototipo consistirá en una web desarrollada en HTML y Java, por lo que será necesario usar un IDE como Eclipse y un servidor de aplicaciones como Apache Tomcat.

<b>Herramienta</b>	<b>Finalidad</b>
Protegé 3.2.1 (o posterior)	Creación ontologías
Racer 1.9.0 (o posterior)	Razonador para validar ontologías
OWL Viz	Visualización de ontologías
Eclipse 3.5 (o posterior)	IDE para Java
Apache Tomcat 5.5 (o posterior)	Servidor de aplicaciones

## 9. WEB SEMÁNTICA

### 9.1. CONCEPTO DE WEB SEMÁNTICA

La web actual ha ido evolucionando en la presentación de la información para ser leída por las personas, ha mejorado sus sistemas de búsqueda, localización de la información y está suponiendo un cambio social en diversos ámbitos: relaciones interpersonales, administración pública a través de la red, accesibilidad y usabilidad o multimedia. La información es en muchos casos generada de manera dinámica y automática.

Sin embargo, estos contenidos, presentados en los sitios web, una gran cantidad en constante crecimiento de datos que en potencia será información de utilidad, deben ser gestionados en gran medida por los usuarios. Las búsquedas de datos en muchos casos no son lo eficaces que se desearía: los motores de búsqueda existentes realizan búsquedas por palabras clave obteniendo en unas ocasiones resultados muy amplios (muchos resultados no esperados) y en otras ocasiones muy limitados (perdiendo matices del sentido de la búsqueda deseada, los resultados no coinciden con lo que se estaba buscando).

Es decir, todo el procesamiento automático que se realiza no es capaz de procesar el sentido de los datos procesados, la semántica de lo que se ha requerido. En los agentes personales podemos encontrar problemas similares.

Incluso en los casos en los que los resultados de la búsqueda son los esperados los usuarios deberán realizar una búsqueda de la información dentro de los recursos encontrados (documentos en distintos formatos, imágenes, vídeos, etc.), una tarea que muy probablemente será tediosa y que requerirá mucho tiempo de trabajo.

La información se presenta para ser procesada por humanos pero no por máquinas. Una de las soluciones es el uso de técnicas complejas que permitan a las máquinas entender los textos encontrados en los sitios web, entender y procesar el lenguaje natural, tal cual se encuentra en la web actual. A pesar de los avances alcanzados en inteligencia artificial aun es una opción lejana, quizá algún día será posible que las máquinas procesen el lenguaje natural. Esta solución no requeriría modificar los contenidos de la web actual pero su complejidad y coste no lo hace posible.

Una solución menos costosa y que se está usando son los sistemas wrappers que permiten de manera semiautomática extraer información de los sitios web actuales.

Otra opción es ampliar los contenidos de la web de tal manera que las máquinas sean capaces de procesar toda la información representada, incluso el valor semántico que en la web tradicional se pierde. Para conseguir esta evolución de la web no será necesario reemplazar una por otra de una manera radical, sino que, la web semántica se está introduciendo de manera gradual en la web actual.

Existen varias definiciones de web semántica, según Lee Feigenbaum “La web semántica hace que cada enlace signifique algo para las máquinas”. Para Berners-Lee “La Web semántica es una extensión de la Web actual en la que la información tiene un significado bien definido que permite a personas y ordenadores trabajar mejor de forma conjunta”. La Web semántica no es una nueva web, sino un complemento a la actual que facilitará el uso de la información existente con un mínimo esfuerzo.

La Web semántica no solo aborda el problema del tratamiento de los datos encontrados en los sitios web, pasando del tratamiento basado en su sintaxis al tratamiento

semántico, sino que aborda los problemas semánticos que se plantean en cualquier tipo de comunicación, incluida la comunicación entre máquinas.

Otra forma de entender la Web semántica es como una ampliación de la web actual que permitirá a las máquinas entender los datos contenidos en la web actual. Aquí es necesario saber a que nos referimos al usar la palabra entender. El hecho de entender implica la capacidad de modelar la realidad, inferir conocimiento partiendo de los modelos obtenidos y ser capaces de responder a preguntas relacionadas con estos conocimientos. La inferencia se puede definir como la capacidad de clasificar, correlacionar y extrapolar los conocimientos adquiridos a otros ámbitos. Las máquinas son capaces de inferir y responder a preguntas, sin embargo, no se ha conseguido que realicen de manera automática un modelado de la realidad, por lo que serán las personas las que al crear la Web Semántica proporcionen éste modelado de los datos. A estos modelos se les llama Ontologías (que se verán más adelante en este documento).

El precursor de la web semántica fue Tim Berners-Lee que a través de su organización W3C está promoviendo añadir semántica a la web. Según Berners-Lee su idea original de la web ya incluía semántica, pero no fue posible con los medios del momento. La primera aparición del concepto de Web semántica aparece en 2001, en un artículo de Berners-Lee, Hendler y Lassila en la revista Scientific American.

## 9.2. MEJORAS APORTADAS POR LA WEB SEMÁNTICA

Algunas de las mejoras que se encontrarán en la web semántica frente a la web tradicional son las siguientes:

- Las búsquedas dejarán de realizarse exclusivamente por palabras clave, se realizarán búsquedas matizadas por la semántica de las frases de búsqueda.
- Se podrá realizar una gestión automatizada de la información. Permitiendo la extracción de nuevo conocimiento y la discriminación de datos no deseados.
- Permitirá restringir el acceso a según qué datos.
- Se posibilitará la comunicación entre máquinas que además atenderán las órdenes recibidas, no solo por el contenido de los datos y su estructura, sino por el sentido de éstos.
- Será posible mantener organizados los datos de manera conceptual y acceder a recursos según las necesidades de información.

Algunos usos concretos de la web actual donde la mejoras serán visibles:

- Los usuarios podrán disponer de una comparación previa de los precios y las características de los productos que le interesen ofrecidos por los distintos proveedores online.
- Aumento del grado de automatización en la cadena de valor externa de las empresas en sus relaciones con proveedores y clientes.
- Automatización de web del conocimiento, aumentando a través de un agente los enlaces entre los distintos recursos añadidos por los usuarios de manera manual, aumentando así el conocimiento incluido en estas web.

- Mejora de los asistentes personales existentes en la actualidad.

## **10. LENGUAJES Y TÉCNICAS DE LA WEB SEMANTICA**

Para conseguir añadir contenido semántico a la web actual no será necesario reemplazar los contenidos actuales, sino que, será necesario modificarlos, añadiéndoles información. Además, será necesario el uso de una serie de agentes que permitan a las máquinas procesar esta nueva información. Alguna de las técnicas y lenguajes más usando son detallados a continuación:

### **10.1. URI/IRI**

Identificador uniforme de recursos, es una cadena de caracteres usada para identificar un recurso. Se usará en la web semántica con este mismo fin, identificar y nombrar los recursos, clases y propiedades del dominio.

### **10.2. XML**

Como se ha comentado más arriba, los contenidos de la web están orientados a la comprensión humana, predomina las técnicas para su presentación visual a través de HTML. Para que las máquinas puedan manejar esta información será necesario añadir nueva información a éstas páginas web.

Para añadir información relativa a la información contenida en la página, meta información, se usa XML. El proceso consistiría en añadir etiquetas XML que indican qué información contienen y que será la que se está mostrando en el HTML. Actualmente la mayoría de los sitios generan dinámicamente los datos que muestran, obtenidos de bases de datos, por lo que añadir estas etiquetas se podría hacer de manera automatizada.

La tarea principal de XML en la web semántica será proporcionar una sintaxis al resto de lenguajes o capas de componentes. Con la ventaja de que XML está cada vez más asentado en las ciencias de la información.

Se usarán los espacios de nombres como medio para identificar de manera única las etiquetas asociadas a un dato.

### **10.3. XML Schema**

Lenguaje para restringir la estructura de documentos XML y adicionalmente proporciona una serie de tipos de datos.

### **10.4. ONTOLOGÍAS**

El término ontología dentro de las ciencias de la computación. Cuando se habla de una ontología se está hablando, a grosso modo, de una conceptualización, término cercano a la definición filosófico de ontología.

Se puede encontrar una gran variedad de definiciones de ontología, una de las más usuales es la de Gruber: “Una ontología es una especificación de una conceptualización” o “una definición de un vocabulario de representación de un dominio compartido”. Más tarde Studer amplió la primera definición de Gruber: “Una ontología es una especificación explícita y formal de una conceptualización”.

En la práctica una ontología consiste en un conjunto finito de términos y sus relaciones (Antoniou). Los conceptos importantes del dominio son representados por términos o clases y las relaciones entre estas clases ofrecen una jerarquización de los términos.

El concepto de ontología permitirá organizar formalmente la meta información necesaria según el dominio de la web.

La definición de una ontología incluye además de las clases y sus relaciones otras herramientas conceptuales como son las propiedades (de las clases), restricciones (sobre los valores que se pueden incluir), disyunciones y otras relaciones lógicas.

Permiten crear la definición común de un dominio permitiendo que distintas máquinas se puedan comunicar. También es posible que a través de una ontología sea posible poner en común las distintas conceptualizaciones usadas por las distintas máquinas. Pero no solo son útiles para las máquinas, al estar organizada la información en categorías o clases, permite facilitar la navegación a los usuarios, ya que se puede mostrar la información relacionada con la información que está incluida en la web en la que se encuentra.

Las búsquedas de información serán más sencillas y eficientes, para las personas y para las máquinas. Gracias a la jerarquía de clases, si una búsqueda no ofrece resultados el buscador puede sugerir una búsqueda más genérica, por el contrario, si la búsqueda obtiene como resultado demasiados resultados, se podría una búsqueda más específica. En general, será posible que las búsquedas sean por conceptos y no por palabra clave, con las limitaciones que esto implica.

## **Tipos de Ontologías**

Se han realizado varias tipologías de ontologías siguiendo distintos criterios:

Fox y Gruninger, para los que una ontología es un “vocabulario más una especificación del significado de dicho vocabulario”, hacen una clasificación de las ontologías basada en el grado de formalidad en la especificación del significado:

- Ontologías informales: usan un lenguaje natural.
- Ontologías semiformales: proporcionan taxonomías.
- Ontologías formales: definen la semántica del vocabulario por una axiomatización completa y efectiva.

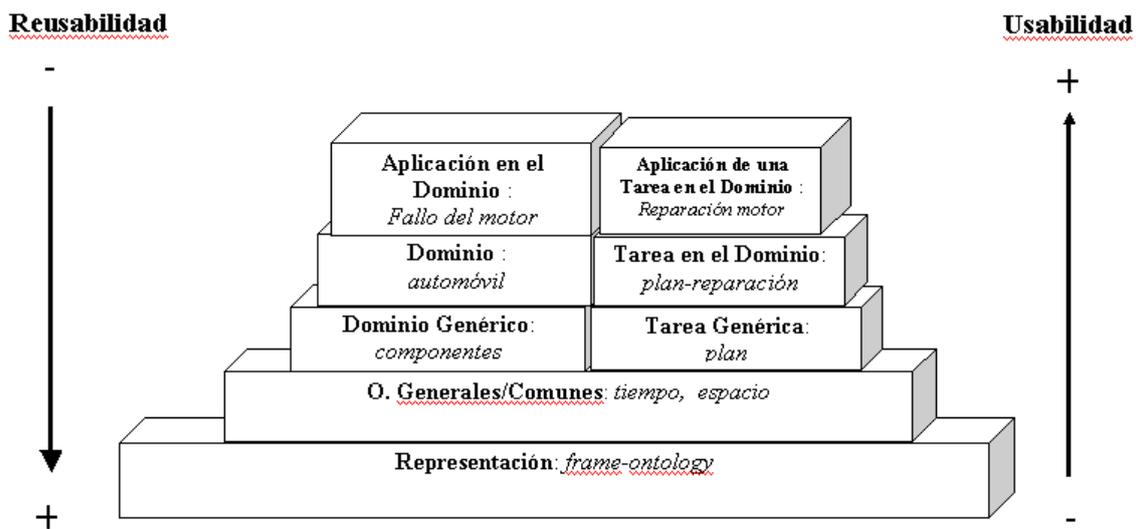
Según Van Heist las ontologías pueden clasificarse de acuerdo a la cantidad y tipo de estructura de la conceptualización en:

- Ontologías terminológicas: especifican los términos que son usados para representar conocimiento en el universo del discurso. Suelen ser usadas para unificar vocabulario en un campo determinado.
- Ontologías de información: especifican la estructura de almacenamiento de bases de datos. Ofrecen un marco para almacenamiento estandarizado de información.
- Ontologías del modelado del conocimiento: especifican conceptualizaciones del conocimiento. Contiene una rica estructura interna y suelen estar ajustadas al uso particular del conocimiento que describen.

Guarino clasifica las ontologías según el grado de generalidad o nivel de dependencia y relación con una tarea específica desde un punto de vista particular:

- Ontologías de Alto nivel o Genéricas: describen los conceptos más generales. Por ejemplo tiempo, materia, espacio, etc. que son independientes de un problema o dominio concreto.
- Ontologías de Dominio: describen un vocabulario relacionado con un dominio genérico. Por ejemplo medicina o informática.
- Ontologías de Tareas o Técnicas básicas: describen una tarea, actividad o artefacto, por ejemplo componentes, procesos o funciones.
- Ontologías de Aplicación: describen conceptos que dependen tanto de un dominio específico como de una tarea específica, y generalmente son una especialización de ambas. Contienen conocimiento esencial para modelar una aplicación particular.

