



**Universidad de Valladolid**

**E.T.S Ingeniería Informática**

**TRABAJO FIN DE GRADO**

**Grado en Ingeniería Informática**

**Curso en Moodle de  
construcción de  
ontologías con Protégé**

**Autor:**

**D. Carlos A. Travieso Merino**

**Tutor:**

**D. Carlos J. Alonso González**



*Agradecimientos*

*A mis familiares y mis amigos, por su comprensión y apoyo en todo momento.*



## Resumen

La idea de Web Semántica está despertando el interés de numerosas personas que quieren formarse en la construcción de ontologías, que permiten compartir un vocabulario común del dominio así como las relaciones entre sus elementos. Este TFG proporciona una herramienta que facilita dicha formación, a través de un curso virtual en Moodle para la construcción de ontologías utilizando el editor Protégé.

Este documento relata cómo se ha llevado a cabo la construcción del curso, desde la planificación, que es la parte más relacionada con la ingeniería, hasta la más pedagógica donde se exponen los objetivos, los contenidos y la metodología utilizada. También incluye una pequeña guía para los posibles usuarios del curso. Además, recoge la información procesada en el trabajo de documentación realizado con anterioridad a la implementación del curso y una breve introducción a Protégé-OWL. La memoria finaliza con la exposición de las conclusiones y líneas de trabajo futuras junto con las referencias web y bibliográficas utilizadas.

## Abstract

Many people are feeling attracted to the idea of Semantic Web and want to be educated in ontology construction. Ontologies allow sharing the vocabulary of a domain and also the relationships among its elements. This TFG provides a tool that makes easier that education through a virtual course in Moodle for constructing ontologies using the Protégé editor.

This document relates how the course has been made, since the planification, which is the aspect most related to the engineering, to the most pedagogical aspects, including objectives, contents and methodology. Besides, there is a chapter with the result of the work done before implementing the course and another with a brief introduction to Protégé-OWL. The report finishes with the conclusions, the future plans and also the references.



## Tabla de contenido

1. Introducción .....	11
Motivación .....	11
Objetivos.....	12
Metodología.....	12
Resumen del contenido de la memoria .....	13
2. Plan de trabajo.....	15
Visión general del proyecto.....	15
Propósito, alcance y objetivos generales .....	15
Entregas del proyecto .....	16
Organización del proyecto .....	17
Roles y responsabilidades .....	17
Gestión del proyecto .....	18
Estimaciones .....	18
Planificación .....	19
Monitorización del proyecto .....	22
Plan de gestión de riesgos.....	22
Hitos importantes .....	28
Fase de Inicio .....	28
Fase de Análisis .....	28
Fase de Diseño .....	28
Fase de Implementación.....	29
Fase de Pruebas .....	29
Proceso técnico.....	30
Hardware .....	30
Sistemas operativos .....	30
Herramientas .....	31
3. Ontologías .....	33
Introducción .....	33
Historia.....	35
Futuro .....	36
Definición de Ontología.....	37

## Curso en Moodle de construcción de ontologías con Protégé

Clasificación de las ontologías .....	39
Componentes de las ontologías .....	40
Operaciones con ontologías .....	41
Relaciones entre ontologías .....	43
Ontologías y SBCs (Sistemas basados en conocimiento) .....	43
Intercambio de conocimiento .....	44
Representaciones flexibles .....	45
Elección de los individuos y las relaciones .....	45
Representaciones gráficas .....	49
Relaciones primitivas frente a derivadas.....	50
Ontologías y conocimiento compartido .....	54
Lógica de descripción .....	58
Ontologías de alto nivel .....	62
La Web semántica .....	70
Implantación.....	71
Dificultades.....	73
4. Protégé y OWL.....	75
OWL .....	75
Protégé.....	76
Relación Protégé-OWL.....	78
5. Curso de Ontologías en Moodle .....	79
Objetivos generales .....	79
Metodología .....	79
Contenidos .....	81
Bloque 1. Introducción .....	81
Bloque 2. Clases en OWL .....	82
Bloque 3. Propiedades. Propiedades de objeto.....	83
Bloque 4. Propiedades. Definiendo clases (Restricciones existenciales) .....	83
Bloque 5. Condiciones necesarias y suficientes .....	84
Bloque 6. El razonador .....	85
Bloque 7. Propiedades. Definiendo clases (Restricciones universales).....	86
Bloque 8. El razonador (2ª parte) .....	86



Bloque 9. Particiones de valor .....	87
Bloque 10. Propiedades. Definiendo clases (Restricciones de cardinalidad).....	88
Bloque 11. Propiedades. Datatype Properties. ....	89
Bloque 12. Definiendo clases. Aspectos avanzados.....	90
Bloque 13. Otros aspectos.....	90
Guía de usuario básica .....	92
Guía profesor.....	93
Guía alumno.....	96
6. Conclusiones y líneas de trabajo futuras .....	99
7. Bibliografía, referencias y recomendaciones.....	101
Bibliografía y referencias web.....	101
Recomendaciones.....	103
Anexo I. Encuesta usuarios de prueba.....	105
Anexo II. Contenido del CD.....	107



## 1. Introducción

Este capítulo está dedicado a exponer las bases sobre las que se asienta este proyecto, los objetivos que se pretenden alcanzar, así como un resumen del contenido de la memoria.

### Motivación

Este Trabajo Fin de Grado (TFG) surge de las inquietudes que me quedaron sobre el tema de ontologías, que se trata en la asignatura Ingeniería del Conocimiento del Grado de Informática (en este caso, el Curso de Adaptación). Quería saber un poco más sobre cómo se construyen las ontologías pero no tanto la vertiente teórica, vista en la asignatura, sino en su vertiente más práctica.

Una ontología, en pocas palabras, es la especificación de los conceptos del dominio de un área del conocimiento y de las reglas con las que se relacionan. Es un vocabulario común que posteriormente pueden compartir distintas aplicaciones y lograr una correcta interacción entre ellas. Además, dependiendo del lenguaje con el que se definan las ontologías, se puede razonar sobre ellas para inferir conocimiento nuevo.

Tras consultar con Carlos J. Alonso, tutor de este proyecto, descubrí el editor de ontologías Protégé, que facilita mucho la tarea de construir una ontología en OWL (Ontology Web Language). Me pareció un tema muy atractivo para trabajar y, aprovechando que existía un gran interés por parte de otros alumnos en las ontologías y sus posibles aplicaciones, entre ellas la Web 3.0, acordé con el tutor desarrollar un curso multimedia que permitiera a las personas interesadas, aprender a construir ontologías utilizando Protégé.

De esta forma, este trabajo permitiría saciar mi curiosidad en este campo, además de facilitar las cosas al resto de personas interesadas en el tema. El curso podría utilizarse posteriormente en diversos contextos como los cursos de formación que oferta la UVa, un posible Máster de Informática, o simplemente como ampliación de temario para los alumnos del Grado con más interés, entre otros muchos.

## Objetivos

Con la realización de este TFG, se tiene por objetivo crear una herramienta de apoyo para la enseñanza de la construcción de ontologías en OWL utilizando Protégé dando respuesta al creciente interés detectado en este tema. El curso va dirigido especialmente a los alumnos que manejen los contenidos de la asignatura Ingeniería del Conocimiento del Grado de Informática de la Universidad de Valladolid.

Por otro lado, los objetivos didácticos de este curso se resumen en los siguientes resultados de aprendizaje que un alumno debería alcanzar tras completarlo:

- Comprender el concepto de ontología.
- Desarrollar una ontología y definir la jerarquía de clases adecuada.
- Manejar los distintos tipos de propiedades: de objeto, de tipos de datos y de anotación.
- Utilizar las propiedades para crear las restricciones que se utilizan para describir las clases: restricciones existenciales, universales, de cardinalidad y de valor.
- Conocer el mecanismo de razonamiento que se lleva a cabo en OWL y ser capaz de aprovechar todas sus utilidades: clasificación automática de clases e individuos y comprobación de la consistencia.
- Conocer y aprovechar las facilidades que brinda Protégé a la hora de construir ontologías.

Desde un punto de vista más personal, mis objetivos a la hora de realizar este trabajo se resumen en los dos siguientes:

- Conocer con más profundidad el tema de las ontologías, especialmente todo aquello relacionado con ontologías de alto nivel y las ontologías aplicadas a la Web Semántica o Web 3.0.
- Ser capaz de manejar con soltura la herramienta Protégé para la construcción de ontologías en OWL.

## Metodología

Antes de comenzar la creación del curso multimedia, dediqué una gran cantidad de horas a formarme en el tema, recopilando y analizando información proveniente de multitud de fuentes. Además, tuve que emplear un tiempo a conocer el funcionamiento de la herramienta Protégé y de la plataforma Moodle, en la que se decidió implementar el curso.

Posteriormente, dio comienzo el proceso de creación del curso que se dividió en varias fases. Inicialmente se realizó un análisis de lo que se quería conseguir y qué requisitos debía satisfacer el curso. Posteriormente, se diseñó la estructura del mismo y se implementó rellenando de contenidos los distintos bloques planificados. Finalmente, el curso pasó por una fase de pruebas que sirvió para depurar diversos fallos.

## Resumen del contenido de la memoria

A continuación se presenta cómo está estructurada, en lo que sigue, esta memoria de TFG:

- **Capítulo 2: Plan de trabajo.** Recoge la información acerca del plan de trabajo y la gestión del proyecto así como de los resultados alcanzados y artefactos generados en cada una de las fases del proyecto..
- **Capítulo 3: Ontologías.** Expone toda la información procesada en la primera fase del proyecto. Es una completa guía para conocer qué son las ontologías y sus aplicaciones.
- **Capítulo 4: Protégé y OWL.** Breve descripción de la herramienta de edición de ontologías Protégé. Describe la relación entre Protégé y el lenguaje OWL y algunas pautas para la instalación de Protégé.
- **Capítulo 5: Curso de ontologías en Moodle.** Recoge los principios metodológicos y los objetivos perseguidos con la creación del curso en Moodle sobre ontologías. Además, describe brevemente los contenidos tratados y los resultados de aprendizaje que se esperan obtener.
- **Capítulo 6: Conclusiones y líneas de trabajo futuras.** Recoge las reflexiones personales sobre lo que he aprendido con la realización de este TFG así como las posibles líneas de trabajo futuras.
- **Capítulo 7: Bibliografía y referencias.** Recoge la bibliografía y los sitios web consultados para la elaboración de este TFG.
- **Anexo I. Encuesta usuarios de prueba:** Encuesta que tendrán que rellenar los usuarios de prueba en la fase de transición del curso.
- **Anexo II. Contenido del CD:** Se presenta el contenido del CD entregado junto con esta memoria.



## 2. Plan de trabajo

Este capítulo recopila toda la información relacionada con las fases y actividades planificadas para realizar este proyecto, así como para controlar su evolución. Este plan de trabajo será utilizado por las siguientes personas:

- **Tutor del proyecto.** Se encargará de validar las entregas puntuales. Se encargará de esta labor D. Carlos J. Alonso González.
- **Autor del proyecto.** Se encargará de seguir el plan de trabajo donde se especifica qué y cuándo debe realizar las diferentes actividades programadas.

En las siguientes secciones vendrán detallados los siguientes puntos:

- **Visión general del proyecto.** Describe el propósito, alcance y objetivos del proyecto, así como las entregas programadas.
- **Organización del proyecto.** Describe la organización del equipo de trabajo.
- **Gestión del proyecto.** Describe las estimaciones, las fases y las técnicas utilizadas para monitorizar el proyecto.
- **Hitos importantes.** Resume las conclusiones y el trabajo realizado en cada uno de los hitos clave para el desarrollo del proyecto.
- **Proceso Técnico.** Describe el hardware y las herramientas software empleadas a lo largo del proyecto.

### Visión general del proyecto

En esta sección se describe el propósito, alcance y objetivos del proyecto, así como las entregas programadas.

#### *Propósito, alcance y objetivos generales*

Con la realización de este TFG, se tiene por objetivo crear una herramienta de apoyo para la enseñanza de la construcción de ontologías en OWL utilizando Protégé dando respuesta al creciente interés detectado en este tema.

Dicha herramienta será un curso implementado en un entorno virtual de aprendizaje que permita a los usuarios seguirlo de forma presencial, semi-presencial o no presencial. El curso deberá tener una duración aproximada de 40 horas y podrá utilizarse posteriormente en diversos contextos como los cursos de formación que oferta la UVa, un posible Máster de Informática, o simplemente

## Curso en Moodle de construcción de ontologías con Protégé

como ampliación de temario para los alumnos del Grado con más interés, entre otros muchos.

El hilo conductor del curso será la implementación de la ontología “Pizza Ontology” diseñada por investigadores de la Universidad de Manchester para facilitar el aprendizaje de Protégé en su faceta de asistente a la creación de ontologías en OWL (Ontology Web Language). Sin embargo, se diseñarán muchas más actividades para que los usuarios puedan afianzar los conceptos.