Otra manera de clasificarlas es por el grado de usabilidad y reusabilidad. Como podemos ver en el siguiente gráfico el aumento de reusabilidad implica una disminución en el grado de usabilidad, y viceversa:



## Metodología

A pesar de que no existe una metodología de trabajo para desarrollar ontologías, me parece interesante y he seguido para realizar este trabajo la propuesta por Honrubia López en su “Introducción a las Ontologías”. Esta metodología está orientada a los sistemas de información y se divide en los siguientes pasos:

1. Identificar el propósito y el alcance: debe conseguirse la especificación del contexto de aplicación y el punto de vista del modelo. Donde, el contexto de aplicación describe el dominio de la aplicación. Debemos saber para qué es necesario construir la ontología. Y el modelado del punto de vista describe el tipo de modelo.
2. Construcción de la ontología: consiste en la captura, codificación en un lenguaje formal e integración con otras ontologías existentes.

3. Evaluación: para conseguir un diseño definitivo que proporcione valor añadido, como puede ser su reutilización. En nuestro caso, mediante la implementación de instancias de la ontología.
4. Documentación y reutilización: es recomendable realizar la documentación en paralelo a los pasos anteriores y servirá entre otras cosas para facilitar su reutilización en otras ontologías.

### **Principios de construcción de una ontología**

Van Heirst, Schreiber y Wielinga propusieron una serie de condiciones que deben alcanzarse en la definición de una ontología:

1. Claridad para el entendimiento del usuario.
2. Objetividad para incluir y excluir términos del dominio.
3. Completitud de los términos.
4. Coherencia para conseguir inferencias consistentes con las definiciones.
5. Máxima extensibilidad monótona. Todas las generalizaciones y especializaciones deben estar incluidas.
6. Principio de distinción ontológica. Las clases de una ontología deben ser disjuntas.
7. Diversificación de las jerarquías para aumentar la potencia de los mecanismos de herencia múltiple.
8. Modularidad. Disminuyendo el grado de acoplamiento.
9. Estandarización de la nomenclatura.
10. Minimización de la distancia semántica entre conceptos emparentados y agrupamiento de los mismos.

### **10.5. LENGUAJES**

El uso de metalenguajes es necesario en la web semántica, pero no añade la semántica necesaria, para esto se usan otros lenguajes como RDF y OWL basados en la sintaxis XML.

**RDF (Resource Description Language):** permite especificar recursos y relaciones entre ellos añadiendo cierto grado de semántica, pero especialmente permite la comunicación entre máquinas que intercambian información. Las declaraciones de los recursos se realizan a través de expresiones con la forma sujeto-predicado-objeto, conocidos como tripletas. Cada uno de los elementos de la expresión está identificado por una URI lo que permite que sean usados en diferentes aplicaciones, o bien, en el caso del valor de las propiedades con valores simples. Esta estructura genera un grafo

dirigido, donde los extremos corresponden a los recursos relacionados o a sus propiedades y los arcos representan estas relaciones.

Un ejemplo de tripleta obtenido de w3.org:

“<http://www.example.org/index.html> has a creator whose value is John Smith”

Donde:

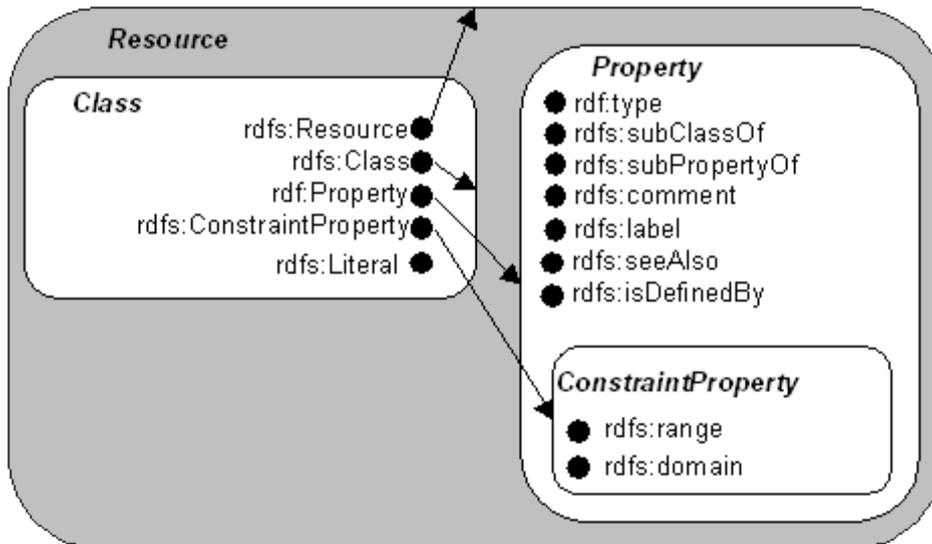
La URL “<http://www.example.org/index.html>” corresponde al sujeto "creator" es el predicado y "John Smith" es el objeto.

Un ejemplo obtenido de la w3.org, donde se puede ver el grafo correspondiente a un recurso de tipo persona cuya propiedad nombre tiene el valor “Eric miller” y la propiedad dirección de correo es <mailto:em@w3.org>. Podemos ver como los recursos y sus relaciones (nodos y arcos del grafo) son identificados por URI (<http://www.w3.org/People/EM/contact#me>, <http://www.w3.org/2000/10/swap/pim/contact#fullname>) y las propiedades toman valores simples como “Eric Miller”.



**RDF Schema:** con una semántica más rica que RDF, parte de éste para añadirle el concepto de jerarquía a sus clases y propiedades. Con RDF es posible describir recursos, sus relaciones con otros recursos y darle valor a ciertas propiedades de los recursos, pero no es posible añadir restricciones a las relaciones entre recursos o a los valores de sus propiedades. Gracias a RDF Schema esto es posible.

En el siguientes esquema vemos los conceptos aportados por RDF (identificados por el espacio de nombre *rdf*) y por RDF Schema (con espacio de nombre *rdfs*).



RDFS incluye el concepto de jerarquía de clases con *Class* y *subClassOf* y de jerarquía de propiedades con *rdf:SubPropertyOf*. Y las restricciones en las relaciones de propiedad con el concepto de rango (*rdfs:range*) o conjunto de valores que la propiedad puede tomar y dominio (*rdfs:domain*) o conjunto de objetos a los que se puede aplicar una propiedad.

**OWL (Web Ontology Language):** este lenguaje añade nuevos conceptos semánticos a los anteriores, como relaciones de disyunción entre clases, restricciones de cardinalidad, la igualdad, enumeraciones de clases y características de las propiedades.

OWL se proporciona en tres lenguajes, con distintos niveles de complejidad expresiva:

OWL Lite: proporciona la jerarquización de clases y algunas restricciones (por ejemplo de la cardinalidad solo proporciona valores 0 y 1).

OWL DL (Description Logics): proporciona la mayor semántica posible para garantizar las propiedades de integridad de cómputo (todas las conclusiones que se alcanzan con la ontología son computables) y de decibilidad (todos estos cálculos obtienen resultados en tiempos finitos). Contiene todo el vocabulario OWL, pero bajo ciertas restricciones que permiten aplicar a las ontologías el cómputo basado en la lógica formal.

OWL Full: proporciona el máximo grado de expresividad que puede ofrece OWL pero no las garantías de cómputo de OWL DL.

En la siguiente imagen proporcionada por w3.org podemos ver cómo los distintos lenguajes van añadiendo capacidad semántica y complejidad unos a otros. Estos lenguajes no solo son compatibles, sino que, van usando la sintaxis de los más sencillos:



En este caso usaremos OWL DL ya que la semántica que proporciona es suficiente y necesitamos de sus garantías de cómputo.

**SPARQL (Query Language for RDF):** Lenguaje que permite realizar consultas en el grafo RDF. Según la implementación se puede realizar consultas en un documento RDF o incluso dar de alta, modificar y eliminar datos.

**SWRL (Semantic Web Rule Language):** Lenguaje de reglas para web semántica, mezcla de OWL DL y lenguaje de marcado. Las reglas están formadas por un antecedente y un consecuente, de tal manera que si se cumplen las reglas del antecedente también se cumplirá las reglas especificadas en el consecuente.

## 10.6. LÓGICA E INFERENCIA

Los lenguajes declarativos permiten describir lo aquellos que sea necesario sin necesidad de pensar en cómo será deducido. La lógica formal ofrece estos lenguajes formales para expresar el conocimiento y una semántica formal.

Permite llegar a encontrar recursos que se encuentran en la web semántica, donde las reglas que los definen están escritas en los lenguajes que se han visto más arriba.

La inferencia permite determinar la consistencia del modelo obtenido. Para realizar inferencias se hace uso de la lógica, que cuanto más expresiva sea menos potencia de inferencia proporcionará. La lógica que más se suele usar es la basada en reglas que es la más sencilla.

## 10.7. AGENTES

Él último de los elementos principales que componen la web semántica. Las máquinas que harán uso de los datos de la web para generar información y gestionarla necesitan del software adecuado. A este software, aplicaciones, se le llama agente.

Los agentes encontrarán los datos gracias a la metainformación incluida en la web ayudados por las definiciones de ontologías existentes en el dominio de su especialización y gracias a la lógica serán capaces de procesar la información. Además, gracias a las ontologías los agentes podrán colaborar.

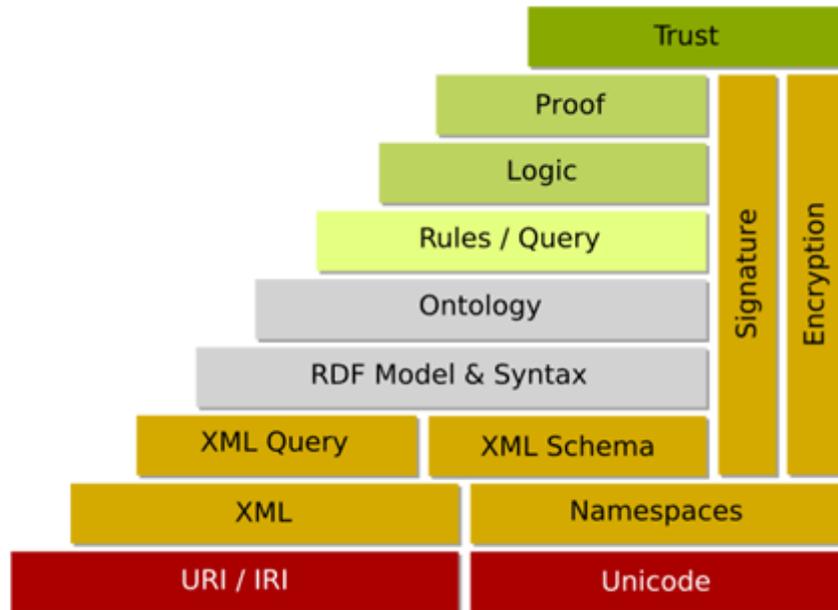
## 11. DESARROLLO DE LA WEB SEMÁNTICA

Los distintos lenguajes y técnicas usados para desarrollar una web semántica parten unos de otros para ir añadiendo complejidad, riqueza semántica y usabilidad. Así partiendo de XML, que nos permite estructurar los datos en la web, añadiendo una semántica muy básica, similar a un modelo de datos, mediante RDF que usa la sintaxis XML. Ampliado el lenguaje RDF y partiendo de él, RDF Schema añade semántica

mediante la jerarquización de los datos definidos en RDF. Por último OWL es el lenguaje para definir ontologías, que aporta los conceptos comentados más arriba.

Uno de los problemas de la web semántica es que no existe un estándar para definir el conocimiento de un dominio, así un mismo dominio puede estar definido en varias ontologías desarrolladas por distintos interesados. Por lo que se está haciendo un gran esfuerzo para unificar en la medida de lo posible todas las definiciones. Partiendo de los niveles de complejidad más sencillos se pueden ir buscando consensos iniciales.

Así se recomienda el desarrollo visto en capas de la web semántica:



Haciendo un resumen:

XML proporciona una sintaxis para conseguir documentos estructurados con datos, pero sin añadir semántica a los datos.

RDF permite definir un modelo de recursos y las relaciones entre ellos, proporcionando una semántica básica usando la sintaxis de XML.

RDF Schema amplía y complementa RDF permitiendo jerarquizar las clases y sus propiedades.

OWL enriquece la semántica de RDF Schema añadiendo los conceptos de disyunción entre clases, la cardinalidad, la igualdad, etc.

## 12. ESTADO DEL ARTE: ONTOLOGIAS TURISMO

El sector turístico puede aprovechar el valor añadido que la web semántica le puede aportar: facilitando el acceso de los usuarios a la información turística sin necesidad de conocimientos TIC y con el uso cada vez más cercano al lenguaje natural, posibilidad de creación de nuevos productos y generación de aplicaciones cada vez más dinámicas que sean capaces de adaptarse mejor los cambios del mercado.