### ***Entregas del proyecto***

Se realizarán las siguientes entregas a lo largo de este proyecto:

<b>Entrega</b>	<b>Fecha</b>	<b>Descripción</b>	<b>Formato de entrega</b>
Anteproyecto y resultado del trabajo de formación.	20/01/2012	Anteproyecto oficial y documento con la información obtenida y procesada de múltiples fuentes para tener una idea general del tema.	Correo electrónico y reunión con tutor.
Contenidos y EVA (Entorno virtual de aprendizaje)	24/02/2012	Documento informal con los contenidos que debía tratar el curso y la plataforma sobre la que se iba a implementar.	Reunión con tutor.
Instalación y puesta en marcha de la plataforma.	02/03/2012	Puesta en marcha de la página web donde se encuentra alojado el curso.	Página web.
Estructura del curso.	07/03/2012	Documento informal con los bloques de contenidos del curso y su estructura.	Reunión con tutor.
Curso de construcción de ontologías en Protégé.	15/06/2012	Curso virtual implementado listo para ser revisado.	Página web.
Borrador documentación	01/07/2012	Borrador de la documentación para ser revisado por el tutor.	Archivo pdf.



Entrega final (curso revisado + documentación).	12/07/2012	Curso virtual revisado y documentación requerida.	Memoria impresa, CD-ROM y página web.
---	------------	---	---

Tabla 1. Entregas del Proyecto

## Organización del proyecto

En esta sección se describe la organización del equipo de trabajo. Se debe destacar que para la realización de este proyecto se contó con un capital humano de dos personas: Carlos J. Alonso González y Carlos A. Travieso Merino.

### ***Roles y responsabilidades***

En este apartado se muestra una tabla con los roles existentes en el equipo de trabajo, sus responsabilidades y la persona/s asignadas para desempeñar cada rol.

Rol	Funciones	Responsable/s
<b>Jefe de proyecto</b>	<ul style="list-style-type: none"> <li>• Dirigir y organizar el trabajo.</li> <li>• Determinar los objetivos.</li> <li>• Toma de decisiones principales.</li> <li>• Elaborar, revisar y actualizar la planificación.</li> <li>• Elaborar, revisar y actualizar el plan de gestión de riesgos.</li> <li>• Gestionar los recursos.</li> <li>• Realizar el seguimiento del proyecto.</li> </ul>	Carlos A. Travieso Merino Carlos J. Alonso González
<b>Analista</b>	<ul style="list-style-type: none"> <li>• Recopilar y procesar la información necesaria para establecer el dominio del problema.</li> <li>• Especificar y analizar el dominio del problema.</li> </ul>	Carlos A. Travieso Merino
<b>Diseñador</b>	<ul style="list-style-type: none"> <li>• Diseñar la estructura y el formato del curso para dar</li> </ul>	Carlos A. Travieso Merino

	solución a los problemas planteados. <ul style="list-style-type: none"> <li>• Diseñar las ontologías de ejemplo que serán utilizadas en las actividades del curso.</li> </ul>	
<b>Desarrollador</b>	<ul style="list-style-type: none"> <li>• Estructurar y dar formato al curso.</li> <li>• Implementar con el editor Protégé las ontologías de ejemplo utilizadas en el curso.</li> </ul>	Carlos A. Travieso Merino
<b>Probador</b>	<ul style="list-style-type: none"> <li>• Probar el funcionamiento de las actividades del curso.</li> <li>• Probar el funcionamiento de las ontologías creadas para el curso.</li> </ul>	Carlos A. Travieso Merino Carlos J. Alonso González

**Tabla 2. Roles, funciones y responsables del proyecto**

La elaboración de la documentación pertinente la realizará Carlos A. Travieso Merino ejercitando cada uno de los roles que tiene asignados.

## Gestión del proyecto

Este apartado describe las estimaciones, las fases y las técnicas utilizadas para monitorizar el proyecto.

### *Estimaciones*

Las estimaciones de tiempos y recursos para la realización de las diferentes actividades de este proyecto vienen reflejadas en la planificación. Dichas estimaciones se han llevado a cabo a partir de la experiencia acumulada durante el Grado, de experiencias de proyectos anteriores, así como durante el desempeño de la actividad profesional. Además, la experiencia del tutor del proyecto en este tipo de trabajos influyó de forma muy importante en la toma de decisiones con respecto a la temporalización del proyecto.

Se debe tener en cuenta también que las jornadas de trabajo dedicadas al TFG (días en el Diagrama de Gantt) no son de ocho horas ya que a lo largo de este año el autor ha desarrollado otras actividades. Tras un análisis detallado, se

puede realizar una estimación aproximada de 3 horas por cada día completo de trabajo planificado.

### ***Planificación***

La elaboración de este TFG se ha distribuido en cinco fases, habiéndose establecido una fase a mayores pensando en la continuidad en el tiempo de este proyecto y en su aplicación práctica real:

- 1) **Inicio.** En esta fase se definen el ámbito y el alcance del proyecto. Se realiza la planificación temporal y se realiza la mayor parte de las actividades de recopilación y procesado de información sobre el dominio del proyecto. También se llevan a cabo todas las actividades de formación con respecto al manejo de la herramienta Protégé.
- 2) **Análisis.** En esta fase se analiza la información procesada para establecer una serie de recomendaciones acerca de los contenidos a tratar en el curso y la forma de hacerlo. Se toman decisiones importantes como el entorno virtual de aprendizaje elegido para implementar el curso.
- 3) **Diseño.** En esta fase se diseña la estructura del curso, el reparto de los contenidos en bloques y la estructura de cada bloque. También se diseñan las ontologías de ejemplo utilizadas en las actividades del curso. Se realiza la instalación del entorno virtual de aprendizaje elegido y se llevan a cabo las actividades de formación de la herramienta.
- 4) **Implementación.** En esta fase se desarrolla todo el material teórico diseñado para el curso y todas las actividades con sus ontologías asociadas.
- 5) **Pruebas.** En esta fase se prueba el curso en busca de errores de cualquier tipo, tanto en las actividades implementadas en la plataforma virtual como las que se proponen para realizar en Protégé. Se probará, además, el correcto funcionamiento de todas las ontologías propuestas. Se diseña la encuesta que se pasará a los usuarios beta del curso.
- 6) **Transición.** En esta fase se lleva el curso a un contexto semi-real y se prueba con usuarios beta hasta obtener la versión definitiva del curso aplicable en un contexto real. La idea de esta fase es testar el curso con usuarios beta de tal forma que rellenen a la vez un informe con el que poder realizar un análisis cuantitativo y cualitativo de la calidad del curso y de los aspectos a corregir. Una vez realizados los cambios pertinentes en función de dicho análisis, la aplicación ya estaría preparada para ser utilizada en un contexto real.

## Curso en Moodle de construcción de ontologías con Protégé

Este TFG solamente alcanzará la fase cinco de las seis propuestas. Lamentablemente, la sexta fase no ha podido ser llevada a cabo y se propone como línea de trabajo futura.

Hay que comentar que no se sigue un modelo tradicional de cascada en la realización de este TFG, sino que las fases van entrelazadas entre sí y los documentos generados en cada una de ellas son actualizados a medida que se avanza en el proyecto. Esta superposición de las fases se puede comprobar en el diagrama de Gantt que se muestra a continuación donde se ve el orden de las actividades realizadas a lo largo del proyecto.

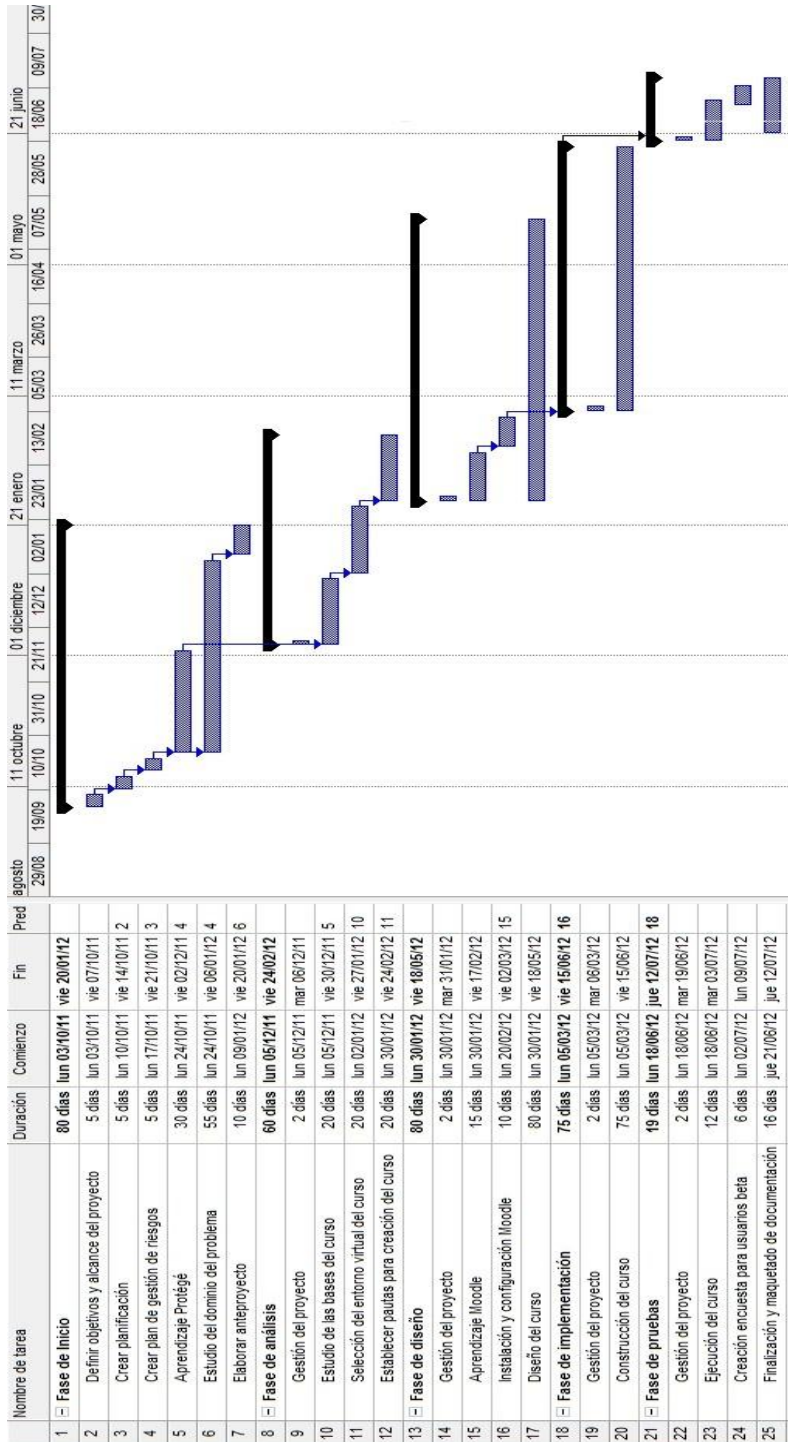


Imagen 1. Diagrama de Gantt del proyecto

### ***Monitorización del proyecto***

Para controlar el desarrollo del proyecto, se comprobará semanalmente el ajuste a la planificación y se contactará con el tutor del proyecto a través del correo electrónico para informarle de los avances, resolver las dudas y comunicarle información importante. Además, se establecerán reuniones periódicas (al menos una al finalizar cada fase del proyecto) para revisar el material generado y examinar el cumplimiento de los plazos.

### ***Plan de gestión de riesgos***

Tras un estudio detallado se identificaron una serie de riesgos que pueden aparecer en cualquiera de las fases del proyecto. A medida que el proyecto se desarrollaba se identificaron nuevos riesgos en las revisiones periódicas que se hacían a este plan. En este apartado se refleja el contenido final del documento del plan de riesgos.

Para documentar cada riesgo se elaboró una tabla en la que se identificaba cada uno de ellos y sus características más importantes. La tabla contiene los siguientes campos:

- **Identificador (ID).** Identifica de forma única a cada riesgo.
- **Descripción.** Explica el riesgo.
- **Probabilidad.** Estimación en porcentaje de la probabilidad de aparición del riesgo.
- **Impacto.** Clasificación de la amenaza para el desarrollo del proyecto en caso de que el riesgo se manifieste. Se clasifican según las siguientes categorías:
  - *Catastrófico:* El proyecto fracasa con alta probabilidad.
  - *Crítico:* El éxito del proyecto queda muy comprometido.
  - *Grave:* El desarrollo del proyecto se ve considerablemente afectado.
  - *Bajo:* La calidad del proyecto se ve reducida.
  - *Marginal:* El desarrollo del proyecto prácticamente no se ve afectado.
- **Indicadores.** Hechos a observar para intuir la aparición del riesgo.