Tanto en el sector privado como en el público han nacido proyectos en los que se han creado catálogos de datos, taxonomías y ontologías cada vez más completas. Se han desarrollado ontologías que abarcan conceptos turísticos genéricos y también otras para subdominios más específicos.

Inicialmente el desarrollo de las ontologías tenía un propósito local, para uso dentro del proyecto en el que se incluían, actualmente se está intentando crear puntos de enlace entre los distintos desarrollos, creando relaciones entre ellas (por ejemplo, el proyecto “Linking Open Data”).

Muchas de las ontologías existentes se basan en especificaciones y tesauros existentes. Las más destacadas:

- El tesoro de conceptos turísticos y actividades de ocio de la Organización Mundial del Turismo (TWO). Este tesoro puede ser usado como guía terminológica consiguiendo una normalización y estandarización en el vocabulario usado en los recursos turísticos. Con 1500 términos registrados en varios idiomas (inglés, francés y español) sirve de base a las ontologías del dominio más importantes.

- La especificación de la Open Travel Alliance (OTA) persigue facilitar la interoperabilidad entre los distintos agentes desarrollados desde los distintos ámbitos del turismo desarrollados por distintas compañías. Esta tarea consiste principalmente en el desarrollo y mantenimiento de una biblioteca de esquemas XML llamada OpenTavel XML, que contiene 200 documentos agrupados en paquetes.

Se detallan a continuación alguna de las ontologías más destacadas:

### **HarmoNET**

Proyecto fundado en 2006 para suceder a Harmonise y Harmo-Ten. Actualmente tiene más de 20 miembros relacionados con el turismo y las tecnología de la información, alguna de los miembros son organismos como WTO (World Tourism Organisation), ETC (European Travel Commission).

El objetivo de HarmoNET es facilitar la interoperabilidad de los distintos sistemas dedicados al turismo, ya que no existe un estándar común lo que hace que el intercambio de información sea costoso o imposible.

Este objetivo no lo consigue mediante la creación de un estándar sino con un sistema abierto de mediación entre los distintos sistemas. Los servicios ofrecidos por este sistema se pueden resumir en:

- Gestión de una ontología: proporciona una definición de dominio común, que sirva de referencia para la construcción de modelos de datos para el turismo.
- Mediación: mediante una solución técnica para la interoperabilidad de los datos heterogéneos ya existentes que permite la comunicación entre dos sistemas que no tienen un definición común mediante la transformación de los conceptos de un extremo a los conceptos el otro extremo.

- Plataforma común: esta plataforma digital facilita la comunicación y el acuerdo entre los miembros del proyecto. También se incentivan métodos más tradicionales para el intercambio de información conferencias, talleres y seminarios.
- Ventajas empresariales: facilitando el acceso a soluciones tecnológicas y las relaciones con otros interesados.

Las tecnologías usadas son el lenguajes RDF con protege.

### **QALL-ME Ontology**

La ontología QALL-ME define conceptos relacionados con el ámbito turístico. Desarrollada en RDF y OWL contiene 261 conceptos 55 propiedades de dichas clases y 55 relaciones entre dichas clases. Abarca conceptos como destino, país, evento, lugar, transporte, etc. que coinciden con las 17 clases que se encuentran en el nivel superior de la ontología.

El objetivo de esta ontología es facilitar el desarrollo de aplicaciones que permita contestar a preguntas relacionadas con el dominio realizadas por usuarios en lenguaje natural, permitiendo el uso de varios idiomas.

Para su desarrollo se han usado los lenguajes RDF Schema y OWL-DL implementados en Protégé 3.2.2.

### **DERI e-Tourism**

El principal objetivo perseguido por eTurism es la creación de un portal web que aproveche las ventajas de la web semántica y que permita el intercambio de información entre agentes y clientes del turismo. Inicialmente el proyecto OnTour dentro del cual se inscribe ésta ontología concebida para el sector en Austria está abierto a nuevos miembros.

Inicialmente se partió de los conceptos establecidos en el tesoro de la WTO

Los principales conceptos que se tratan son los de preferencias de alojamiento, preferencias de actividades y paquetes de vacaciones. El usuario definirá sus preferencias y el sistema le propondrá los paquetes de vacaciones ofrecidos por los agentes. Además contiene información sobre la posición geográfica de las actividades y de los alojamientos lo que permite darle al usuario la distancia entre unos y otros.

La ontología DERI está desarrollada en OWL DL con la ayuda de Protégé 3.0.

### **EON Travelling Ontology**

Ontología que contiene varias sub-ontologías relacionadas con el turismo llegando a disponer de 300 conceptos y 100 relaciones, donde se mezclan conceptos del turismo con conceptos de viajes.

### **cDott Ontology**

Como en alguna de las anteriores ontologías, cDott incluye la definición del dominio turístico desde el ámbito del turismo, su interés, y desde el ámbito de turista, sus

intereses. Intentando unir la semántica del viaje con la del viajero. Para conseguirlo está construido con una estructura modular donde se integran ontologías preexistentes (como alguna de las que se presentan en este documento: Harmonise y EON, además de otras) y que permite realizar una búsqueda más dinámica en función de las preferencias del usuario, consigue alinear el conocimiento de las ontologías de turismo con el conocimiento relacionado con las necesidades de los usuarios. No solo utiliza ontologías directamente relacionadas con el turismo, sino que usa ontologías con conocimiento relacionado, como posición geográfica (w3c Geo), ontología de tiempo (w3c Time Ontology), haciendo un uso muy beneficioso de ontologías ya definidas y establecidas.

### **TAGA Travel Ontology**

Forma parte del proyecto TAGA (Travel Agent Game in Agentcities), un agente experimental para la simulación de un mercado mundial de viajes en Internet, centrada en el ámbito de los viajes y escrita en OWL, pretende ser una plataforma para la investigación de la web semántica en sistemas distribuidos en dominios turísticos.

### **GETESS Ontology**

Ésta ontología contiene 1043 conceptos y 201 relaciones, proporcionando cada término en inglés y alemán. Fue desarrollada hace 10 años dentro del proyecto GETESS (German Text Exploitation and Search System) para la obtención de información encontrada en sitios web turísticos para su clasificación, interpretación y presentación como resultado de preguntas en lenguaje natural realizadas por el usuario.

### **Ontología Cruzar**

Financiado por el ayuntamiento de Zaragoza y desarrollado en 2007 dentro del proyecto “Un visitante, una ruta” con el objetivo de desarrollar una aplicación para el cálculo de rutas dentro de la ciudad de Zaragoza creadas a partir de las preferencias del usuario. Basada en la ontología de alto nivel DOLCE y de los datos contenidos en distintas bases de datos: “Agenda Zaragoza” e IDZARa (Infraestructura de Datos espaciales de la ciudad de Zaragoza).

Se ha desarrollado con el lenguaje OWL-DL.

### **Ontología ANOTA**

Bajo el proyecto ANOTA financiado por la Unión Europea y el Ministerio de Educación de España para la evolución del uso de anotaciones semánticas de las PYMES. Reutiliza ontologías y esquemas XML turísticas existentes, como el de la OTA.

### **Conclusiones**

No solo existen las ontologías turísticas aquí presentadas, se pueden añadir otras tanto del ámbito de la investigación (proyecto GODO) como usadas en la web (OnTourism, e-Tourism, Mondeca Tourism Ontology y Hi-Touch Ontology, etc.), es decir se han desarrollado y se usan gran número de ontologías en el ámbito estudiado y existe un esfuerzo por estandarizar términos y conseguir que los distintos sistemas puedan interactuar. Esto es posible gracias al gran auge que el sector turístico ha sufrido

últimamente y especialmente a través de internet. Por otro lado las facilidades y el valor añadido que proporciona la web semántica en el sector es elevado.

### 13. DESARROLLO DE LA ONTOLOGÍA

Definido el alcance de la ontología en el plan del proyecto se ha pasado a obtener información relacionada con las rutas turísticas, creando una lista de términos asociados, de esta forma se ha obtenido una abstracción del tema. En esta primera versión muy probablemente no están incluidos todos los términos, la ontología irá creciendo con la puesta en práctica de la misma.

Las fuentes de información se pueden consultar en el apartado de bibliografía. Otra de las fuentes de información que se han usado han sido las ontologías existentes, principalmente: Harmonise, Qall-me y la ontología cruzar, que han sido estudiadas no solo con el fin de obtener información sobre los conceptos que pueden incluirse en la ontología, sino para ser reutilizados.

El espacio de nombre utilizado para distinguir las entidades de nuestra ontología del resto es *uoc*.

La definición se ha desarrollado con Protégé 3.4.5 y OWL 1.

Las clases (en *cursiva*) que se han incluido en esta primera versión de la ontología son las siguientes (se presentan sin espacio de nombres):

- *Ruta*: representa los distintos tipos de rutas permitidos, como las rutas a motor (*RutaMotor*) y rutas sin motor (*RutanoMotor*). Dentro de las rutas a motor se han incluido cuatro categorías, estas categorías no pretenden diferenciar todas las incluidas en la DGT, pero si hacer una clasificación que incluya a aquellas con las que se suele hacer una ruta turística: *RutaAutocaravana*, *RutaMotocicleta* (incluye rutas en motocicletas, ciclomotores y similares), *RutaCuatriciclo* (puede incluir motocicletas y ciclomotores de más de dos rueda, como pueden ser los quartz) y *RutaCoche* (puede incluir automóviles y turismos), se han incluido las propiedades *alturaMaximaPermitidoVehiculo*, *anchoMaximoPermitidoVehiculo* y *pesoMaximoPermitidoVehiculo* que pueden indicar, si existe, la altura, el ancho y el peso que la ruta permite para el vehículo utilizado. Para las rutas sin motor se han incluido: rutas caminando (*RutaPie*), rutas a caballo o en mula (*RutaCaballo*), rutas en bicicleta de montaña y de carretera (*RutaBiciMontaña* y *RutaBiciCarretera*).
- *Servicio*: distintos tipos de servicio que se pueden encontrar en la ruta, se ha incluido una clasificación genérica que puede ser ampliada por las ontologías turísticas existentes (como Harmonise y Qall-me), de momento se han incluido: *Alojamiento para los posibles alojamientos como hoteles*, moteles, posadas, etc. ; *AreaServicio* servicios básicos gratuitos en carretera; lugares donde es posible comer, como restaurantes, *Restauracion*; centros de asistencia sanitaria, hospitales, cruz roja, etc., *Sanitarios* y lugares donde se ofrecen servicios de transporte, estaciones de tren, ferrocarril, etc., *Transporte*. También se incluyen servicios de información turística (*Informacion*).

- *PuntoInteres*: en la ruta podemos encontrar distintos puntos de interés que pueden ser puntos de interés rurales (*Rural*), culturales (*Cultura*), de naturaleza (*Naturaleza*) o bien de ocio (*Ocio*).
- *Formato*: los recursos existentes relacionados con las rutas pueden ser ficheros de audio (*Audio*), los distintos documentos de texto que se pueden encontrar en distintos formatos (*Documento*), imágenes que pueden ser fotografías y mapas (*Imagen*) y videos (*Video*). Cada uno de estos formatos tienen una propiedad *urlFormato* donde se indica la URL del recurso en internet.
- *OrdenPaso*: clase que indica el orden de paso indicada en su propiedad *orden*.
- *OrdenPasoRuta*: clase que indica el orden de paso de un punto de interés para una ruta.

Alguna de las jerarquías de clases no se han desarrollado porque no son exclusivas del dominio de las rutas y ya son tratadas en ontologías que en próximas versiones se indicarán (como los servicios ofrecidos), en otros casos se ha preferido esperar al desarrollo del prototipo para ir añadiendo clases y propiedades que deban ser incluidas en el dominio de la ontología.

Otra clase que se ha incluido en esta versión se han importado de w3.org [http://www.w3.org/2003/01/geo/wgs84\\_pos#](http://www.w3.org/2003/01/geo/wgs84_pos#) para la localización geográfica de los puntos de interés, servicios, etc. Se considera de gran importancia la reutilización de ontologías permitiendo una estandarización de las ontologías, en este caso tenemos una ontología madura y que ya está siendo utilizada por otras ontologías. El espacio de nombres usado es “geo”:

- *Point*: punto geográfico caracterizado por su latitud, longitud y altitud.