Las siguientes dos tablas muestran los riesgos detectados:

ID	Descripción	Probabilidad
R1	Planificación inadecuada	40-60%
R2	Falta de tiempo por incompatibilidades con otras actividades	70-80%
R3	Cambios en los objetivos	5-10%
R4	Falta de experiencia en el tema del proyecto	40-60%
R5	Dificultades con el uso de Protégé	20-30%
R6	Dificultades con el uso de Moodle	50-60%
R7	Baja temporal	5-10%
R8	Imposibilidad de presencia física en momentos importantes	90-100%
R9	Procrastinación	5-10%

Tabla 3. Riesgos y probabilidades

ID	Impacto	Indicadores
R1	Crítico	Entrega de artefactos fuera de plazo
R2	Grave	Retrasos detectados en las revisiones semanales
R3	Crítico	Se informa de que cambia alguno de los objetivos del proyecto
R4	Grave	Se paraliza temporalmente el proyecto por la falta de conocimiento
R5	Grave	Se paraliza temporalmente el proyecto por no saber implementar correctamente las ontologías con Protégé
R6	Grave	Se paraliza temporalmente el proyecto por no saber cómo implementar en Moodle las actividades diseñadas
R7	Grave	Incapacidad para trabajar por motivos de salud o personales de algún miembro del equipo
R8	Bajo	Necesidad de algún miembro del equipo de desplazarse fuera de la ciudad durante algún

		tiempo
R9	Bajo	Acumulación de trabajo en las fechas cercanas a las entregas internas establecidas

Tabla 4. Riesgos e indicadores

Para hacer frente a los riesgos, se siguió una estrategia en la que se abordan tres aspectos clave:

- Reducción del riesgo
- Supervisión del riesgo
- Gestión del riesgo (plan de contingencia por si aparece el riesgo)

En las tablas que se presentan a continuación se detalla la estrategia a seguir para la prevención y supervisión de cada uno de los riesgos y para la gestión de los mismos en caso de que se manifiesten para ayudar al equipo del proyecto a controlar el correcto desarrollo del mismo.

<b>R1. Planificación inadecuada</b>
<b>Plan de reducción:</b>
<ul style="list-style-type: none"> <li>• Ser realistas en la elaboración de la planificación.</li> </ul>
<b>Plan de supervisión:</b>
<ul style="list-style-type: none"> <li>• Comprobaciones semanales y reuniones periódicas con el tutor del proyecto.</li> </ul>
<b>Plan de contingencia:</b>
<ul style="list-style-type: none"> <li>• Adaptar la planificación para adaptarla a las nuevas circunstancias.</li> </ul>

Tabla 5. Estrategia R1

<b>R2. Falta de tiempo por incompatibilidades con otras actividades</b>
<b>Plan de reducción:</b>
<ul style="list-style-type: none"> <li>• Ser realistas a la hora de asignar trabajo en las fechas en las que se desarrollan las otras actividades.</li> </ul>
<b>Plan de supervisión:</b>
<ul style="list-style-type: none"> <li>• Llevar una agenda paralela del trabajo que implican las otras actividades y comprobar semanalmente si la planificación se puede seguir.</li> </ul>
<b>Plan de contingencia:</b>
<ul style="list-style-type: none"> <li>• Adaptar la planificación para adaptarla a las nuevas circunstancias.</li> </ul>

Tabla 6. Estrategia R2



<b>R3. Cambios en los objetivos</b>
<b>Plan de reducción:</b>
<ul style="list-style-type: none"> <li>• Fijar los objetivos del TFG de forma clara y concisa en la fase de inicio.</li> </ul>
<b>Plan de supervisión:</b>
<ul style="list-style-type: none"> <li>• Reuniones periódicas con el tutor del proyecto.</li> </ul>
<b>Plan de contingencia:</b>
<ul style="list-style-type: none"> <li>• Aumentar la carga de trabajo para repasar todo el material elaborado y comprobar que se ajusta a los nuevos objetivos definidos.</li> </ul>

Tabla 7. Estrategia R3

<b>R4. Falta de experiencia en el tema del proyecto</b>
<b>Plan de reducción:</b>
<ul style="list-style-type: none"> <li>• Asignar un tiempo suficiente en la fase de inicio a la búsqueda, procesado y estudio de la teoría necesaria para la realización de este TFG.</li> </ul>
<b>Plan de supervisión:</b>
<ul style="list-style-type: none"> <li>• Reuniones periódicas con el tutor del proyecto.</li> </ul>
<b>Plan de contingencia:</b>
<ul style="list-style-type: none"> <li>• Aumentar la carga de trabajo para rellenar el vacío de conocimiento que pueda existir en un determinado momento.</li> </ul>

Tabla 8. Estrategia R4

<b>R5. Dificultades con el uso de Protégé</b>
<b>Plan de reducción:</b>
<ul style="list-style-type: none"> <li>• Asignar un tiempo suficiente en la fase de inicio al aprendizaje del uso de esta herramienta.</li> <li>• Leer y seguir manuales de Protégé en la fase de inicio del proyecto hasta controlar correctamente la herramienta.</li> <li>• Consultas con expertos en el manejo de la herramienta Protégé.</li> </ul>
<b>Plan de supervisión:</b>
<ul style="list-style-type: none"> <li>• Reflexión al término de la fase de inicio.</li> </ul>
<b>Plan de contingencia:</b>
<ul style="list-style-type: none"> <li>• Aumentar la carga de trabajo para completar más manuales de uso de Protégé o practicar más con la herramienta.</li> </ul>

Tabla 9. Estrategia R5

<b>R6. Dificultades con el uso de Moodle</b>
<b>Plan de reducción:</b>
<ul style="list-style-type: none"> <li>• Asignar un tiempo suficiente en la fase de diseño al aprendizaje del uso de esta herramienta.</li> <li>• Leer y seguir manuales de Moodle en la fase de diseño del proyecto hasta manejar correctamente la herramienta.</li> <li>• Aprovechar las clases del Máster al que asisto durante este curso académico donde se explica el manejo básico de la herramienta Moodle.</li> </ul>
<b>Plan de supervisión:</b>
<ul style="list-style-type: none"> <li>• Reflexión al término de la fase de diseño.</li> </ul>
<b>Plan de contingencia:</b>
<ul style="list-style-type: none"> <li>• Aumentar la carga de trabajo para practicar más con la herramienta.</li> </ul>

Tabla 10. Estrategia R6

<b>R7. Baja temporal</b>
<b>Plan de reducción:</b>
<ul style="list-style-type: none"> <li>• Elaborar un calendario de disponibilidad personal para trabajar.</li> </ul>
<b>Plan de supervisión:</b>
<ul style="list-style-type: none"> <li>• Comprobación periódica del calendario.</li> </ul>
<b>Plan de contingencia:</b>
<ul style="list-style-type: none"> <li>• Adaptación de la planificación a las nuevas circunstancias.</li> <li>• Aumentar la carga de trabajo para compensar las horas de trabajo perdidas en el periodo de baja.</li> </ul>

Tabla 11. Estrategia R7

<b>R8. Imposibilidad de presencia física en momentos importantes</b>
<b>Plan de reducción:</b>
<ul style="list-style-type: none"> <li>• Elaborar un calendario de presencia física en Valladolid y planificar las reuniones presenciales conforme a dicho calendario.</li> </ul>
<b>Plan de supervisión:</b>
<ul style="list-style-type: none"> <li>• Comprobación periódica del calendario.</li> </ul>
<b>Plan de contingencia:</b>
<ul style="list-style-type: none"> <li>• Adaptación de la planificación a las nuevas circunstancias.</li> <li>• Tratar los asuntos importantes a través del correo electrónico si fuera posible.</li> <li>• Celebrar reuniones por videoconferencia si llegara a ser necesario.</li> </ul>

Tabla 12. Estrategia R8

<b>R9. Procrastinación</b>
<b>Plan de reducción:</b>
<ul style="list-style-type: none"> <li>• Elaborar una agenda personal de trabajo semanal conforme a la planificación del proyecto.</li> </ul>
<b>Plan de supervisión:</b>
<ul style="list-style-type: none"> <li>• Comprobación semanal del cumplimiento de la planificación.</li> </ul>
<b>Plan de contingencia:</b>
<ul style="list-style-type: none"> <li>• Aumentar la carga de trabajo de alguna semana si se detecta que ha habido procrastinación en las anteriores.</li> </ul>

Tabla 13. Estrategia R9

## **Hitos importantes**

En esta sección se resumen las conclusiones, resultados obtenidos, artefactos generados y el trabajo realizado en cada una de las fases del proyecto.

### ***Fase de Inicio***

En esta fase se celebraron los primeros contactos entre el tutor y el autor del proyecto para definir los objetivos del mismo y determinar el alcance. Estos primeros contactos se realizaron prácticamente al comenzar el curso lectivo y la fase de inicio se extendió hasta el mes de Enero ya que se realizó la mayor parte del trabajo de búsqueda y procesamiento de información cuyo resultado se expone en el capítulo 3 de este documento. En este tiempo, también se aprendió a utilizar Protégé. A continuación está la lista de resultados obtenidos y artefactos creados al finalizar esta fase:

- Conocimiento del dominio del problema (ontologías).
- Capítulo 3 de este documento al 90%.
- Dominio de la herramienta Protégé.
- Anteproyecto.

### ***Fase de Análisis***

En esta fase se decidió definitivamente utilizar el manual desarrollado por la Universidad de Manchester para aprender Protégé a través de la ontología de la Pizza como referencia en el que basar el curso en Moodle. Además se eligió Moodle como plataforma donde desarrollar el curso virtual y se establecieron algunas pautas para su creación: debía contener aspectos teóricos, actividades de práctica, entregas periódicas y material complementario sobre los temas tratados.

### ***Fase de Diseño***

A lo largo de esta fase se estableció la distribución de los contenidos a tratar en bloques y se determinó que cada uno de ellos debía contener: material de trabajo, material complementario, una lección (practicar conceptos) y una entrega basada en la ontología de la Pizza, la cual irían desarrollando a lo largo del curso para asentar los conceptos aprendidos. Este trabajo tenía asociado el desarrollo

de las ontologías con las que realizar prácticas en las diferentes lecciones del curso.

En esta fase, también se llevó a cabo la instalación, configuración y gestión de la plataforma Moodle. Se mantuvieron contactos con los Técnicos de la Escuela para la creación de una base de datos que pudiera utilizar la página web donde está alojada la plataforma. Se tuvo que realizar todo el trabajo de instalación de Moodle y de creación, configuración y gestión del curso y de los usuarios.

Al término de esta fase ya se disponía de:

- Conocimientos sobre la utilización y gestión de la plataforma Moodle.
- Plataforma Moodle instalada, correctamente configurada y funcionando.
- Estructura del curso y actividades planificadas.

### ***Fase de Implementación***

En esta fase se trasladaron al curso en Moodle todas las ideas, materiales y actividades que se habían planificado en la fase de diseño. Al término de esta fase, ya se disponía del curso completo listo para ser probado.

### ***Fase de Pruebas***

En esta fase, los componentes del equipo de pruebas completaron el curso completo en busca de errores tanto en los materiales, las actividades propuestas como en el funcionamiento de las ontologías de ejemplo utilizadas. Además, en esta fase se elaboró la encuesta que deberían completar los usuarios beta del curso en la fase de transición. Al término de esta fase, se disponía de:

- Curso completo y probado.
- Documentación requerida para el TFG finalizada.
- Fase de transición preparada.

## **Proceso técnico**

En esta sección se describe el hardware y las herramientas software empleadas a lo largo del proyecto.

### ***Hardware***

Con respecto al hardware utilizado para el desarrollo de este proyecto, se debe comentar que se ha utilizado el ordenador personal del autor, si bien se podría haber utilizado cualquier otro ordenador ya que ese detalle no tiene gran importancia debido a la naturaleza del proyecto que requería de:

- Elaboración de documentos y búsqueda de información en Internet.
- Trabajo con la herramienta Protégé.
- Creación del curso en Moodle (se trabajo on-line en la página web creada).

Por tanto, cualquier computadora que permitiera realizar las tres tareas anteriores era válida para trabajar en el proyecto.

También se ha utilizado un servidor externo donde colocar la página web con el curso de Moodle. Para ello, se ha utilizado Jair, el servidor de la Escuela.

### ***Sistemas operativos***

Se ha trabajado con el sistema operativo Windows Vista para todo lo relacionado con las tres actividades mencionadas en el apartado anterior, trabajos con ofimática y consultas de Internet, trabajo con Protégé y creación del curso en Moodle. Si bien, cualquier sistema operativo hubiera sido viable para la realización de este TFG, habiéndose elegido Windows por comodidad y preferencias personales.

Sin embargo, la parte de instalación y configuración de la página web donde se encuentra alojado el curso en Moodle se tuvo que realizar en Linux, ya que el servidor Jair así lo requiere. Se tuvo que poner en marcha, por tanto, la versión para Linux proporcionada por el distribuidor de Moodle.