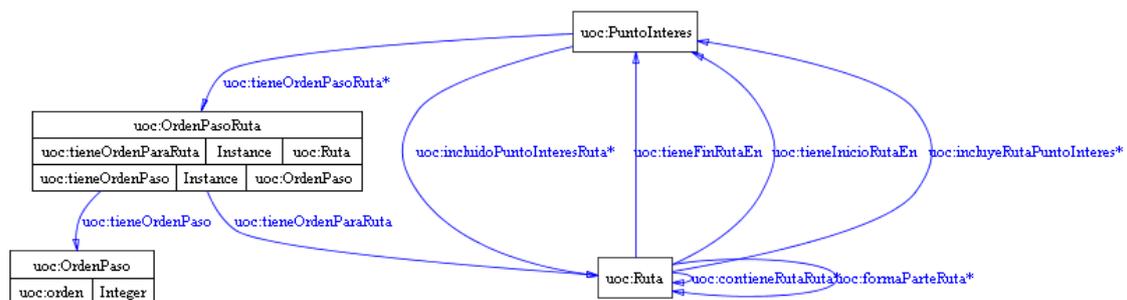
Las propiedades de objeto más relevantes que se han incluido en esta versión son las siguientes:

- *formaParteRuta*: una ruta puede formar parte de otra ruta. Propiedad transitiva, puesto que si una ruta r1 forma parte de otra ruta r2 y esta a su vez forma parte de r3, entonces r1 también formará parte de la ruta r3.
- *contieneRutaRuta*: una ruta puede contener otras rutas. Esta propiedad es inversa a *formaParteRuta*, si una ruta r1 forma parte de una ruta r2 entonces r2 contiene a la ruta r1.
- *tieneServicioEnRuta*: un servicio estará en una o varias rutas. Esta es la propiedad inversa de *tieneRutaServicio*.
- *tieneRutaServicio*: una ruta puede tener algún servicio incluido. Esta propiedad es inversa funcionalmente a *estaServicioEnRuta*.
- *incluyeRutaPuntoInteres*: todas las rutas incluyen algún punto de interés. Propiedad inversa de *incluyeRutaPuntoInteres*.

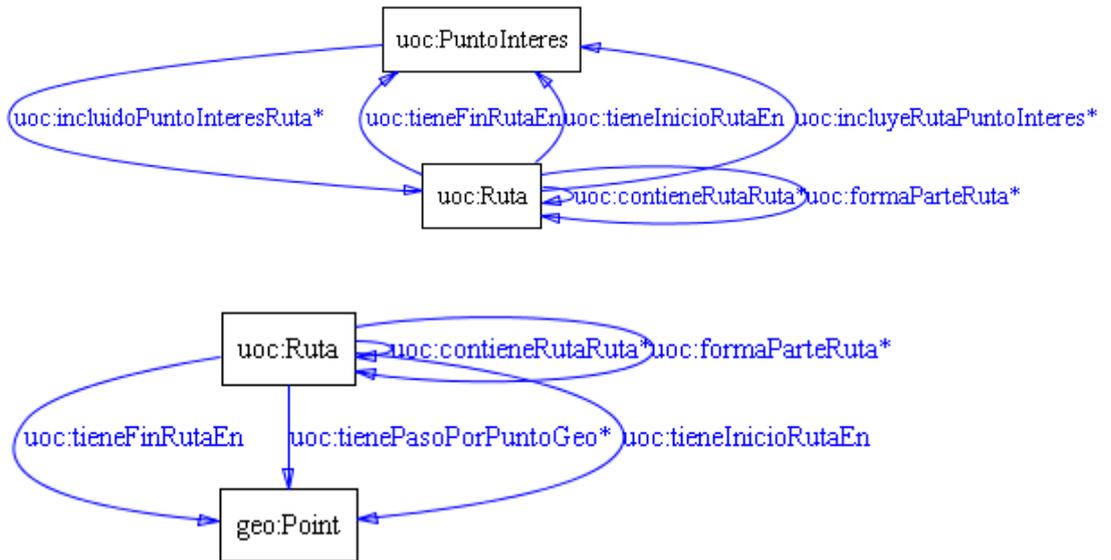
- *incluyeRutaPuntoInteres*: todo punto de interés se encuentra en alguna ruta. Propiedad inversa de *incluyeRutaPuntoInteres*.
- *situadoEnPosicionGeo*: los puntos de interés y los servicios estarán situados en un punto geográfico.
- *tienePasoPorPuntoGeo*: una ruta puede incluir varios puntos geográficos marcados.
- *tienePuntoInteresRecursoInternet*: un punto de interés debe tener al menos un recurso en internet. Propiedad inversa de *seRefiereFormatoPuntoInteres*.
- *seRefiereFormatoPuntoInteres*: un recurso de internet se puede referir a uno o más puntos de interés. Propiedad inversa con *tienePuntoInteresRecursoInternet*.
- *tieneFinRutaEn*: toda ruta tiene un punto geográfico o de interés que será marcado como el inicio de la misma.
- *tieneInicioRutaEn*: toda ruta tiene un punto geográfico o de interés que será marcado como el inicio de la misma.
- *tieneOrdenPasoRuta*: un punto de interés o posición geográfica tiene una orden de paso por una ruta.
- *tieneOrdenPaso*: el orden de paso de una ruta debe tener un orden de paso.
- *tieneOrdenParaRuta*: el orden de paso de una ruta debe tener una ruta.

Alguna de las clases presentadas y las propiedades de objeto que las relaciones se muestran a continuación:

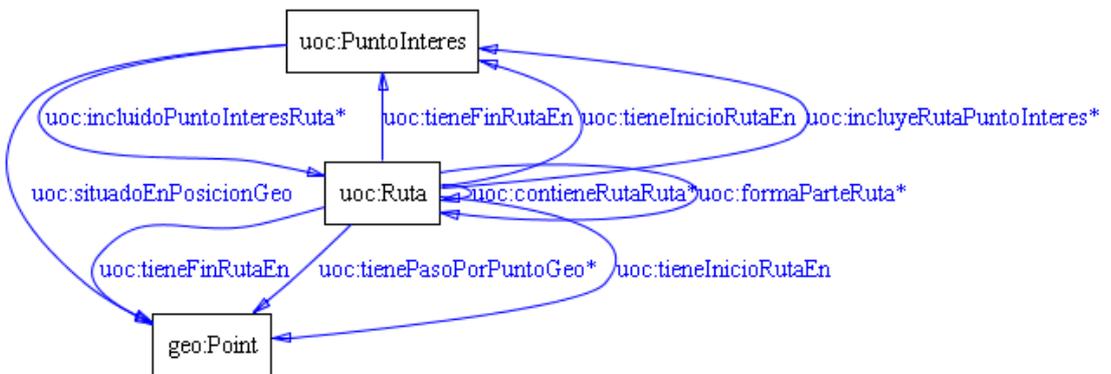
- Orden de paso de en una ruta: para indicar el orden en que se encuentra un punto de interés dentro de una ruta ha sido necesario crear la clase intermedia *OrdenPasoRuta* que gracias a las propiedades *tieneOrdenPasoRuta* ( $\leftarrow$  *PuntoInteres*), *tieneOrdenPaso* ( $\rightarrow$  *OrdenPaso*) y *tieneOrdenParaRuta* ( $\rightarrow$  *Ruta*) podemos asociar el orden de paso de un punto de interés a una ruta:



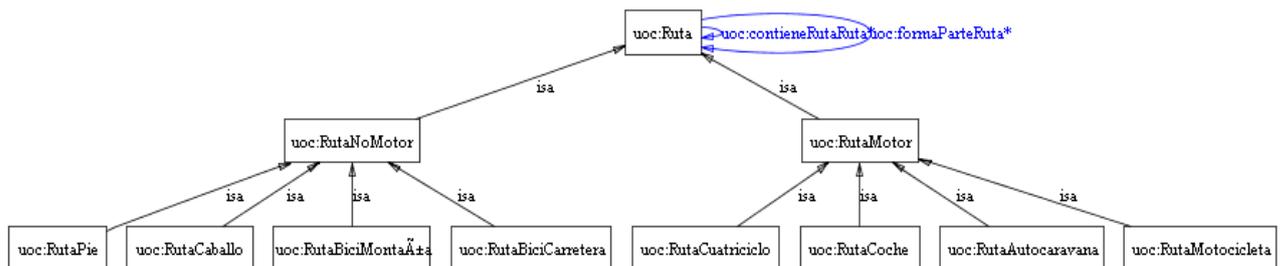
- Puntos de interés y Rutas: las rutas contienen puntos de interés y pueden contener otras rutas, también pueden tener asociadas distintas localizaciones geográficas, además puede terminar o comenzar en un punto de interés o localización geográfica:



- Todos los puntos de interés y los servicios están situados en algún punto geográfico y las rutas pueden contener, iniciarse o finalizar en uno de éstos puntos:



- Disponemos de distintos tipos de rutas:



La jerarquía completa de la ontología se puede ver en la siguiente imagen:



La propiedad de Ruta que indica si es circular tiene asociadas reglas definidas mediante SWRL, de tal manera que si en inicio y el final (y viceversa) de una ruta coinciden entonces la ruta será circular y si la ruta es circular esto implicará que el inicio y el final de la ruta serán el mismo:

<input checked="" type="checkbox"/>	<a href="http://www.uoc.com/reglaFinInicioFinRutaIguar">http://www.uoc.com/reglaFinInicioFinRutaIguar</a>	$\rightarrow$	$uoc:esCircular(?x, true) \wedge uoc:tieneFinRutaEn(?x, ?y) \rightarrow uoc:tieneInicioRutaEn(?x, ?y)$
<input checked="" type="checkbox"/>	<a href="http://www.uoc.com/reglaInicioFinRutaIguar">http://www.uoc.com/reglaInicioFinRutaIguar</a>	$\rightarrow$	$uoc:esCircular(?x, true) \wedge uoc:tieneInicioRutaEn(?x, ?y) \rightarrow uoc:tieneFinRutaEn(?x, ?y)$
<input checked="" type="checkbox"/>	<a href="http://www.uoc.com/reglaRutaCircular">http://www.uoc.com/reglaRutaCircular</a>	$\rightarrow$	$uoc:tieneFinRutaEn(?x, ?y) \wedge uoc:tieneInicioRutaEn(?x, ?y) \rightarrow uoc:esCircular(?x, true)$

Las instancias que se han creado han sido de prueba, por lo que no se ha creído de interés incluirlas en este documento y han sido eliminadas del proyecto de Protégé, en la próxima entrega se añadirán varias instancias de rutas reales.

## 14. INTRODUCCIÓN AL PROTOTIPO

El prototipo del proyecto consiste en una web que permita a los usuarios hacer uso de la ontología desarrollada. El ámbito de la ontología es el de las rutas turísticas por lo que la web estará proporcionará al usuario una funcionalidad relacionada con este ámbito.

En esta entrega se ha desarrollado todo el aplicativo relacionado con la estructura de la ontología, es decir, con el manejo de la clases y propiedades de las mismas. En concreto se presenta al usuario un filtro donde puede introducir sus preferencias, estas preferencias están directamente relacionadas con las definiciones o clases de la ontología.

## 15. DATOS TÉCNICOS DEL PROTOTIPO

Para realizar el desarrollo se ha seleccionado como principal lenguaje de programación Java en su versión 1.6. Se ha elegido este lenguaje ya que es uno de los más usados en desarrollos web dinámicos, proporciona variedad de librerías relacionadas con XML y Web Semántica y adicionalmente tengo experiencia con el mismo.

Además se ha desarrollado en JSP y JSTL. Mediante JSP se pueden realizar páginas HTML de manera dinámica, pero independizando la vista del modelo. JSTL nos permite crear componentes HTML más complejos de manera estándar .

Por último se usa algo de Java Script para añadir comportamiento dinámico en el lado del cliente.

Para el manejo de la ontología (carga en memoria del fichero .owl con la definición, acceso a las clases y sus propiedades, etc.) se ha seleccionado Jena en su versión 2.6.4 (Última disponible en el momento de iniciar el desarrollo) dentro del abanico de posibles opciones como Sesame, OWLAPI, Pellet, etc. Se ha seleccionado Jena puesto que sigue las recomendaciones de la W3C, permite consultas RDF, RDFS, OWL y

SPARQL, se encuentra actualizado y tiene una nutrida documentación y el código es de acceso gratuito (licencia Open Source Initiative).

El Desarrollo de la aplicación se ha realizado con el IDE Eclipse (Helios Service Release 2) como un proyecto web dinámico para ser usado en un servidor web Apache Tomcat 5.5 con un JRE versión 6.

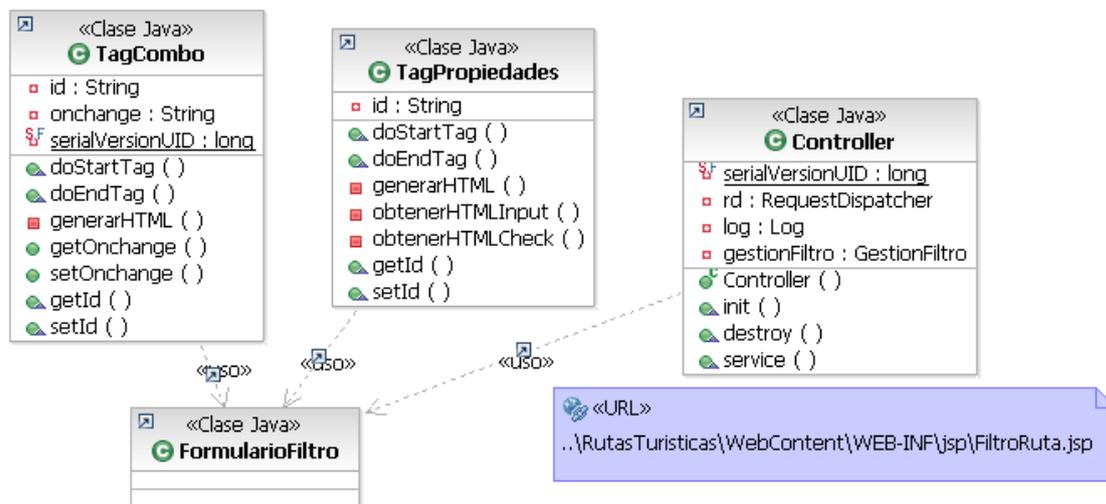
## 16. DISEÑO DE LA APLICACIÓN

La aplicación web se ha diseñado siguiendo un patrón MVC simplificado con una clase Servlet controladora encargada de atender las peticiones HTTP del lado cliente y servir la respuesta.