## **Herramientas**

Se han utilizado dos herramientas de distintos tipos a lo largo del proyecto según lo que se quería hacer en cada momento. De esta forma, podemos distinguir entre tres tipos de herramientas, las utilizadas para elaborar la documentación, el material y para llevar a cabo el seguimiento del proyecto, las herramientas utilizadas para desarrollar las ontologías y las herramientas utilizadas para crear el curso virtual.

- 1) Ofimática: desarrollo de material y documentación y seguimiento del proyecto.
  - Microsoft Word 2007
  - Microsoft Project 2003
- 2) Desarrollo de ontologías y plugins de utilidad.
  - Protégé 4.2
  - OWL Viz – Graph Viz 2.28
- 3) Creación del curso en Moodle.
  - Moodle 2.2.3+

También se ha necesitado el uso de otras herramientas para hacer funcionar Moodle en el servidor Jair, si bien no ha sido necesario manejarlas directamente ya que esa es la tarea de los Técnicos de la Escuela.





### **3. Ontologías**

La creciente automatización de los procesos en general ha provocado una proliferación de sistemas de información con semánticas que se solapan. Por otro lado existe la necesidad de reutilizar el conocimiento, para hacer más eficiente esta automatización. Los desarrollos de sistemas basados en conocimiento y de software en general, se construyen en diferentes contextos y siguiendo distintos puntos de vista y suposiciones acerca de su materia de estudio. Cada uno usa su propio vocabulario, pueden tener distintos términos para designar un mismo concepto y diferentes métodos y estructuras. Esto ha provocado problemas de comunicación por falta de entendimiento compartido, limitando la interoperabilidad y el potencial de reutilización y de compartir la información [1].

En ciencias de la computación y en informática, una ontología formal representa el conocimiento como un conjunto de conceptos dentro de un dominio y las relaciones entre estos conceptos. Puede ser utilizado para razonar sobre las entidades de ese dominio y también puede ser utilizado para describir el dominio.

Una ontología especifica una conceptualización, una forma de ver al mundo. Una ontología contiene las definiciones que nos proveen del vocabulario para referirse a un dominio. Sin embargo, la conceptualización es independiente del lenguaje mientras que la ontología depende del lenguaje en el que está construida [2].

Las ontologías son los marcos estructurales para organizar la información y se utilizan en la inteligencia artificial, la Web Semántica, ingeniería de sistemas, ingeniería de software, informática biomédica, marcadores de empresas y la arquitectura de la información como una forma de representación del conocimiento sobre el mundo o parte del mismo.

### **Introducción**

Esta sección es el resultado de la combinación de aportaciones propias y una adaptación de múltiples párrafos del libro de Poole y Mackworth [3].

Por todas partes existen agentes y, en general cualquier sistema que necesite de datos y hechos para poder funcionar, que trabajan con conocimiento. Dicho conocimiento debe ser representado a través de símbolos y a dichos símbolos se les debe dar un significado de tal manera que tanto los desarrolladores como los sistemas los entiendan y los utilicen correctamente.

Una ontología es una especificación del significado de los símbolos usados en un sistema de información.

*En el caso más simple, si un agente utiliza una representación basada en estados, la ontología especifica el mapeo entre el conjunto de estados y el mundo exterior. Sin ese mapeo, el agente puede saber que se encuentra en el estado 57 pero, sin ontología, esa información no sería más que un número sin significado para otro agente u otra persona. En otros casos, la ontología define las características de los individuos y sus relaciones. Es lo que se necesita para convertir datos sin significado en algo con significado para el agente o para que una persona u otra fuente de conocimiento adquiera información llena de significado de esos datos [3].*

Las ontologías suelen ser construidas por comunidades de forma independiente a una base de conocimiento particular o una aplicación específica. Es conocimiento compartido que permite una comunicación efectiva y la interoperabilidad de los datos desde múltiples fuentes (sensores, humanos y bases de datos). Lógicamente, se debe construir la ontología previamente a la definición de los datos y del conocimiento previo de un sistema. Sin una ontología, los humanos no sabríann qué introducir al sistema; es la ontología la que da significado a los datos. Frecuentemente las ontologías evolucionan a medida que el sistema es desarrollado.

*Se puede ver a modo de ejemplo la comunicación entre humanos. Todo el mundo sabe pronunciar palabras, que no son más que sonidos emitidos uno a continuación del otro. Sin unas reglas que rijan el significado de las palabras, el orden en el que tienen que ser pronunciadas, etc. sería imposible una comunicación efectiva entre personas ya que se pronunciarían palabras aleatoriamente, sin un significado entendible por el resto. Esto se puede ver como un símil con lo que ocurre en los sistemas de información. Ellos pueden enviar, recibir y almacenar gran cantidad de datos, pero sin un significado, no sería más que unos y ceros apilados sin ningún sentido. Ahí es donde entran las ontologías, que se encargan de dar significado a los datos y de definir las reglas con las que se comunican los sistemas de información entre sí y con los humanos [Travieso, 2012].*

La ontología especifica un nivel o varios de abstracción y si la ontología cambia, los datos deben cambiar.

*Por ejemplo, si un robot utiliza una ontología que define obstáculos físicos para ser esquivados, si la ontología es ampliada para diferenciar entre mesas, sillas, personas, etc. se requerirán datos diferentes del mundo exterior [3].*

No cabe ninguna duda, que los sistemas de información que obtienen su conocimiento a través de fuentes externas, serían mucho más potentes si tanto el sistema como todas sus fuentes compartieran la misma ontología.

*Por ejemplo, fijémonos en un sistema que facilita a los usuarios las horas de despegue y aterrizaje de los aviones en un determinado aeropuerto. Para ello, debe acceder a las diferentes webs de las diferentes compañías y consultar los*

*datos requeridos. Cada una de esas compañías tiene un sistema de acceso diferente con interfaces distintas. Si todas ellas compartieran la misma ontología podrían usarse de manera conjunta ya que los mismos símbolos tendrían el mismo significado. Cuando los mismos símbolos tienen el mismo significado se dice que existe interoperabilidad semántica. Ésta es la semilla de la web semántica que se explicará posteriormente. Cuando no se comparte ontología, debe existir un mapeo entre las mismas para permitir la interoperabilidad [3].*

## Historia

El término ontología tiene su origen en la filosofía y a lo largo del tiempo ha sido aplicado de múltiples maneras. Históricamente, las ontologías se derivan de la rama de la filosofía conocida como la metafísica, que trata sobre la naturaleza de la realidad, de lo que existe. Esta rama fundamental se ocupa de analizar los diferentes tipos o modos de existencia, a menudo con especial atención a las relaciones entre particulares y universales, entre las propiedades intrínsecas y extrínsecas, y entre esencia y existencia. El elemento *onto* proviene del griego y significa “ser, lo que es”. El objetivo tradicional de la investigación ontológica, en particular, consiste en dividir el mundo "en sus articulaciones" para descubrir las categorías fundamentales o clases a las que los objetos del mundo pertenecen de forma natural [4].

Durante la segunda mitad del siglo XX, los filósofos debatieron intensamente sobre los posibles métodos o enfoques para la construcción de ontologías sin llegar a la construcción de ninguna ontología lo suficientemente elaborada. Por el contrario, en informática se han construido ontologías grandes y robustas, tales como WordNet y Cyc, con relativamente poco debate sobre la forma en que fueron construidas.

Desde mediados de 1970, los investigadores en el campo de la inteligencia artificial (IA) han reconocido que la captura de conocimiento es la clave para la construcción de sistemas de inteligencia artificial grandes y poderosos. Los investigadores de IA argumentaban que se podrían crear nuevas ontologías como modelos computacionales que permitieran ciertos tipos de razonamiento automatizado. En la década de 1980, la comunidad de inteligencia artificial comenzó a utilizar el término ontología para referirse tanto a la teoría de un mundo modelado como a un componente de los sistemas de conocimiento. Algunos investigadores, inspirándose en las ontologías filosóficas, vieron la ontología computacional como una especie de filosofía aplicada [5].

En 1995, el libro "Toward Principles for the Design of Ontologies Used for Knowledge Sharing" de Tomhas Gruber [6] contribuyeron al asentamiento de la ontología como un término técnico en informática.

Gruber introdujo en 2001 el término ontología en el sentido de especificación de una conceptualización, es decir, "Una ontología es una descripción (como una especificación formal de un programa) de los conceptos y relaciones que pueden existir formalmente en un agente o un conjunto de agentes. Esta definición es consistente con el uso de la ontología como un conjunto de definiciones de conceptos, pero más general. Y es un sentido diferente de la palabra de su uso en filosofía". Según Gruber [7], "Las ontologías son a menudo sinónimo de jerarquías taxonómicas de clases, definiciones de clase y relaciones de subsunción, pero las ontologías no tienen por qué limitarse a estas formas. Las ontologías tampoco están limitadas a definiciones conservadoras, es decir, definiciones en el sentido de la lógica tradicional, que sólo introducen la terminología y no aportan ningún conocimiento sobre el mundo. Para especificar una conceptualización, se necesita plantear axiomas que limiten las posibles interpretaciones de los términos definidos" [8].

Lo que muchas ontologías tienen en común tanto en informática como en filosofía es la representación de entidades, ideas y eventos mediante propiedades y relaciones de acuerdo a un sistema de categorías. En ambos campos, se plantean problemas sobre la relatividad ontológica y debates acerca de si una ontología normativa es viable. Las diferencias entre las dos es principalmente una cuestión de enfoque. Los filósofos están menos interesados en establecer vocabularios fijos y controlados que los investigadores en ciencias de la computación, mientras que los informáticos están menos involucrados en las discusiones de los primeros principios, como el debate de si existen cosas tales como esencias fijas o si las entidades deben ser ontológicamente más primarias que los procesos [9].

## Futuro

En los últimos tiempos, se aprecia una progresiva sustitución de los modelos convencionales del proceso de recuperación de la información por modelos cognitivos que aplican el modelo humano de memoria y aprendizaje a los agentes inteligentes de recuperación de la información, proponiendo las ontologías para estructurar las bases del conocimiento. Por ello, los Servicios de ontologías ofrecerán un marco contextualmente rico y moderno para elaborar modelos, prestar servicios y gestionar la terminología en cualquier materia.

Cuando se integre con instrumentos de búsqueda basados en Internet, facilitará mucho la recuperación de recursos, no sólo proporcionando acceso a los documentos específicos que una determinada persona está buscando, sino también ofreciendo sugerencias relativas a otros recursos conexos que sean potencialmente pertinentes.

Esta funcionalidad adicional no sólo aumentará espectacularmente el alcance de los motores de búsqueda basados en Internet, sino también revolucionará la forma en que los usuarios interactuarán con la Web en lo que es conocido como Web Semántica o Web 3.0 [10].

## Definición de Ontología

Esta sección es el resultado de la combinación de aportaciones propias y una adaptación de múltiples párrafos del libro de Poole y Mackworth [3].

Antes de dar una definición de ontología en el campo de la informática, conviene hablar de los orígenes, es decir, de las definiciones aristotélicas.

La categorización de objetos, la base de las ontologías modernas, tiene una larga historia. Aristóteles (350 A.C.) sugirió la definición de una clase C en términos de:

- Género: una superclase de C. El plural de género es géneros.
- Diferenciación: las propiedades que hacen que los miembros de la clase C sean diferentes de otros miembros de la superclase de C.

Él anticipó muchos de los problemas que surgen en las definiciones:

*Si los géneros son diferentes y “coordinados”, su diferenciación está en sus propias diferencias de clase. Tomemos como ejemplo el género “animal” y el género “conocimiento”. “Con pies”, “Con 2 pies”, “Con alas”, “Acuático”, son diferenciaciones entre “animales”; las especies de conocimiento no están distinguidas con la misma diferenciación. Una especie de conocimiento no se diferencia de otra en “tener 2 pies” [Aristóteles (350 A.C.)].*

Se debe puntualizar que “coordinados” significa que ninguno está subordinado a otro. Siguiendo el estilo de las ontologías modernas, se podría decir que “animal” es una clase y “conocimiento” es otra clase. La propiedad “Con 2 pies” tiene como dominio la clase “animal”. Si algo es una instancia de conocimiento, no tiene que tener valor para la propiedad “Con 2 pies”.

Para construir una ontología basada en definiciones Aristotélicas:

- Para cada clase que se quiera definir hay que determinar una superclase relevante y después, seleccionar aquellos atributos que distinguen la clase de otras subclases. Cada atributo proporciona una propiedad y un valor.
- Para cada propiedad hay que definir la clase más general para la cual tenga sentido y definir el dominio de la propiedad para que esté en dicha clase. Se debe definir el rango de clases que tienen esta propiedad.

Esto puede ser bastante complicado. Por ejemplo, al definir “muebles de lujo”, quizás la superclase que se quiere es “muebles” y las características distintivas son coste alto y la suavidad. La suavidad de los muebles es diferente a la de las rocas, por lo que probablemente se quiera distinguir la rugosidad de la textura (ambos pueden ser considerados como suaves).