### 3.1 VISTA

En la vista se dispone de una JSP (FiltroRuta.jsp) encargada de generar dinámicamente las páginas de filtro, esta JSP se ayuda de dos componentes JSTL (ambos extienden la clase TagSupport):

- TagCombo (uoc.pfc.ws.tag) encargado de generar el tag Select de HTML a partir del listado en sesión de elementos indicado mediante su propiedad id. Estas listas de sesión se corresponden con las clases del modelo ontológico.
- TagPropiedades (uoc.pfc.ws.tag) encargado de mostrar las propiedades de un elemento de sesión. Estas propiedades se corresponden con las propiedades de una clase del modelo identificado por la propiedad id del componente.



Para relacionar la vista con el modelo de la ontología se dispone de la estructura de dato uoc.pfc.ws.to.FormularioFiltro, esta estructura (TO) se encuentra almacenado en la sesión del usuario, lo que permite mantener en memoria la selección que el usuario vaya realizando del filtro de selección, con cada selección del usuario se irá rellenado, por un lado con la lista de clases o propiedades que existan en el modelo y por otro con la selección que sobre estos haga.

Uno de los objetivos del proyecto es que sea lo más dinámico posible. Así el diseño se compone de tres apartados fijos correspondientes a las clases principales de la ontología:

- Tipo y sub-tipo de ruta.
- Tipos de servicios disponibles.
- Tipos de puntos de interés.

Dentro de estos tres apartados se carga de manera dinámica las opciones disponibles que se cargan de manera dinámica según la selección que se haya realizado (de cada uno de los tipos enumerados más arriba).

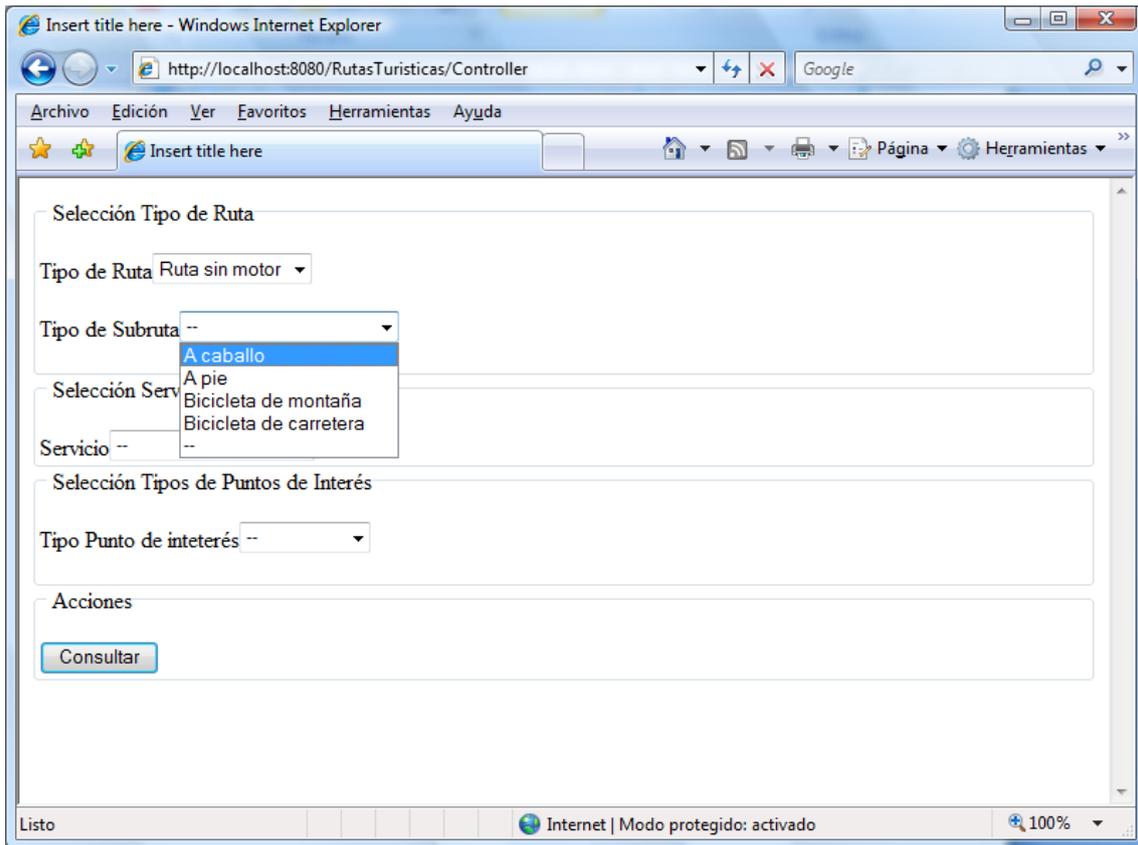
Así cuando el usuario entra en la aplicación se encuentra con la siguiente pantalla:

The screenshot shows a web browser window with the following elements:

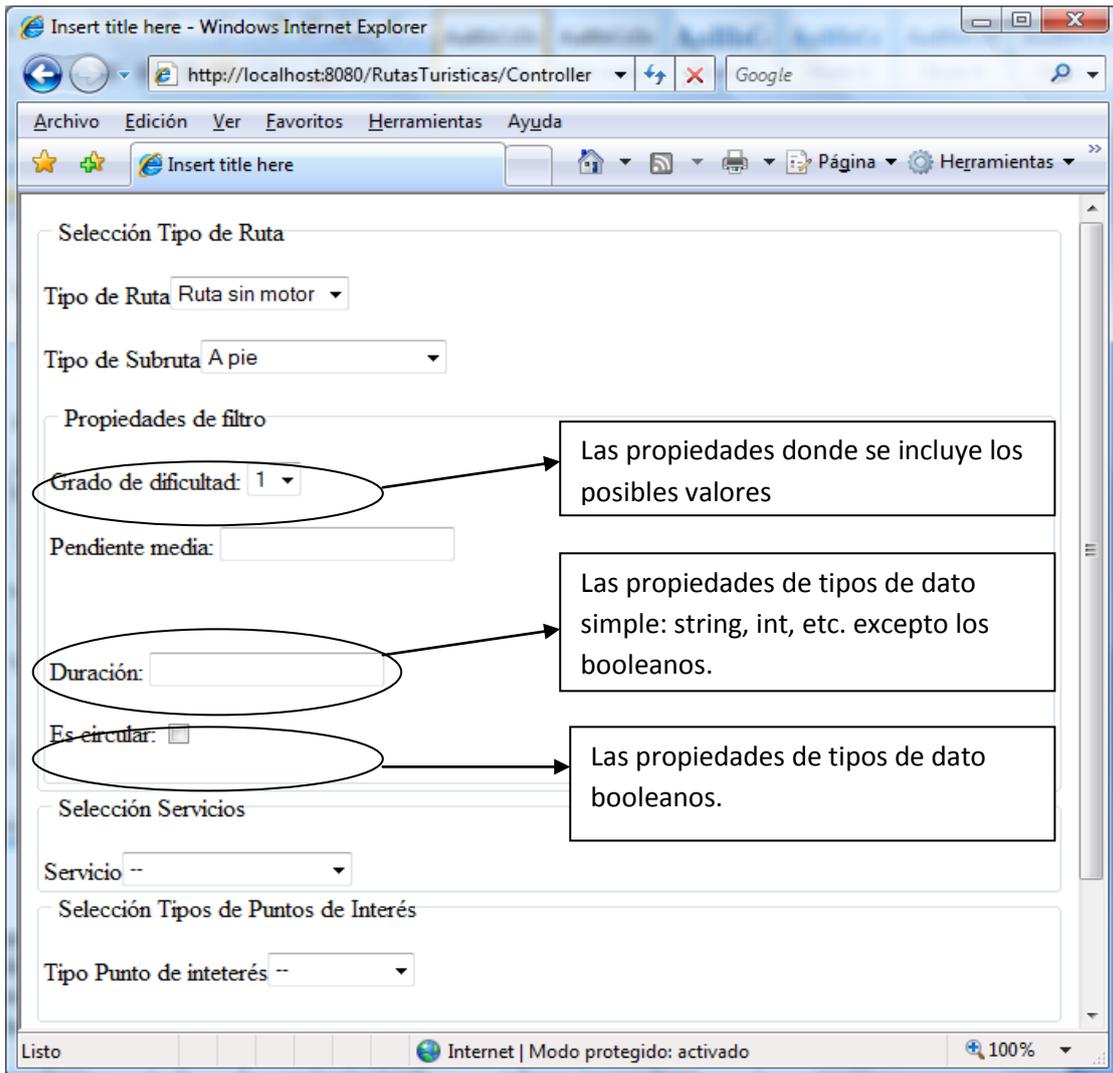
- Browser title: Insert title here - Windows Internet Explorer
- Address bar: http://localhost:8080/RutasTuristicas/Controller
- Menu bar: Archivo, Edición, Ver, Favoritos, Herramientas, Ayuda
- Page title: Insert title here
- Form sections:
  - Selección Tipo de Ruta:** Includes dropdowns for 'Tipo de Ruta' and 'Tipo de Subruta'.
  - Selección Servicios:** Includes a dropdown for 'Servicio'.
  - Selección Tipos de Puntos de Interés:** Includes a dropdown for 'Tipo Punto de interés'.
  - Acciones:** Includes a 'Consultar' button.
- Status bar: Listo, Internet | Modo protegido: activado, 100%

Donde el tipo de ruta, de servicio y de puntos de interés ya se han cargado de manera dinámica del modelo, dando opción al usuario de realizar la selección según sus preferencias. Estos datos estarán almacenados en las listas correspondientes de la clase FormularioFiltro.

Si selecciona un tipo de ruta dinámicamente se actualizarán los sub tipos de ruta (cargándose en la sesión del usuario junto a la selección del tipo de ruta realizada):



Cuando seleccione el sub tipo de ruta además se le proporcionará una lista de las propiedades asociadas a esta clase:



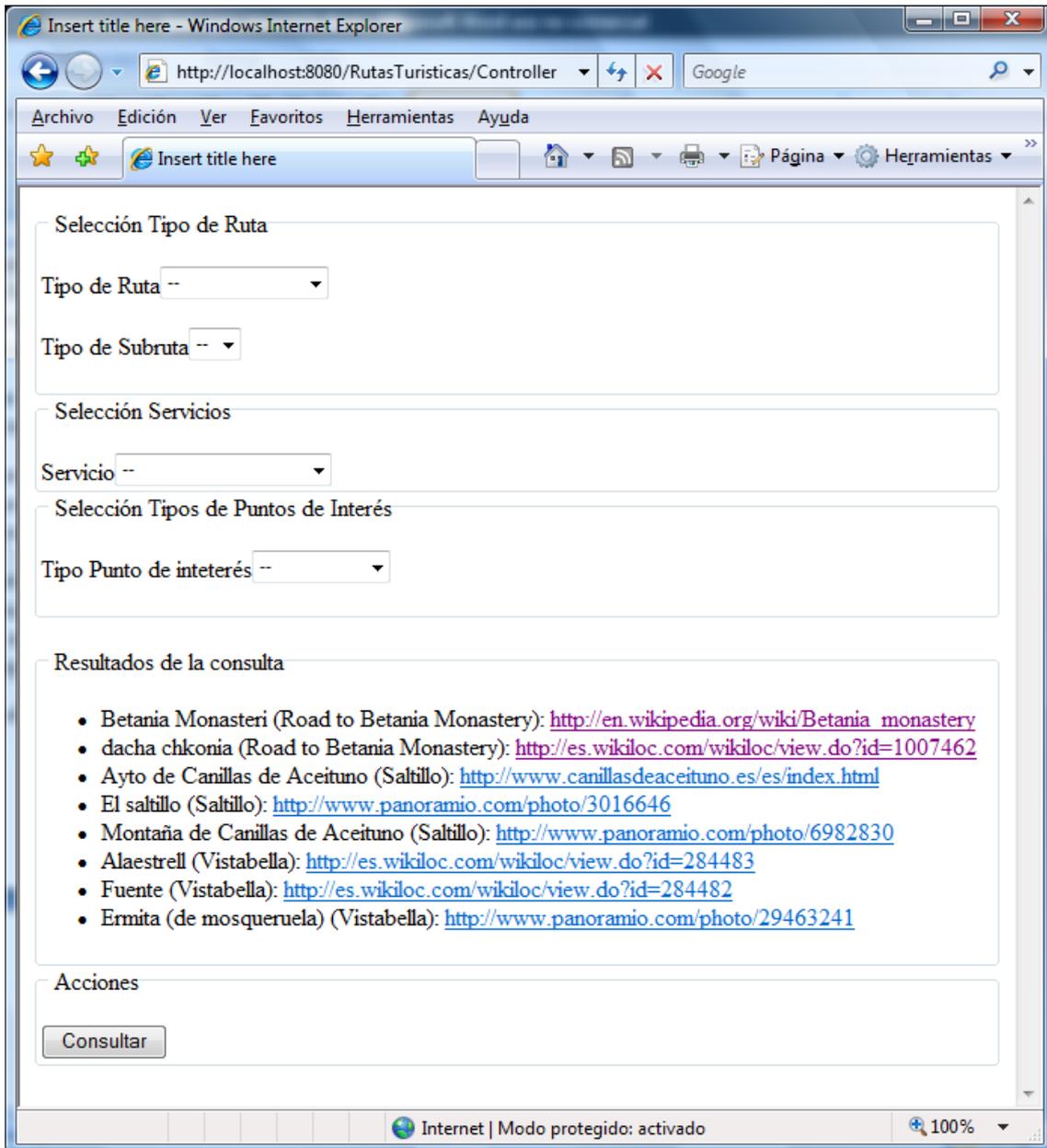
Según el tipo que sea la propiedad así se mostrará un elemento HTML u otro. Si es una cadena o un entero se mostrará una caja de texto, en caso de ser un valor lógico se muestra un check-box y si es una propiedad donde se ha incluido una lista de valores a seleccionar se muestra como una Select con dichos valores.

Los valores que incluya el usuario siguiendo sus preferencias se almacenarán en sesión para su posterior consulta.

Como se puede ver la gestión del modelo es dinámico de tal manera que si se añaden Tipos de rutas, servicios o puntos de interés así como las propiedades asociadas con los mismos se verán reflejados en la vista sin necesidad de modificar la aplicación.

Hasta ahora se ha visto como se ha generado de manera dinámica el filtro para que el usuario seleccione sus preferencias. Mediante el botón "Consultar" que se encuentra en el apartado de acciones el usuario puede consultar las rutas que se encuentran disponibles en el sistema en forma de instancias de la ontología.

Si el usuario no ha realizado un filtro previo se le mostrarán todas las rutas disponibles:



En el apartado “Resultados de la consulta” puede verse un listado con los puntos de interés y entre paréntesis a su lado el nombre de la ruta al que pertenece y por último la url donde se encuentra el recurso asociado.

Si el usuario selecciona ahora un Tipo de ruta y de sub ruta (se hace la consulta filtrando por tipo de sub ruta) solo aparecerán aquellos puntos de interés de ese subtipo:

Insert title here - Windows Internet Explorer

http://localhost:8080/RutasTuristicas/Controller

Google

Archivo Edición Ver Favoritos Herramientas Ayuda

Insert title here

Tipo de Ruta Ruta sin motor

Tipo de Subruta A pie

Propiedades de filtro

Grado de dificultad: 1

Pendiente media:

Duración:

Es circular:

Selección Servicios

Servicio --

Selección Tipos de Puntos de Interés

Tipo Punto de interés --

Resultados de la consulta

- Ayto de Canillas de Aceituno (Saltillo): <http://www.canillasdeaceituno.es/es/index.html>
- El saltillo (Saltillo): <http://www.panoramio.com/photo/3016646>
- Montaña de Canillas de Aceituno (Saltillo): <http://www.panoramio.com/photo/6982830>

Internet | Modo protegido: activado 100%

### 3.3 CONTROL

Al ser una aplicación pequeña donde la navegación web y el negocio no es muy complejo se ha usado un Servlet (`uoc.pfc.ws.controller.Controller`) para realizar todas las funciones de controlador sin existir un segundo nivel de manejo de peticiones.



Dependiendo de la acción realizada por el usuario el Servlet redirigirá la petición al método de negocio contenido en la clase que agrupa el negocio (uoc.pfc.ws.negocio.GestionFiltro). Adicionalmente se encargará de transformar los datos de entrada (procedentes de la vista) a datos de negocio, y viceversa, además de proporcionar la respuesta al usuario (manejando la sesión y la navegación).

La primera vez que el usuario entra en la aplicación no hay acción como tal, pero este Servlet realiza una función muy importante: carga el modelo del fichero (.owl) en memoria. Esta carga solamente se hace una vez para cada usuario, se mantiene en memoria en la instancia del Servlet del usuario en su propiedad gestionFiltro del tipo GestionFiltro, cuando se llama al constructor del negocio.

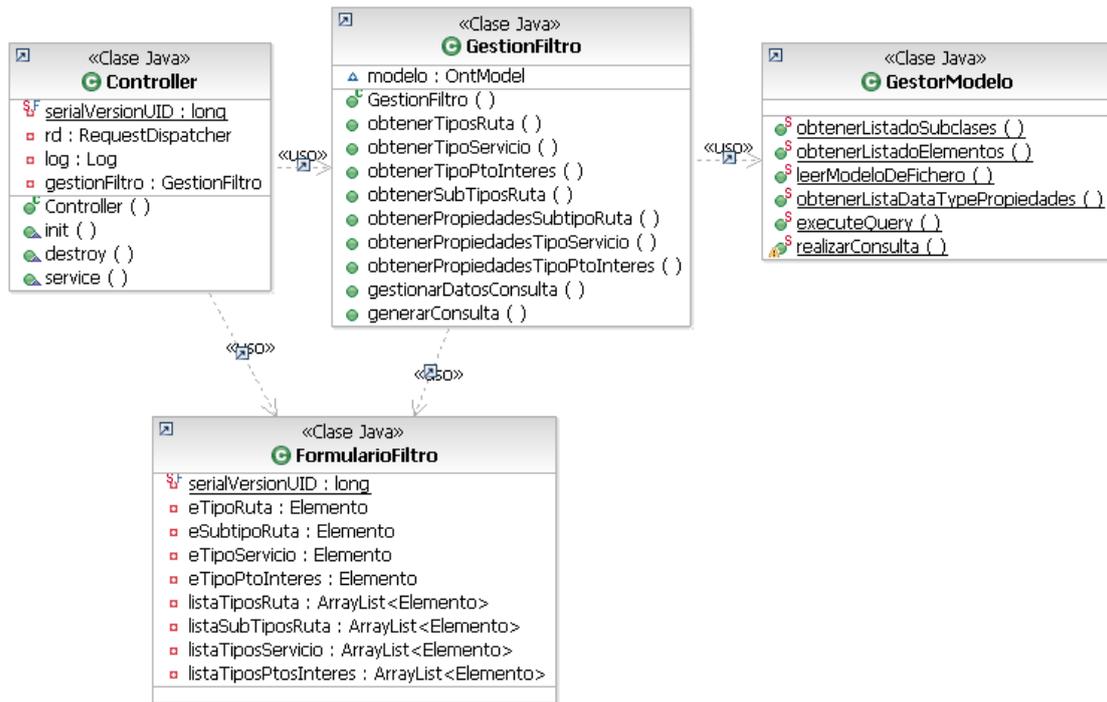
El controlador recibe en la request el parámetro “accion”.

La clase uoc.pfc.ws.util.Constantes se mantienen las distintas acciones disponibles.

### 3.3 MODELO

Como se ha indicado más arriba la clase encargada de concentrar el negocio del filtro es uoc.pfc.ws.negocio.GestionFiltro, esta clase se comunica con la clase que gestiona el modelo de la ontología (uoc.pfc.ws.manager.GestorModelo) para acceder al modelo a bajo nivel.

Como se puede comprobar en las clases indicadas en la parte de control no se accede a clases del modelo (las que proporciona la API de Jena) y en la clase de negocio solamente se acceden de estas a las que corresponde a las clases y al modelo.



Para mantener el modelo genérico de datos de la aplicación de una manera independiente se han incluido las estructuras de datos:

- uoc.pfc.ws.to.Elemento: representación genérica de los elementos disponibles en la ontología. Sus propiedades son el nombre (correspondiente al nombre de la clase o propiedad), etiqueta (correspondiente a la etiqueta de la clase o propiedad), valor (de tipo Valor para las propiedades) y listaPropiedades (con la lista de propiedades de una clase) que se identifican por el nombre de la propiedad por cada propiedad que tiene el valor en un objeto de la clase Elemento (esto permite que sea muy dinámico a la hora de mantener una estructura de datos, aunque añade complejidad y cierta redundancia en algunas ocasiones).
- uoc.pfc.ws.to.Valor: contiene las propiedades valor (de tipo genérico) y tipo (de tipo cadena) para poder almacenar cualquier valor de tipo simple de la ontología y el tipo del mismo.



Esta estructura al mismo tiempo que sencilla proporciona un gran potencial para representar la arquitectura de la ontología.

Para acceder a la arquitectura o definición del modelo (la ontología definida) se dispone de la clase `uoc.pfc.ws.manager.GestionModelo`, que proporciona la funcionalidad de carga de acceso al modelo en un fichero físico del servidor a memoria, acceso a dicha estructura (acceso a una clase por su nombre, acceso a las subclases de una clase, acceso a las propiedades de una clase y ejecución de una consulta).

### 3.4 ACCESO INSTANCIAS

El acceso a las instancias se realiza mediante consultas SPARQL. La funcionalidad actual permite obtener los puntos de interés con la ruta a la que está asociada y la URL en la cual contiene recursos asociados en internet. También se filtra la duración máxima que debe tener las rutas buscadas.

La consulta se genera dinámicamente según la selección del usuario. Si no hay preferencias la consulta tendrá como resultado todos los puntos de interés existentes. Debe tenerse en cuenta que por razones de tiempo no se ha refinado mucho la consulta, el objetivo es usar SPARQL para acceder a las instancias de manera introductoria.

Para una consulta genérica tenemos la siguiente sentencia:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX uoc: <http://www.uoc.com/pfc_otg.owl#>
SELECT ?nombreRuta ?nombrePtoInt ?url
WHERE {
?class uoc:incluyeRutaPuntoInteres ?ptoInt .
?ptoInt uoc:tienePuntoInteresRecursoInternet ?recurso .
?recurso uoc:urlFormato ?url .
?class uoc:nombreRuta ?nombreRuta .
?ptoInt uoc:nombre ?nombrePtoInt
} ORDER BY ?nombreRuta

```

Se obtiene el nombre de la ruta (*?nombreRuta*), el nombre del punto de interés (*?nombrePtoInt*) y la url (*?url*) del recurso asociado ordenados por el orden de la ruta (*ORDER BY ?nombreRuta*).

Si el usuario selecciona un tipo de ruta (por ejemplo, una ruta pie) se debe añadir la siguiente línea a la sentencia anterior, dentro del bloque de las condiciones (where):

```
?class a uoc:RutaPie .
```

Por último, un ejemplo de filtro: si el usuario introduce una duración (de valor 12345) se construirá la consulta añadiendo dentro del bloque de condiciones (where):