Esta metodología rara vez proporciona un árbol jerárquico de clases. Los objetos pueden pertenecer a diferentes clases y el tratar de forzar dicha estructura en algunos dominios puede ser muy poco exitoso. Cada clase no tiene una sola superclases. Sin embargo, sigue siendo sencillo comprobar si una clase es subclase de otra, comprobar el significado de una clase y determinar si la clase se corresponde con un concepto que esté en la mente.

El concepto de ontología ha evolucionado muchísimo a lo largo de la historia. Sin embargo, como ya hemos visto anteriormente, desde su aplicación al mundo de la informática, surge una nueva interpretación más ligada a las ciencias de la computación.

Existen numerosas definiciones de diversos autores acerca de las ontologías en su versión ligada a la informática. En lo que sigue, se trata de exponer una definición lo más ajustada y completa posible para que el lector se haga más a la idea de lo que se tratará posteriormente. La definición escogida es la aportada por Gruber en 1993, uno de los autores que más ha trabajado, sin duda, sobre el concepto de ontología:

*Una ontología es una especificación explícita y formal de una conceptualización compartida [7].*

Y traducido a un lenguaje más sencillo:

*Una ontología proporciona un vocabulario compartido que puede ser usado para modelar un dominio, es decir, el tipo de objetos y/o conceptos que existe y sus propiedades y relaciones.*

Una ontología define el vocabulario de un área de conocimiento mediante un conjunto de términos básicos y relaciones entre dichos términos, así como las reglas que combinan términos y relaciones que amplían las definiciones dadas en el vocabulario.

Comprendiendo esto, ya se tienen las nociones necesarias para continuar explorando las ontologías y sus posibles aplicaciones.

### ***Clasificación de las ontologías***

Las ontologías, admiten varios tipos de clasificaciones. A continuación, se ofrecen dos clasificaciones de las mismas atendiendo a diversos criterios.

Según Van Heist [11] las ontologías se pueden clasificar según el tipo de estructura de la conceptualización en:

- **Ontologías terminológicas** que 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.

Según su dependencia y relación con una tarea específica desde un punto de vista, Guarino [12] clasifica las ontologías en:

- **Ontologías de Alto nivel** o Genéricas describen conceptos generales que son independientes de un problema particular.
- **Ontologías de Dominio** describen un vocabulario relacionado con un dominio genérico especificando los términos introducidos en la ontología de mayor nivel.
- **Ontologías de Tareas** o Técnicas básicas describen una tarea o actividad especificando los términos introducidos en la ontología de mayor nivel.
- **Ontologías de Aplicación** describen conceptos que dependen tanto de un dominio específico como de una tarea específica, y generalmente son una especialización de ambas.

Las ontologías de niveles más bajos son importadas de ontologías de alto nivel y añaden conocimiento específico. Las ontologías de tarea y de dominio pueden ser independientes y se fusionan para crear ontologías de aplicación. Las ontologías de alto nivel son las más reutilizables mientras que las ontologías de aplicación son adecuadas para una sola aplicación.

Cuando se desarrolla una nueva ontología es deseable reutilizar lo máximo posible. La nueva ontología debería empezar por importar una ontología de alto nivel apropiada si es que existiera. Eso simplificaría el desarrollo y permitiría a los

desarrolladores centrarse en el conocimiento específico del dominio o aplicación. Además, sería más fácil la integración de aplicaciones en el futuro ya que compartirían partes de sus ontologías.

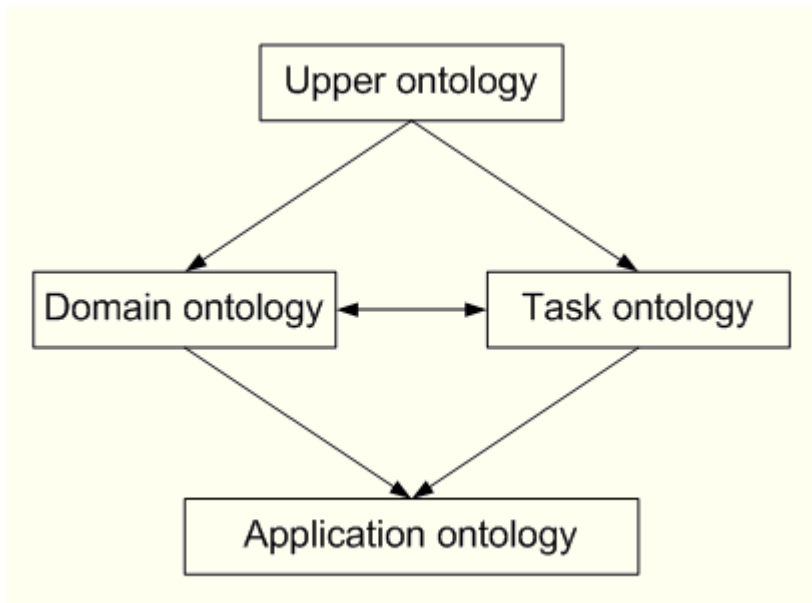


Imagen 2. Clasificación ontologías Guarino

### ***Componentes de las ontologías***

Las ontologías actuales comparten muchas similitudes estructurales, independientemente del idioma en que se expresen. Como ya se mencionó anteriormente, la mayoría de las ontologías describen a los individuos (instancias), las clases (conceptos), atributos y relaciones. Algunos componentes comunes de las ontologías son los siguientes [9]:

- **Individuos:** las instancias u objetos (los objetos básicos).
- **Clases:** conjuntos, colecciones, conceptos, clases en programación, tipos de objetos o tipo de cosas.
- **Atributos:** aspectos, propiedades, características o parámetros que los objetos (y clases) pueden poseer.
- **Relaciones:** las formas en que las clases y los individuos pueden estar relacionados entre sí.



- **Términos de función:** estructuras complejas formadas a partir de ciertas relaciones que se pueden utilizar en lugar de cualquiera de sus cláusulas en una declaración.
- **Restricciones:** descripciones declaradas formalmente de lo que debe ser cierto para que algunas afirmaciones sean aceptadas como entrada.
- **Reglas:** las declaraciones en la forma “si-entonces (antecedente-consecuente)” que describe las inferencias lógicas que se pueden extraer de una afirmación en una forma particular.
- **Axiomas:** afirmaciones (incluidas las normas) en una forma lógica que en conjunto constituyen la teoría general que la ontología describe en su dominio de aplicación, es decir, los axiomas propios de la lógica formal. Esta definición difiere de los axiomas lógicos que son sólo declaraciones reafirmadas con conocimiento previo. Tal como se usan aquí, los axiomas también incluyen la teoría derivada de las declaraciones axiomáticas.
- **Eventos:** el cambio de atributos o las relaciones.

### ***Operaciones con ontologías***

Es posible que una aplicación trabaje con diferentes ontologías cuando se integran con otros sistemas. En esos casos será necesario realizar ciertas operaciones con las ontologías para trabajar con ellas. A continuación se presentan algunas de las posibles operaciones que se pueden realizar, si bien el vocabulario que se empleará no está estandarizado y según los autores que se consulten, éste puede cambiar. A continuación listan las operaciones con ontologías definidas por Marek Obitko [2].

- **Fusión:** Creación de una nueva ontología vinculando las ya existentes. Es un requisito convencional que la nueva ontología recoja todo el conocimiento de las anteriores. Sin embargo, este requisito no tiene por qué ser totalmente satisfecho, teniendo en cuenta el hecho de que las ontologías de partida no tienen por qué ser consistentes entre sí. En la nueva ontología se pueden introducir nuevos conceptos y relaciones que sirvan de puente entre términos de la ontología de partida.
- **Mapeo:** Se trata de expresar cómo se pueden traducir los estados de una ontología a los de otra. Significa traducir los conceptos y relaciones de una en los de la otra y no siempre eso se puede hacer uno a uno. Por ello, puede que se pierda información en el mapeo pero nunca se deben introducir inconsistencias.

- **Alineación:** Es un proceso de mapeo entre ontologías en ambas direcciones existiendo la posibilidad de modificar las ontologías originales con el fin de obtener una traducción adecuada. Eso implica que es posible añadir nuevos conceptos y relaciones a las ontologías para que satisfagan la equivalencia en el mapeo.
- **Refinamiento:** Es el mapeo de una ontología O1 en otra ontología O2 de modo que cada concepto de O1 tenga su equivalente en O2. Sin embargo, los conceptos primitivos de O1 se pueden corresponder con conceptos no primitivos de O2.
- **Unificación:** es la alineación de todos los conceptos y relaciones en ontologías de tal manera que las inferencias de una ontología puedan ser mapeadas en la otra. Es un refinamiento de las ontologías en ambas direcciones.
- **Integración:** Es un proceso de búsqueda de las mismas partes en dos ontologías O1 y O2 durante el desarrollo de una tercera ontología O3 que permita la traducción entre O1 y O2 y por lo tanto permita interoperar a los sistemas que las usan. La nueva ontología O3 puede sustituir a las anteriores o servir como puente para la traducción entre ambas ontologías. Dependiendo de las diferencias entre O1 y O2 puede que no se necesite crear la ontología O3 y que el resultado de la integración sea simplemente la traducción entre O1 y O2. El nivel de integración puede ir desde el alineamiento hasta la unificación.
- **Herencia:** Significa que una ontología O1 hereda todo de otra ontología O2. Se heredan los conceptos, las relaciones y los axiomas y no se introducen inconsistencias en el nuevo conocimiento que contenga la ontología O1. Esta operación es importante en el diseño modular de ontologías donde las de alto nivel introducen conceptos generales y las de niveles más bajos añaden el conocimiento necesario para aplicaciones particulares.

No todas estas operaciones se pueden aplicar a todas las ontologías. En general, son tareas difíciles de llevar a cabo y no existe una forma automática de realizarlas al completo.

## ***Relaciones entre ontologías***

Una clasificación del grado de traducción entre dos ontologías es la que se propone a continuación, la proporcionada por FIPA Ontology Service Specification y que Marek Obitko expuso en 2007 [2].

- **Extensión:** Una ontología O1 extiende a otra ontología O2 cuando todos los símbolos en O2 están definidos en O1 junto a todas las restricciones, significados y relaciones axiomáticas de dichos símbolos de O2.
- **Idénticas:** O1 y O2 son ontologías idénticas cuando el vocabulario, la axiomatización y el lenguaje son físicamente idénticos y sólo varían en la nomenclatura.
- **Equivalentes:** O1 y O2 son equivalentes cuando el vocabulario lógico y la axiomatización lógicas son los mismos pero el lenguaje (la sintaxis) es diferente. Además son fuertemente traducibles en ambos sentidos.
- **Fuertemente traducibles:** O1 es fuertemente traducible en O2 cuando el vocabulario de O1 puede ser traducido totalmente en el de O2, la axiomatización de O1 se sostiene en la de O2, no hay pérdida de información al traducir y no se introduce ninguna inconsistencia. Sin embargo, los lenguajes pueden ser diferentes.
- **Débilmente traducibles:** O1 es débilmente traducible en O2 si permite pérdida de información al traducir pero no se introduce ninguna inconsistencia.
- **Aproximadamente traducibles:** O1 es aproximadamente traducible en O2 si permite incluso la introducción de inconsistencias.

## **Ontologías y SBCs (Sistemas basados en conocimiento)**

Esta sección es el resultado de la adaptación de múltiples partes del libro de Poole y Mackworth [3].

Relacionando este tema con los SBCs, nos encontramos con que las ontologías son fundamentales a la hora de resolver dos de los principales problemas que existen en ese tipo de sistemas:

- La adquisición de conocimiento y cómo representar el conocimiento humano en una representación abstracta efectiva.
- La representación del conocimiento en términos de una estructura de datos que una máquina pueda procesar.

Según Smith [13], el problema más serio en el desarrollo de una teoría de la computación adecuada es más ontológico que semántico. Esto no significa que los problemas semánticos desaparezcan; siguen estando ahí como un desafío. Simplemente están unidos, en primer plano, a los problemas ontológicos que precisan de solución.

¿Cómo conseguir una representación del conocimiento del mundo en la que sea fácil adquirir, depurar, mantener, comunicar, compartir y razonar sobre dicha representación? Se debe explorar la forma en la que se especifica el significado de los símbolos en los agentes inteligentes, la forma en la que se usa el significado cuando se depura y se realizan explicaciones basadas en conocimiento, y por supuesto, la forma en la que un agente representa su propio razonamiento y la forma en la que se debe usar para construir sistemas basados en conocimiento. Como Smith señalaba anteriormente los problemas de ontología se dan principalmente en la construcción de agentes computacionales inteligentes.

### ***Intercambio de conocimiento***

Tener una representación adecuada es sólo una parte del proceso de construcción de un agente del conocimiento. También se debe ser capaz de asegurar que los conocimientos se puedan adquirir, incluso cuando el conocimiento proviene de diversas fuentes y en diferentes momentos de tiempo y se debe interoperar con otro conocimiento. También se debe garantizar que el conocimiento se puede razonar de forma eficaz.

Recordando puntos anteriores, una ontología es una especificación de los significados de los símbolos en un sistema de información. Aquí, un sistema de información es una base de conocimientos o cualquier fuente de información, como por ejemplo un termómetro. El significado está a veces sólo en la mente del diseñador de la base de conocimientos o en los comentarios puestos en la misma. Pero cada vez más, se pretende que la especificación del significado esté en una forma interpretable por las máquinas. Esta especificación formal es importante para la interoperabilidad semántica, es decir, la capacidad de las diferentes bases de conocimiento para trabajar juntas.

*Un agente de compras tiene que saber, cuando un sitio web afirma que tiene un buen precio en "chips", cuando se trata de patatas fritas, chips informáticos, fichas de póquer o cualquier tipo de "chip" atendiendo a sus distintas acepciones en inglés como por ejemplo "chip de madera" que significa viruta. Una ontología especificaría el significado de la terminología utilizada por el sitio web. En lugar de utilizar el símbolo "chip", un sitio web que se adhiere a las ontologías puede usar el símbolo "WoodChipMixed" según lo definido por alguna organización particular, que haya publicado una ontología. Mediante el uso de este símbolo y*

*declarando la ontología a la que pertenece, la utilización de la palabra "viruta" debe ser inequívoca. Una representación formal de la página web, usaría "WoodChipMixed", que puede traducirse en Inglés simplemente como "chip". Si otra fuente de información utiliza el símbolo "ChipOfWood", un tercero podrá declarar que el uso del término "ChipOfWood" en esa fuente de información corresponde a una "WoodChipMixed" y por lo tanto permitir que las fuentes de información puedan ser combinadas. En ese caso se estaría realizando un mapeo entre dos ontología [3].*

Antes de discutir cómo se especifican las ontologías, se analizará cómo la lógica (con variables, términos y relaciones) se puede utilizar para construir representaciones flexibles que permiten la adición modular de conocimiento, incluyendo la adición de argumentos a las relaciones.

Dada una especificación del significado de los símbolos, un agente puede usar ese significado para la adquisición de conocimiento, la explicación y la depuración en el nivel de conocimiento.

## **Representaciones flexibles**

Esta sección es el resultado de la adaptación de múltiples partes del libro de Poole y Mackworth [3].

A continuación se considerará una forma de construir representaciones flexibles utilizando las herramientas de la lógica. Estas representaciones flexibles son la base de las ontologías modernas.

### ***Elección de los individuos y las relaciones***

Dado un lenguaje lógico de representación y un mundo para razonar sobre él, la gente que diseña bases de conocimiento debe elegir a qué referirse. Es decir, hay que elegir qué individuos y qué relaciones existen. Sin embargo, el mundo no determina los individuos que existen. Cómo el mundo se divide en individuos es determinado por quien está modelando el mundo. El modelador divide el mundo en porciones para que el agente pueda hacer referencia a las partes del mundo que tengan sentido para la tarea en cuestión.

*Puede parecer como que "rojo" es una propiedad razonable para atribuir a las cosas en el mundo. En el mundo, hay superficies que absorben algunas frecuencias y reflejan otras frecuencias de la luz. Algunos usuarios pueden haber decidido que, para alguna aplicación, un conjunto particular de propiedades de reflexión se debe llamar "rojo". Algún otro modelador del dominio puede decidir sobre otra asignación del espectro y sobre el uso de los términos de colores rosa,*

*escarlata, rubí y carmesí. Otro modelador puede dividir el espectro en regiones que no se corresponden a las palabras en ningún idioma, pero que son más útiles para distinguir las diferentes categorías de individuos [3].*

Como modelador se debe elegir qué individuos representar, y también escoger las relaciones que usar. Hay, sin embargo, algunos principios que son útiles para la elección de las relaciones y los individuos. Éstos se mostrarán a través de una secuencia de ejemplos.

*Si se decide que "rojo" es una categoría adecuada para la clasificación de los individuos se puede tratar el nombre rojo como una relación unaria y describir que un individuo (a) es de color rojo:*

*Red(a)*

*Si representa la información de color de esta manera, entonces se puede preguntarse qué es de color rojo:*

*Ask(red(x))*

*Lo que se devuelve en x son todos los individuos de color rojo.*

*Con esta representación, es difícil hacer la pregunta, "¿De qué color es un individuo?" ya que en la sintaxis de las cláusulas definidas, no se puede preguntar*

*Ask X(a)*

*porque en los lenguajes basados en la lógica de primer orden, los nombres de predicado no pueden ser variables. En lógica de segundo orden o superior, se devolvería cualquier propiedad que tuviera el individuo "a".*

*Hay representaciones alternativas que permiten preguntar por el color del individuo a. No hay nada en el mundo que obligue a hacer un predicado de color rojo. Por lo tanto, se podría fácilmente decir que los colores son individuos también, y se podría utilizar el rojo como constante para indicar el color rojo. Teniendo en cuenta que el rojo es una constante, se puede utilizar el predicado color donde color(Ind, Val) significa que Ind individuo físico tiene color Val. "El individuo a es rojo", ahora se puede escribir como*

*Color(a,red)*

*Lo que se debe hacer es repensar el mundo: el mundo ahora se compone de colores como las personas que se pueden nombrar. Ahora hay una nueva relación binaria "color" entre las personas físicas y los colores. Bajo esta nueva representación se puede preguntar: "¿De qué color es un individuo?" con la consulta*

*Ask color(a,C)*

Para hacer que un concepto abstracto sea objeto se debe reificar. En el ejemplo anterior, se ha reificado el color rojo.

*Parece que no hay ningún inconveniente para la nueva presentación de colores del ejemplo anterior. Todo lo que se podía hacer antes, se puede hacer ahora. Además, no es mucho más difícil escribir  $\text{color}(x, \text{Red})$  que  $\text{red}(X)$ , pero ahora se puede preguntar sobre el color de las cosas. Entonces, surge la pregunta de si se puede hacer esto para cada relación y con qué se termina.*

*Se puede hacer un análisis similar para el predicado color y para el predicado red del ejemplo anterior. La representación con color como predicado no permite la pregunta ¿qué propiedad del individuo a tiene valor “rojo”? donde la respuesta adecuada sería “color”. Llevando a cabo una transformación similar a la del ejemplo anterior, se pueden ver propiedades como el color como individuo y se puede crear una relación de apoyo y escribir “El individuo a tiene color rojo” como:*

*$\text{prop}(a, \text{color}, \text{red})$ .*

*Esta representación permite todas las consultas de éste y del ejemplo anterior. Además, se pueden escribir todas las relaciones en términos de relaciones de apoyo.*

La representación “individuo-propiedad-valor” está en términos de una sencilla relación de apoyo donde

$\text{prop}(\text{Ind}, \text{Prop}, \text{Val})$

significa que ese individuo “Ind” tiene valor “Val” para la propiedad “Prop”. Esto también es llamado triple representación porque todas las relaciones están representadas como triplets. El primer elemento es llamado sujeto, el segundo es el verbo y el tercero es el objeto, usando la analogía con una sencilla frase de tres palabras.

El verbo de la tripleta es una propiedad. El dominio de propiedad p es el conjunto de individuos que pueden aparecer como sujeto de un triplete cuando p es el verbo. El rango de una propiedad es el conjunto de valores que pueden aparecer como objeto en el triplete cuando p es el verbo.

Un atributo es una pareja “propiedad-valor”. Por ejemplo, un atributo de un individuo puede ser que su color es rojo. Dos individuos pueden ser idénticos si tienen los mismos atributos, es decir, los mismos valores para todas sus propiedades.

Hay algunos predicados que pueden parecer demasiado simples para la representación en triplets.

*Para transformar  $\text{parcel}(a)$ , que significa que a es una paquete, no parece adecuado usar propiedades o valores. Existen 2 formas de transformar esto en una representación de triplete. La primera es re-hacer el concepto paquete y decir que a es un paquete:*

*prop(a, type, parcel).*

*Aquí, type es una propiedad especial que relaciona a un individuo con una clase. La constante parcel denota la clase que es el conjunto de todas las cosas que son paquetes (reales o potenciales). Esta tripleta especifica que el individuo a está en la clase paquete.*

*La segunda forma es convertir paquete en propiedad y escribir “a es un paquete” como*

*prop(a, parcel, true).*

*En esta representación, parcel es una propiedad booleana que es cierta para los individuos que son paquetes.*

Una propiedad booleana es una propiedad cuyo rango de valores es {true, false}, donde true y false son símbolos de constante en el lenguaje.

Algunos predicados pueden parecer demasiado complicados para la triple representación:

*Si se quiere representar la relación*

*scheduled(C, S, T, R),*

*que significa que la sección S del curso C está programada para empezar a la hora T en el cuarto R. Por ejemplo, “sección 2 del curso cs422 está programada para empezar a las 10:30 en el cuarto cc208” es escrito como:*

*scheduled(cs422, 2, 1030, cc208).*

*Para representar esto con la representación de tripletas, se puede inventar un nuevo individuo, una reserva. Así, la relación scheduled es re-hecha como un individuo reserva.*

*Una reserva tiene una serie de propiedades como nombre del curso, sección, hora de inicio y cuarto. Para representar “sección 2 del curso cs422 está programada para empezar a las 10:30 en el cuarto cc208”, se llama a la reserva con una constante (por ejemplo b123) y se escribe*

*prop(b123, course, cs422).*

*prop(b123, section, 2).*

*prop(b123, start\_time, 1030).*

*prop(b123, room, cc208).*

*Esta nueva representación tiene una serie de ventajas. La más importante es que es modular ya que cada valor va con su propiedad y puede ser visto fácilmente y es fácil añadir nuevas propiedades como el instructor o la duración. Con esta nueva representación, es fácil añadir que “Fran enseñará la sección 2*



del curso cs422, programado para las 10:30 en el cuarto cc208" o que la duración es de 50 minutos:

*prop(b123, instructor, fran).*

*prop(b123, duration, 50).*

Con *scheduled* como predicado, era muy difícil añadir el instructor o la duración porque requería añadir argumentos extra para cada instancia del predicado.

## Representaciones gráficas

Se puede interpretar la relación de apoyo gráficamente, donde la relación

Prop(Ind, Prop, Val)

es representada con Ind y Val como nodos con un arco llamado Prop entre ellos. Este gráfico es llamado red semántica visto anteriormente como uno de los posibles formalismos de representación de información en los SBCs. Dada esta representación gráfica, hay una asignación directa a una base de conocimiento usando la relación de apoyo.

La figura siguiente muestra una red semántica para un robot de reparto mostrando el tipo de conocimiento que el robot debe tener acerca de una computadora particular. Una parte del conocimiento representado en la red es:

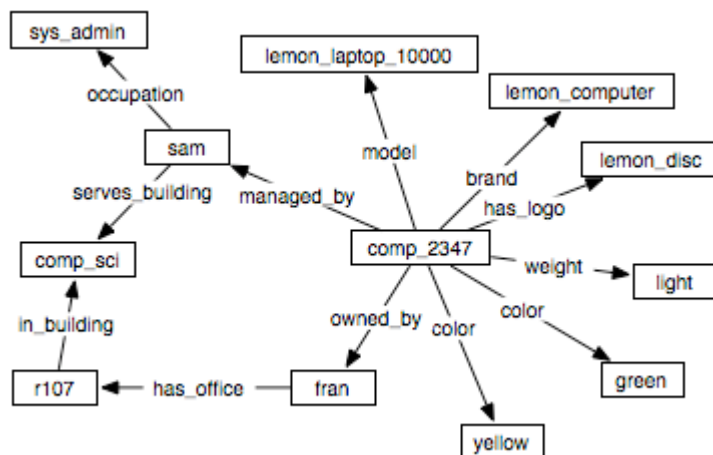


Imagen 3. Red semantica ejemplo

*prop(comp\_2347 , owned\_by , fran).*

*prop(comp\_2347 , managed\_by , sam).*

*prop(comp\_2347 , model , lemon\_laptop\_10000).*

*prop(comp\_2347 , brand , lemon\_computer).*

```
prop(comp_2347 , has_logo , lemon_disc).  
prop(comp_2347 , color , green).  
prop(comp_2347 , color , yellow).  
prop(comp_2347 , weight , light).  
prop(fran , has_office , r107).  
prop(r107 , in_building , comp_sci).
```

*La red también muestra cómo está estructurado el conocimiento. Por ejemplo, es fácil ver que el computador número 2347 pertenece a alguien (Fran) cuya oficina (r107) está en el edificio comp\_sci. La indexación directa evidente en el gráfico puede ser usada por humanos y máquinas.*

Esta notación gráfica tiene una serie de ventajas:

1. Es fácil para un humano ver las relaciones sin necesidad de aprender la sintaxis de una lógica particular. La notación gráfica ayuda al constructor de bases de conocimiento a organizar su conocimiento.
2. Se pueden ignorar las etiquetas de nodos que carecen de nombres con sentido, por ejemplo, el nombre b123 del ejemplo de las reservas o comp\_2347 de la figura anterior. Se pueden dejar esos nodos en blanco y construir un nombre arbitrario si se debe asignar con una forma lógica.