```
FILTER(?duracion <= 12345 )
```

Quedando la consulta completa:

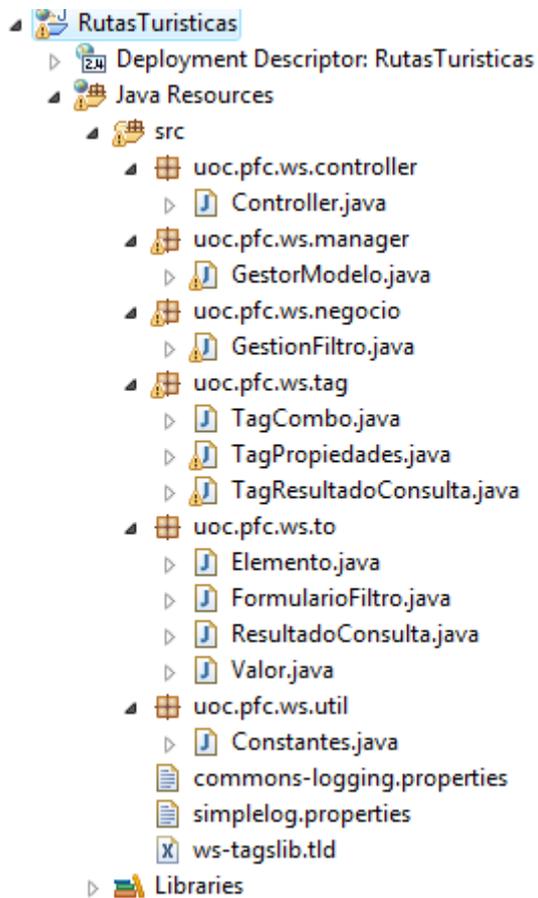
```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX uoc: <http://www.uoc.com/pfc_otg.owl#>
SELECT ?nombreRuta ?nombrePtoInt ?url WHERE {
?class a uoc:RutaPie .
?class uoc:duracion ?duracion .
?class uoc:incluyeRutaPuntoInteres ?ptoInt .
?ptoInt uoc:tienePuntoInteresRecursoInternet ?recurso .
?recurso uoc:urlFormato ?url .
?class uoc:nombreRuta ?nombreRuta .
?ptoInt uoc:nombre ?nombrePtoInt
FILTER(?duracion <= 12345 )}
ORDER BY ?nombreRuta

```

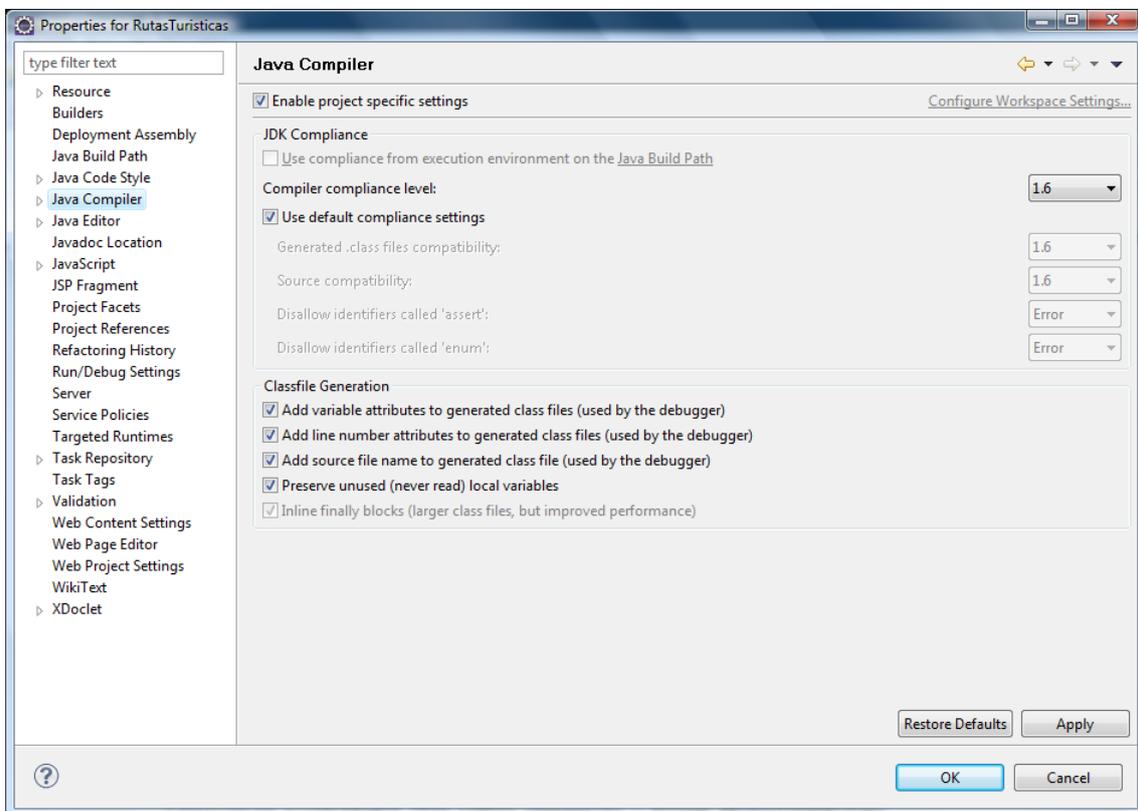
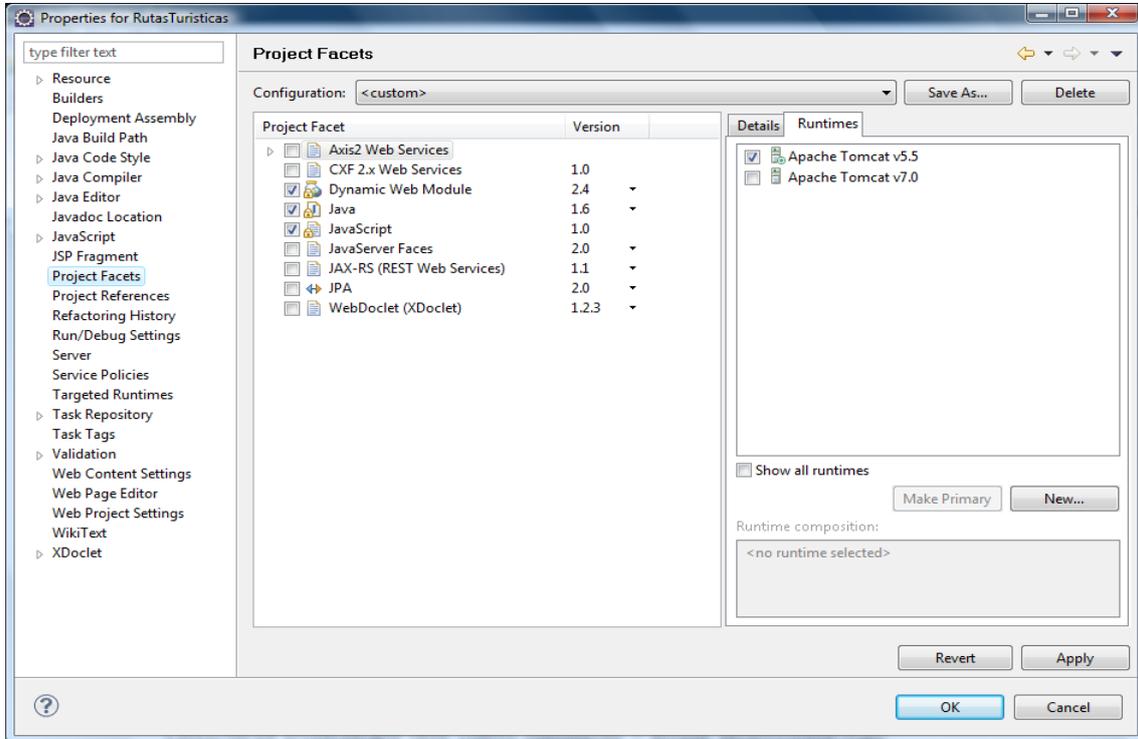
### 3.5 ESTRUCTURA DE CLASES

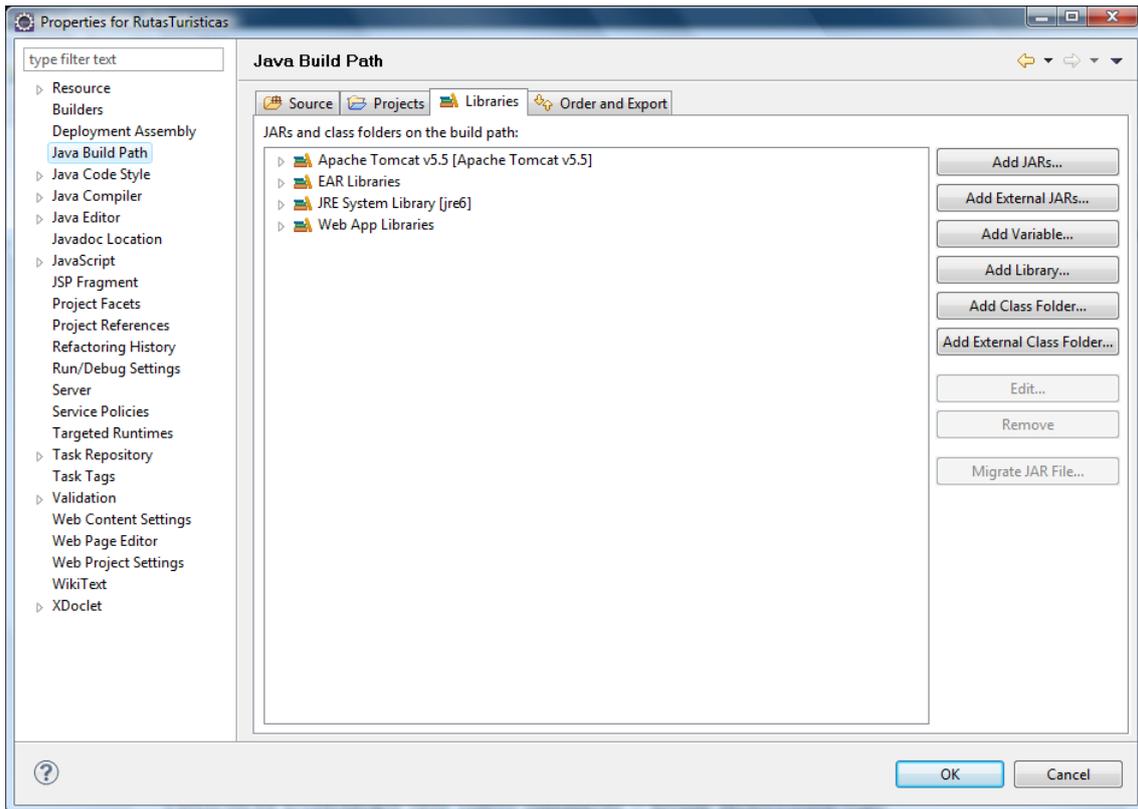
En el proyecto web podemos ver las clases empaquetadas de la siguiente manera (tal y como se ha visto en los apartados anteriores):



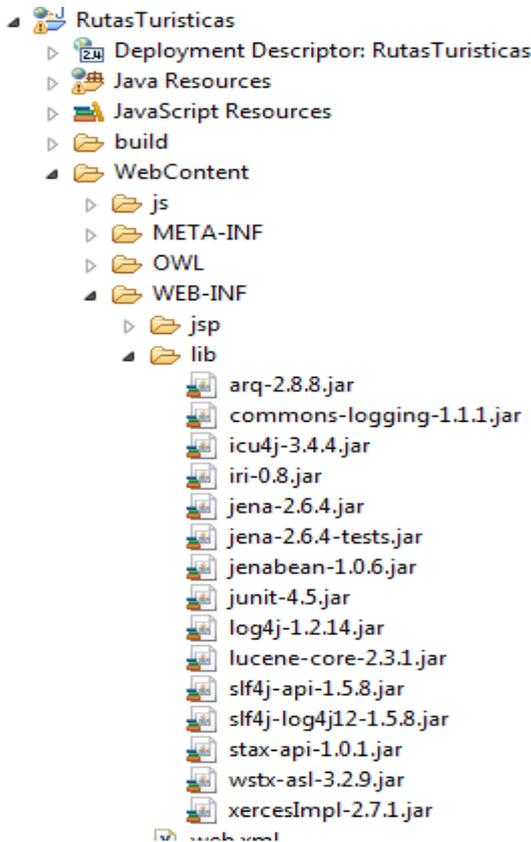
## 17. CONFIGURACIÓN Y ENTREGA PROTOTIPO

En las decisiones técnicas ya se ha indicado el IDE, servidor, etc. usados, en la siguiente imagen podemos ver las facetas y las clases de referencia:





Las librerías que necesita el prototipo se han incluido a nivel de aplicación, puesto que no se tiene acceso a un servidor genérico se ha decidido incluir las clases en el lib del proyecto:



Todas ellas disponibles en la web Source Forge de Jena:

<http://jena.sourceforge.net/downloads.html> Estas librerías no se incluyen en la entrega ya que su tamaño impide que se realice el envío (las que se han usado son para Windows de 32 bits). La entrega está compuesta por el presente documento y el proyecto en dos formatos: en un fichero .war que está generado por Eclipse de manera óptima para Tomcat 5.5, que contiene los .java en los mismos directorios que los .class y un .rar donde se encuentra la estructura original del proyecto desarrollado. Para acceder con el navegador web se debe acceder a la vía de acceso /EstadísticasDiarias (si se mantiene la configuración del proyecto) junto al servidor y el puerto (en caso de ser un servidor local el servidor será localhost y el puerto 8080).

Toda la documentación usada para el desarrollo del prototipo se encuentra en el API de Jena. También puede encontrarse documentación en el proyecto Source Forge de Jena en <http://jena.sourceforge.net/> y en la página de los desarrolladores (Stamford) en <http://protege.stanford.edu/>.

## 18. FUTUROS DESARROLLOS

Dada la limitación temporal y de recursos de que se disponen para el desarrollo del proyecto queda abierta la posibilidad de futuras mejoras y ampliaciones. A continuación se proponen una serie de modificaciones posibles:

- Mejora en la consulta del prototipo: la consulta existente tiene en cuenta el tipo de ruta seleccionado por el usuario y la duración máxima de ésta. En una futura implementación podría tenerse en cuenta el resto de preferencias del usuario como la existencia de algún tipo de punto de interés o servicio y el valor de propiedades. También se podría ampliar la información mostrada al usuario y la forma en que ésta se presenta.
- Administración del prototipo e instancias a través del prototipo: el prototipo actual permite consultar el modelo y sus instancias, pero no permite su administración, por lo que cabe añadir una funcionalidad que permite añadir y modificar instancias y otra que permita modificar el modelo, por ejemplo la jerarquía de la clase “Ruta”, puesto que la implementación actual lo permite sin ser modificada.
- Selección del idioma: gracias a la etiqueta label de las clases y propiedades, que permite asociarle un texto descriptivo a cada una de ellas en función de un idioma, sería sencillo añadir la opción de selección del idioma al usuario para consultar la etiqueta label de cada elemento mostrado de la ontología. Actualmente las etiquetas solo tienen un texto para español por lo que solo muestra la vista en este idioma. Por lo que en la consulta de la etiqueta de las clases y las propiedades se proporciona este lenguaje.
- Aumentar la persistencia: actualmente la persistencia para el usuario está ligada con la sesión, excepto la definición de la ontología y las instancias de la misma. El usuario al salir de la aplicación pierde todas sus preferencias, por lo que podría ampliarse la funcionalidad para que no se pierda esta información.
- Funcionalidad para Servicios Web: puesto que uno de los principales objetivos de la web semántica es la interacción entre máquinas se podría añadir dicha funcionalidad. Sería necesario incluir una interface con la funcionalidad

proporcionada, que podemos encontrar en la clase GestionFiltro, y un descriptor del servicio web (WSDL).

- Reutilización de ontologías: ya se ha reutilizado la ontología que define una posición geográfica. Sin embargo, con la cantidad de ontologías disponibles en el ámbito del turismo y de las rutas es posible reutilizar parte de éstas. Habría que realizar un análisis más exhaustivo de dichas ontologías y fijar unas preferencias (si se quiere ampliar la ontología definida, si se quiere estandarizar, etc.) para elegir cuales usar y como.
- Añadir capa de seguridad: sería posible añadir a la definición de la ontología un nivel de seguridad que permita acceder a los datos según el perfil del usuario que esté accediendo.

## 19. ENTREGA FINAL

La entrega final del proyecto contiene los siguientes componentes:

- Memoria final: presente documento (jparragau\_memoria) tanto en formato .docx como .pdf.

Dentro del fichero comprimido jparragau\_entregafinal:

- Ontología: en el directorio /Ontología se encuentran los ficheros de la definición de la ontología.
- Ficheros comprimidos (.rar y .war) RutasTuristicas con el prototipo.
- Memoria final: también se incluye dentro del fichero comprimido.

## 20. BIBLIOGRAFÍA

### Web semántica

- Corchuelo, R. (19/01/2007). “Introducción a la Web Semántica”. Universidad de Sevilla, Depto. de Lenguajes y Sistemas informáticos. [Fecha de consulta: 25/05/2011].  
[www.tdg-seville.info/Download.ashx?id=60](http://www.tdg-seville.info/Download.ashx?id=60)
- Honrubia López, F.J. (25/11/2002) “Introducción a las Ontologías”. Escuela Universitaria Politécnica de Albacete. [Fecha de consulta: 25/05/2011].  
<http://www.dsi.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-Ontologias.pdf>
- VV.AA. (2006). “Sistemas de información: nuevos escenarios basados en ontologías”. Revista de Gestão da Tecnologia e Sistemas de Informação. Universidad Nacional de Santiago del Estero. [Fecha de consulta: 25/05/2011].  
<http://www.jistem.fea.usp.br/index.php/jistem/article/download/10.4301%252FS1807-17752006000100001/42>  
ISSN online: 1807-1775
- Sanz, I., Jiménez-Ruiz, E. (05/2007). “Ontologías en informática”. Universitat Jaume I. [Fecha de consulta: 25/05/2011].  
<http://krono.act.uji.es/publications/techrep/Book-Chapter-Protege-report2007.pdf>

- Teso del, E. “Tecnologías XML y Web Semántica”. Universidad de Oviedo. [Fecha de consulta: 25/05/2011].  
[www.di.uniovi.es/~labra/cursos/vero6/pres/Ontologias.pdf](http://www.di.uniovi.es/~labra/cursos/vero6/pres/Ontologias.pdf)
- VV.AA. (2007). “Los lenguajes de representación semántica y su uso en la construcción de ontologías”. Revista de Ciencias Sociales, Universidad del Zulia. [Fecha de consulta: 25/05/2011].  
<http://redalyc.uaemex.mx/pdf/280/28013105.pdf>  
ISSN (Versión impresa): 1315-9518
- Castells, Pablo. “La web semántica”. Escuela Politécnica Superior. Universidad Autónoma de Madrid. [Fecha de consulta: 25/05/2011].
- [8] VV.AA. (2005). “Semantic Web”. [Fecha de consulta: 25/05/2011]. W3.org.  
<http://www.w3.org/2005/Talks/1111-Delhi-IH/>