## ***Relaciones primitivas frente a derivadas***

Normalmente, se sabe más sobre un dominio que de una base de datos de hechos. Se conocen normas generales con las cuales se pueden derivar otros hechos. Qué hechos están explícitamente determinados y cuáles se derivan es una decisión que se hará en el diseño y la construcción de una base de conocimiento.

El conocimiento primitivo es el conocimiento que se define explícitamente por los hechos. El conocimiento derivado es el conocimiento que se puede deducir de otros conocimientos. Los conocimientos obtenidos se suelen especificar en términos de normas.

El uso de normas permite una representación más compacta de los conocimientos. Las relaciones derivadas permiten extraer conclusiones a partir de observaciones del dominio. Esto es importante porque normalmente no se observa directamente todo lo relacionado con un dominio. Mucho de lo que se sabe acerca de un dominio se infiere de las observaciones y de conocimiento más general.

Una manera estándar de utilizar los conocimientos derivados es organizar a los individuos en clases. Se definen propiedades generales a las clases que los individuos heredan. La razón por la que se agrupan los individuos en clases se

debe a que los miembros de una clase tienen atributos en común, o propiedades comunes que tengan sentido para ellos.

Una clase es el conjunto de los individuos reales y potenciales que serían miembros de dicha clase. En lógica, se trata de un conjunto intensional, definido por una función característica que es cierta en los miembros del conjunto y falsa en otros individuos. La alternativa es un conjunto extensional, que se define por la lista de sus elementos. Por ejemplo, la clase silla es el conjunto de todas las cosas que serían sillas. No se quiere que la definición sea el conjunto de cosas que son sillas, porque las sillas que aún no se han construido también pertenecen a la clase sillas. Tampoco quiere decir que dos clases sean equivalentes sólo porque tienen los mismos miembros.

La definición de clase permite que cualquier conjunto pueda ser descrito como una clase.

*Por ejemplo, el conjunto compuesto por el número 17, la Torre de Londres, y el pie izquierdo de Julio César puede ser una clase, pero no es muy útil.*

Un tipo natural de clase es aquella donde describir objetos con la clase es más breve que describirlos objetos sin ella.

*Por ejemplo, "mamífero" es una clase natural, ya que describir los atributos comunes de los mamíferos hace que una base de conocimientos que utiliza "mamífero" como clase sea más rápida de usar que otra que describe para cada mamífero todos sus atributos, es decir, que no usa la clase "mamíferos".*

Se utiliza la propiedad *tipo* en el sentido de "es un miembro de la clase." Así, en el lenguaje de las cláusulas definitivas,

$\text{prop}(X, \text{tipo}, C)$

significa que cada individuo  $X$  es un miembro de la clase  $C$ .

La relación entre los tipos y subclases se puede escribir como una cláusula definida:

$\text{prop}(X, \text{tipo}, C) \leftarrow \text{prop}(X, \text{tipo}, S) \wedge \text{prop}(S, \text{subClassOf}, C)$

Se puede tratar el tipo y subClassOf como propiedades especiales que permiten la herencia de propiedades. Herencia de la propiedad es cuando el valor de una propiedad se especifica en el nivel de clase y es heredado por los miembros de la clase. Si todos los miembros de la clase  $C$  tienen valor  $v$  para la propiedad  $p$ , esto puede ser escrito como

$\text{prop}(\text{Ind}, p, v) \leftarrow \text{prop}(\text{Ind}, \text{tipo}, c)$

que, junto con la norma mencionada que relaciona los tipos y subclases, se pueden utilizar para la herencia de propiedades.

*Ejemplo: En una empresa de computadoras llamada “Limón”, todos los equipos tienen un disco de limón como logotipo y tienen color verde y amarillo (véase los arcos logo y color en la siguiente figura). Esto puede ser representado por*

*prop (X, has\_logo, lemon\_disc)  $\leftarrow$  prop (X, tipo, lemon\_computer).*

*prop (X, color verde)  $\leftarrow$  prop (X, tipo, lemon\_computer).*

*prop (X, el color, amarillo)  $\leftarrow$  prop (X, tipo, lemon\_computer).*

*Las relaciones prop que se pueden derivar de estas cláusulas son esencialmente las mismas que se pueden derivar de la red semántica plana de la imagen del apartado “representaciones gráficas”. Con la representación estructurada, si se incorpora un nuevo Lemon\_laptop\_10000, sólo podrá declarar que es un Lemon\_laptop\_10000 y las propiedades de color y el logotipo se derivan a través de la herencia.*

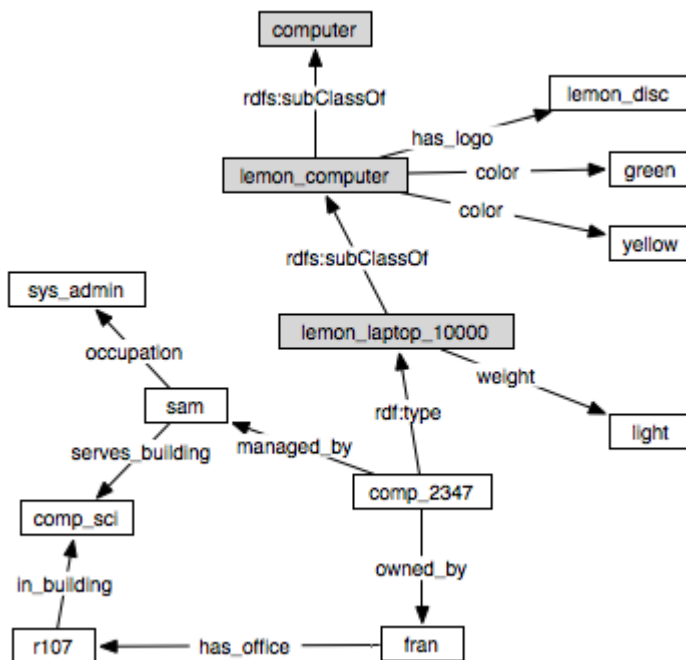


Imagen 4. Red semántica que permite herencia

Algunos consejos generales útiles para decidir lo que debe ser primitivo y lo que debería ser derivado son:

- Al asociar un atributo con un individuo, seleccionar la clase C, más general, de manera que el individuo se encuentra en C y todos los miembros de C tengan ese atributo, y asociar el atributo con la clase C. La herencia puede ser utilizada para obtener el atributo del individuo y

todos los miembros de la clase C. Este método de representación tiende a hacer las bases de conocimiento más concisas, y eso significa que es más fácil incorporar nuevas personas, ya que heredan automáticamente el atributo si es un miembro de la clase C.

- No asociar un atributo contingente de una clase con la clase. Un atributo contingente es aquel cuyo valor cambia cuando cambian las circunstancias. Por ejemplo, puede ser cierto para el entorno informático actual que todos los equipos se venden en cajas marrones. Sin embargo, no es buena idea poner eso como un atributo de la clase computador, ya que puede que deje de ser cierto.
- Axiomatizar en la dirección de causalidad. Si existe la opción de hacer la causa primitiva o el efecto primitivo, se debe hacer primitiva la causa. De esta forma, es más probable que la información sea estable cuando el dominio cambie.

### ***Las clases en Bases de Conocimiento y Programación Orientada a Objetos***

El uso de "individuos" y "clases" en los sistemas basados en el conocimiento es muy similar al uso de "objetos" y "clases" en la programación orientada a objetos (POO) en lenguajes como Smalltalk o Java. Esto no debería ser demasiado sorprendente, ya que tienen una larga historia común. Sin embargo, hay diferencias importantes que tienden a hacer la analogía directa sea más confusa que útil:

- Los objetos en programación orientada a objetos son objetos de cómputo, son estructuras de datos y programas asociados. Una "persona" objeto en Java no es una persona. Sin embargo, los individuos en una base de conocimiento (KB) son típicamente cosas en el mundo real. Una "persona" en una KB puede ser una persona real. Un individuo de "silla" puede ser una silla real donde sentarse. En Java, se puede enviar un mensaje a una silla objeto y obtener respuesta, mientras que una silla en el mundo real tiende a ignorar lo que le dicen. Una KB no se suele utilizar para interactuar con una silla, sino para razonar sobre una silla. Una silla real se queda donde está, a menos que sea movida por un agente físico.
- En una KB, una representación de un objeto es sólo una aproximación a uno (o varios) niveles de abstracción. Los objetos reales tienden a ser mucho más complicados que lo que se representa. Por lo general no se representan las fibras individuales en una silla. En un sistema de programación orientado a objetos, están sólo las propiedades que se representan de un objeto. El sistema lo puede saber todo acerca de un objeto de Java, pero no se trata de un individuo real.

- La estructura de clases de Java representa intencionadamente objetos diseñado por un analista o programador. Por ejemplo, en Java, un objeto es sólo un miembro de una clase de nivel más bajo. No hay herencia múltiple. Los objetos reales no son tan bien “educados”. La misma persona puede ser un entrenador de fútbol, un matemático, y un padre.
- Un programa de ordenador no puede tener dudas sobre sus estructuras de datos. tiene que seleccionar las estructuras de datos particulares que va a usar. Sin embargo, no podemos estar seguros del tipo de cosas que hay en el mundo.
- Las representaciones en una KB en realidad no hacen nada. En un sistema de programación orientada a objetos, los objetos realizan tareas computacionales. En una KB, sólo representan, es decir, se refieren a objetos en el mundo.
- Aunque un lenguaje de modelado orientado a objetos, como UML, se puede utilizar para la representación de KB, no es la mejor opción. Una herramienta de modelado orientada a objetos ofrece facilidades para crear buenos diseños. Sin embargo, el mundo que se está modelando puede no tener un buen diseño. Tratar de forzar un paradigma bien diseñado en un mundo desordenado, puede no ser productivo.

## Ontologías y conocimiento compartido

Los servicios y sistemas basados en conocimiento son caros de construir, probar y mantener. Se necesita una metodología de ingeniería del software basada en unas especificaciones formales para el conocimiento compartido, componentes reutilizables y servicios estándar. Se cree que las especificaciones para un vocabulario compartido pueden jugar un papel fundamental en dicha metodología.

Existen muchos problemas técnicos en el desarrollo de software basado en conocimiento reutilizable y compartido. Al igual que las aplicaciones convencionales, los sistemas basados en conocimiento están basados en plataformas hardware heterogéneas, lenguajes de programación y protocolos de red. Sin embargo, los sistemas basados en conocimiento tienen requisitos especiales para su interoperabilidad. Dichos sistemas operan y se comunican usando declaraciones en una representación formal del conocimiento. Toman el conocimiento básico como entrada y negocian e intercambian conocimientos como agentes en un entorno distribuido de Inteligencia Artificial.

Construir sistemas grandes basados en conocimiento es complejo porque [3]:

- El conocimiento normalmente viene de múltiples fuentes y deben ser integradas. Además, estas fuentes pueden no tener la misma división del

mundo. Normalmente el conocimiento viene de diferentes campos que tienen terminologías distintivas propias y dividen el mundo de acuerdo con sus necesidades.

- Los sistemas evolucionan con el tiempo y es difícil anticipar todas las futuras distinciones que deberán hacerse.
- La gente involucrada en el diseño de bases de conocimiento debe elegir qué individuos y relaciones representar. El mundo no está dividido en individuos; eso es algo hecho por agentes inteligentes para entender el mundo. Las distintas personas involucradas en la construcción de sistemas basados en conocimiento deberían estar de acuerdo en la división del mundo.
- Si ya es difícil recordar lo que significa la notación de uno mismo, mucho más difícil es descubrir lo que significa la notación de otra persona. Esto se puede mirar de dos formas diferentes:
  - Dado un símbolo usado en una computadora, determinar lo que significa.
  - Dado un concepto en la mente de alguien, determinar qué símbolo se debería usar; esto es determinar si el concepto ha sido usado antes y, si es así, descubrir qué notación ha sido usada para él.

Para compartir y comunicar conocimiento, es importante ser capaz de llegar a un vocabulario común y ponerse de acuerdo en el significado de ese vocabulario.

Una conceptualización es una correspondencia entre símbolos usados en el ordenador (ej: el vocabulario) y los individuos y relaciones del mundo. Proporciona una abstracción particular del mundo y la notación para esa abstracción. Una conceptualización para pequeñas bases de conocimiento puede estar en la cabeza del diseñador o especificada en lenguaje natural dentro de la documentación. Esta especificación informal de una conceptualización no es escalable para sistemas más grandes donde la conceptualización debe estar distribuida.