- Siegel, D. (14/12/2010). “What’s the Semantic web?” Semantic Focus. [Fecha de consulta: 25/05/2011].  
<http://www.semanticfocus.com/blog/entry/title/introduction-to-the-semantic-web-vision-and-technologies-part-2-foundations/>
- Parada, R.A. “Introducción a la Web Semántica” [Fecha de consulta: 11/06/2011].  
[http://ramonantonio.net/contents/web\\_semantica](http://ramonantonio.net/contents/web_semantica)
- Berners-Lee, T. (17/05/2001). “The Semantic Web A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities”. Scientific American. [Fecha de consulta: 11/06/2011].  
<http://www.scientificamerican.com/article.cfm?id=the-semantic-web>
- Gruber, T.R. (1993). “Toward Principles for the Design of Ontologies Used for Knowledge Sharing”. Stanford Knowledge Systems Laboratory. [Fecha de consulta: 11/06/2011].  
<http://tomgruber.org/writing/onto-design.htm>
- Gruber, T.R. (1992). “What is an Ontology?” [Fecha de consulta: 11/06/2011]  
<http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>
- Gruber, T.R. (1993). “A Translation Approach to Portable Ontology Specifications”. [Fecha de consulta: 11/06/2011]  
[http://ksl-web.stanford.edu/KSL\\_Abstracts/KSL-92-71.html](http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html)
- VV.AA. (2005). “Semantic Web”. W3C (MIT, ERCIM, Keio). [Fecha de consulta: 11/06/2011]  
[http://www.w3.org/2005/Talks/1111-Delhi-IH/#\(1\)](http://www.w3.org/2005/Talks/1111-Delhi-IH/#(1))

### **Ontologías Turísticas**

- VV.AA. (24/03/2011). “Hacia la mejora de la creación de rutas turísticas a partir de información semántica”. V JORNADAS DE SIG LIBRE. Universitat de Girona. [Fecha de consulta: 05/06/2011].  
[www.sigte.udg.edu/jornadassiglibre/uploads/articulos/art13.pdf](http://www.sigte.udg.edu/jornadassiglibre/uploads/articulos/art13.pdf)
- VV.AA. (2007). “Tourism Ontology and Semantic Management System: State-of-arts analysis”. University of Innsbruck. [Fecha de consulta: 05/06/2011].  
[www.iadis.net/dl/final\\_uploads/200712C070.pdf](http://www.iadis.net/dl/final_uploads/200712C070.pdf)
- VV.AA. (2008). “Development and Alignment of a Domain-Specific Ontology for Question Answering”. Research Group in Computational Linguistics, University of Wolverhampton, UK, German Research Centre for Artificial

- Intelligence GmbH (DFKI), Germany. Fondazione Bruno Kessler – FBK, Italy. [Fecha de consulta: 05/06/2011].  
[www.lrec-conf.org/proceedings/lrec2008/pdf/561\\_paper.pdf](http://www.lrec-conf.org/proceedings/lrec2008/pdf/561_paper.pdf)
- Cardoso, J. (2005). "E-Tourism: Creating Dynamic Packages using Semantic Web Processes". Universidade da Madeira. [Fecha de consulta: 05/06/2011].  
<http://www.w3.org/2005/04/FSWS/Submissions/16/paper.html>
  - VV.AA. "Comparing Ontologies - A Case Study". University of Karlsruhe. [Fecha de consulta: 05/06/2011].  
<http://ontobroker.semanticweb.org/ontos/comparing.html>
  - VV.AA. (2007) "Proyecto Godo II". Consorcio Universidad Carlos III y Atos Origin. [Fecha de consulta: 24/04/2011].
  - Gutiérrez Losada, I. (2010). "Ontologías Turísticas Geográficas: Creación de una Ontología sobre Rutas Turísticas (a Pie o en Bicicleta) por Espacios Naturales". [Fecha de consulta: 05/06/2011].  
<http://hdl.handle.net/10609/2284>
  - VV.AA. (07/2008). "Qall-me Ontology". [Fecha de consulta: 05/06/2011].  
<http://qallme.fbk.eu/index.php?location=ontology>
  - VV.AA. "E-Tourism Working Group". E-Tourism. [Fecha de consulta: 05/06/2011].  
<http://e-tourism.deri.at/index.html>
  - VV.AA. (01/05/2009). "cDOTT". ISIS. [Fecha de consulta: 05/06/2011].  
[http://www.faw.uni-linz.ac.at/index.php?id=55&PROJECT\\_ID=107](http://www.faw.uni-linz.ac.at/index.php?id=55&PROJECT_ID=107)
  - VV.AA. (02/11/2007). "Ontología Cruzar. Proyecto 'Un visitante, una ruta'. Especificación". Fundación CTIC. [Fecha de consulta: 05/06/2011].  
<http://idi.fundacionctic.org/cruzar/turismo.html>
  - VV.AA. (2011). Harmonet.  
 [Fecha de consulta: 11/06/2011]  
<http://www.harmonet.org/>
  - VV.AA. (07/2008). Qall-me.  
 [Fecha de consulta: 11/06/2011]  
<http://qallme.fbk.eu/index.php?location=ontology>
  - VV.AA. (01/05/2009). cDOTT. Johannes Kepler Universitat.  
 [Fecha de consulta: 11/06/2011]  
[http://www.faw.uni-linz.ac.at/index.php?id=55&PROJECT\\_ID=107&no\\_cache=1](http://www.faw.uni-linz.ac.at/index.php?id=55&PROJECT_ID=107&no_cache=1)
  - VV.AA. [Fecha de consulta: 11/06/2011].  
<http://es.wikiloc.com/wikiloc/home.do>
  - VV.AA. [Fecha de consulta: 11/06/2011].  
<http://www.boletin-turistico.com/diccionarioturismo>
  - VV.AA. [Fecha de consulta: 11/06/2011].  
<http://deconceptos.com/ciencias-sociales/ruta-turistica>
  - VV.AA. [Fecha de consulta: 11/06/2011].  
[http://www.autocity.com/tramites\\_dgt/legislacion/reglamento\\_vehiculos/anexo2.html](http://www.autocity.com/tramites_dgt/legislacion/reglamento_vehiculos/anexo2.html)

### **Creación de la ontología**

- VV.AA. (2001). "Ontologías en documentación". Universidad Politécnica de Valencia. [Fecha de consulta: 05/06/2011].  
<http://personales.upv.es/ccarrasc/doc/2001-2002/ontologias/INICIO.htm>
- VV.AA. (03/2001). "Ontology Development 101. A Guide to Creating Your First Ontology". Stanford University. [Fecha de consulta: 05/06/2011].

- <http://www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html>
- VV.AA. (04/2008). “Semantic Web for the Working Ontologist. Effective Modeling in RDFS and OWL”. Ed. Morgan Kauffmann. ISBN: 978-0-12-373556-0.
  - Antoniou, G. (2010). “Manual de Web Semántica”. Grigoris. Editorial Comares.
  - VV.AA. (10/02/2004). “RDF Primer”. W3C 2004. [Fecha de consulta: 05/06/2011].  
<http://www.w3.org/TR/rdf-primer/>
  - VV.AA. (10/02/2004). “RDF Vocabulary Description Language 1.0: RDF Schema”. W3C 2004. [Fecha de consulta: 05/06/2011].  
<http://www.w3.org/TR/rdf-schema/>
  - VV.AA. (10/02/2004). “RDF Semantics”. W3C 2004. [Fecha de consulta: 05/06/2011].  
<http://www.w3.org/TR/rdf-mt/>
  - VV.AA. (09/2007) “OWL Web Ontology Lenguaje Primer”. W3C. [Fecha de consulta: 05/06/2011].  
<http://www.w3.org/2007/09/OWL-Overview-es.html>
  - VV.AA. (10/02/2004). “Resource Description FrameWork (RDF): Concepts And Abstract Syntaxis”. W3C 2004. [Fecha de consulta: 05/06/2011].  
<http://www.w3.org/TR/rdf-concepts/>
  - VV.AA. (09/2007). “Lenguaje de Ontologías Web (OWL). Vista General”. W3C 2004. [Fecha de consulta: 05/06/2011].  
<http://www.w3.org/2007/09/OWL-Overview-es.html>
  - VV.AA. (10/02/2004). “OWL Web Ontology Language Guide”. W3C 2004. [Fecha de consulta: 05/06/2011].  
<http://www.w3.org/TR/owl-guide/>
  - VV.AA. (10/02/2004). “OWL Web Ontology Language Overview”. W3C 2004. [Fecha de consulta: 05/06/2011].  
<http://www.w3.org/TR/owl-features/>
  - VV.AA. (2005). “Preguntas frecuentes sobre el lenguaje de Ontologías Web (OWL) del W3C”. W3C 2005. [Fecha de consulta: 05/06/2011].  
<http://www.w3c.es/Traducciones/es/SW/2005/owlfaq>
  - VV.AA. (2008). “SPARQL Query Language for RDF” ”. W3C. [Fecha de consulta: 12/06/2011].  
<http://www.w3.org/TR/rdf-sparql-query/>
  - VV.AA. (2004). “SWRL: A Semantic Web Rule Language Combining OWL and RuleML”. W3C. [Fecha de consulta: 12/06/2011].

## 21. DOCUMENTACIÓN RELACIONADA

Documentación del proyecto relacionada con este documento.

DOCUMENTACIÓN RELACIONADA	
Documento	Descripción
Plan docente	Plan docente de la asignatura. PFC-XML y Web semántica (marzo-julio 2011)
Apuntes MGPI	Apuntes y prácticas de la asignatura Metodología de gestión de proyectos informáticos.
PMBOK	Tercera edición del PMBOK.
PEC	Textos presentados como solución de las PEC.
Vídeo Presentación	Vídeo con la presentación del trabajo disponible en el blog de la asignatura.