En filosofía, una ontología es el estudio de lo que existe. En IA, una ontología es una especificación del significado de los símbolos en un sistema de información, es decir, una especificación de una conceptualización (Gruber, 1993). Es una especificación de qué individuos y relaciones se supone que existen y qué terminología es usada para ellos. Típicamente, especifica qué tipos de individuos serán modelados, especifica qué propiedades serán usadas y proporciona algunos axiomas que restringen el uso de ese vocabulario.

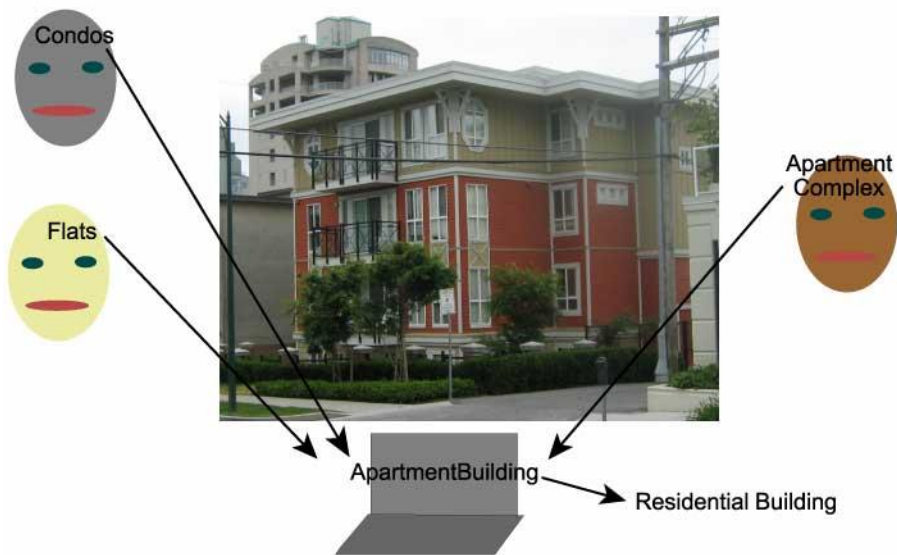


Imagen 5. Ejemplo apartamentos

*Una ontología de individuos que podrían aparecer en un mapa podría especificar que el símbolo “ApartmentBuilding” represente los edificios de apartamentos. La ontología no definirá un edificio de apartamentos, pero lo describirá de forma suficiente para que otros puedan entender la definición. Se quiere que otra gente, que pueda preferir el uso de otros símbolos, sea capaz de usar la ontología para encontrar el símbolo apropiado que necesita (ver figura anterior). Mucha gente será capaz de usar el símbolo de forma consistente. Una ontología debería también permitir a una persona verificar lo que el símbolo significa. Eso es, dado un concepto, ser capaz de encontrar el símbolo y, dado el símbolo, ser capaz de determinar lo que significa.*

*Una ontología debe proporcionar axiomas que restrinjan el uso de algunos símbolos. Por ejemplo, debe especificar que los edificios de apartamentos son edificios, que son construcciones humanas. Debe dar alguna restricción del tamaño de los edificios para que las cajas de zapatos no puedan ser edificios o que las ciudades no puedan ser edificios. Debe afirmar que un edificio no puede estar en 2 lugares diferentes al mismo tiempo (si quitas una parte de un edificio y lo llevas a otro lugar, ya no es un edificio simple). Porque los edificios de apartamentos son edificios, estas restricciones también se aplican a los edificios de apartamentos [3].*

Las ontologías son normalmente escritas de forma independiente a una aplicación particular y frecuentemente involucran a una comunidad para estar de acuerdo con el significado de los símbolos. Una ontología consiste en:

- Un vocabulario de las categorías de las cosas (clases y propiedades) que la base de conocimiento debe representar.



- Una organización de las categorías, por ejemplo una jerarquía de herencia usando subclases de o sub-propiedades de, o usando definiciones Aristotélicas.
- Un conjunto de axiomas restringiendo los significados de algunos de los símbolos para que reflejen mejor su significado, por ejemplo que alguna propiedad sea transitiva, o que el dominio y el rango estén restringidos, o que haya alguna restricción en el número de valores que una propiedad puede tomar para cada individuo. Algunas veces, las relaciones son definidas en términos de más relaciones primitivas pero, al final, las relaciones se basan en relaciones primitivas que no están definidas.

Una ontología no especifica a los individuos no conocidos durante la fase de diseño. Por ejemplo, una ontología de edificios no debería incluir los edificios actuales. Especificaría aquellos individuos que son fijos y que deberían ser compartidos como los días de la semana o los colores.

*Considérese un agente de ventas que está diseñado para buscar alojamientos. Los usuarios podrían usarlo para describir qué alojamiento quieren. El agente podría buscar en distintas bases de conocimiento para encontrar el alojamiento más adecuado y para notificar al usuario cuándo estará disponible. Una ontología requiere especificar el significado de los símbolos para el usuario y para permitir que las distintas bases de conocimiento interactúen. Proporciona el pegamento semántico para vincular las necesidades de los usuarios a las bases de conocimiento.*

*En dicho dominio, casas y edificio de apartamentos deben ser edificios residenciales. Aunque sería razonable sugerir el alquiler de una casa o un apartamento en un edificio de apartamentos, no sería razonable sugerir el alquiler de un edificio de apartamentos a alguien que no especifique que quiere alquilar el edificio completo. Una "unidad de vivienda" podría ser definida como un conjunto de habitaciones donde viven conjuntamente un grupo de personas. Una unidad de vivienda también puede ser lo que una agencia de alquiler ofrezca en alquiler. En algún punto, el diseñador deberá decidir sobre si una habitación de alquiler en una casa es una unidad de vivienda o incluso si parte de una habitación compartida que es alquilada por separado lo es. Frecuentemente, los casos límite, que son los no previstos inicialmente, no están claramente delineados pero quedan mejor definidos a medida que la ontología evoluciona.*

*La ontología no contendría descripciones de las casas o apartamentos actuales porque el alojamiento disponible cambiaría a lo largo del tiempo. Sin embargo, el significado del vocabulario no cambiaría [3].*

El principal objetivo de una ontología es documentar qué significan los símbolos (el mapeo entre los símbolos de la computadora y los conceptos). Dado un símbolo, una persona es capaz de usar la ontología para determinar qué significa. Cuando alguien necesita representar un concepto, la ontología es usada para encontrar el símbolo apropiado o para determinar que dicho concepto no existe en la ontología. El otro propósito, alcanzado con el uso de axiomas, es permitir inferencias para determinar si una combinación de valores es inconsistente. El mayor desafío en la construcción de ontologías es la organización de los conceptos para permitir a los humanos mapearlos con símbolos en la computadora y para permitir a la computadora inferir nuevo conocimiento útil de los hechos establecidos.

### ***Lógica de descripción***

Este apartado es el resultado de la adaptación de múltiples partes del libro de Poole y Mackworth [3] y del libro de van Harmelen, Lifschitz y Porter [14].

Las lógicas de descripción son una familia de lenguajes de representación del conocimiento que pueden ser usados para representar el conocimiento de un dominio de aplicación de una forma estructurada. Su nombre proviene del hecho de que las nociones importantes del dominio están descritas por descripciones de los conceptos. Además, las lógicas descriptivas difieren de sus predecesoras como redes semánticas y marcos en que están equipadas con semánticas formales basadas en la lógica.

Las descripciones emplean constructores booleanos: conjunciones ( $\Pi$ ) que se interpreta como el conjunto intersección, disyunciones ( $\cup$ ) que se interpreta como el conjunto unión y negación ( $\neg$ ) que se interpreta como el conjunto complemento. También utilizan los constructores restricción existencial ( $\exists$ ), restricción universal ( $\forall$ ) y restricción de valor.

Las descripciones de conceptos pueden ser usadas para construir sentencias en una base de conocimiento que consta de dos partes: la terminológica y la de aserciones.

En la parte terminológica se pueden describir las nociones relevantes del dominio de una aplicación fijando las propiedades, los roles y las relaciones entre ellos. En su forma más simple, permite introducir nombres (abreviaciones) para descripciones complejas. El conocimiento que se puede describir con la lógica de primer orden, también se puede describir con lógica descriptiva pero de una forma mucho más sencilla para ser leída. De hecho, la capacidad expresiva de las lógicas descriptivas es mayor y son perfectas para combinar expresividad y decidibilidad en algunos problemas importantes de razonamiento.

La parte de aserciones se usa para describir una situación concreta fijando las propiedades de los individuos.

Las lógicas descriptivas modernas proporcionan servicios de razonamiento que pueden deducir automáticamente el conocimiento implícito desde el conocimiento representado explícitamente, y siempre encuentran una respuesta correcta en un tiempo finito. Este razonamiento tiene en cuenta tanto la parte terminológica como la de aserciones de la base de conocimiento. El algoritmo de subsunción determina las relaciones de sub-conceptos y súper-conceptos. El algoritmo de instancias determina las relaciones de instancia. El algoritmo de consistencia determina si la base de conocimientos (que consiste en un conjunto de aserciones y un conjunto de axiomas terminológicos) es o no contradictoria.

En una aplicación, se puede empezar construyendo la parte terminológica y utilizar el razonador para asegurarse de que ningún concepto es subsumido por el concepto “fondo”, que siempre se interpreta como un conjunto vacío. Además, se puede utilizar el algoritmo de subsunción para establecer la jerarquía de de subsunciones y comprobar si coincide con las intenciones del modelador. Después, se puede definir la parte de aserciones y utilizar el razonador para comprobar la consistencia e identificar los individuos que son instancias de la descripción de los conceptos.

Para asegurar un comportamiento razonable y predecible del sistema DL, estos problemas de inferencia deberían por lo menos ser decidibles para la DL empleada en el sistema y preferiblemente de baja complejidad. Por eso, la capacidad expresiva de la DL en cuestión debe ser restringida de la forma apropiada. Si las restricciones son muy severas, sin embargo, nociones importantes del dominio no podrían ser expresadas.

Un identificador uniforme de recursos (URI) tiene un significado porque alguien publicó que tiene ese significado y porque la gente lo usa con ese significado. Esto funciona, pero estaría bien disponer de significados que permitan a un ordenador realizar inferencias.

Los lenguajes modernos de ontologías se basan en la lógica de descripción. Una lógica de descripción se utiliza para describir clases, propiedades e individuos. Una de las ideas principales detrás de una lógica de descripción es separar la base de conocimientos en dos partes como ya se ha indicado antes:

- Una base de conocimientos que describa la terminología, la cual debe permanecer constante mientras se modelan los cambios en el dominio.
- Una base de conocimientos de aserciones que describe lo que es verdad en algún dominio en algún momento en el tiempo.

Por lo general, la base de conocimiento terminológica se define en tiempo de diseño del sistema y define la ontología, y sólo cambia el significado con los cambios de vocabulario, que deberían ser raros. La base de conocimientos de

aserciones por lo general, contiene el conocimiento de una situación específica y sólo se conoce en tiempo de ejecución.

Es habitual el uso de tripletas para definir la base de conocimientos de aserciones y un lenguaje de ontologías moderno para definir la base de conocimiento de la terminología.

Alguien que defina o que quiera adoptar una ontología debe asegurarse de que utiliza las propiedades y las clases en el mismo sentido que los demás usuarios de la ontología.

Una ontología de domino es una ontología sobre un dominio de interés particular. La mayoría de ontologías existentes están en un dominio estrecho que la gente escribe para aplicaciones específicas. Hay algunas pautas que se han desarrollado para la escritura de ontologías de dominio para permitir el intercambio de conocimiento:

- Si es posible, utilizar una ontología existente. Esto significa que la base de conocimientos será capaz de interactuar con otras personas que utilizan la misma ontología.
- Si una ontología existente no coincide exactamente con las necesidades, se importa y se añade conocimiento a la misma. No empezar de cero, ya que así los que quieran usar la "ontología mejorada" tendrán que elegir. Si la ontología elaborada incluye y mejora la otra, otras personas que quieran adoptar una ontología elegirán la nueva, porque será capaz de interactuar con los adoptantes de cualquier ontología.
- Asegurarse de que la ontología se integra con las ontologías de los dominios relacionados. Por ejemplo, una ontología sobre centros turísticos tendrá que interactuar con ontologías sobre la comida, playas, actividades recreativas, y así sucesivamente. Hay que tratar de asegurarse de que se utiliza la misma terminología para las mismas cosas.
- Tratar de encajar con ontologías de nivel superior (se definirán posteriormente). Esto hará que sea mucho más fácil para otras personas integrar sus conocimientos.
- Si tiene que diseñar una nueva ontología, hacer una amplia consulta con otros usuarios potenciales. Esto hará que sea más útil y tenga más probabilidades de ser adoptada.
- Seguir las convenciones de nomenclatura. Por ejemplo, llamar a una clase por el nombre singular de sus miembros. Por ejemplo, llamar a una clase "Bicicleta" y no "Bicicletas". Hay que resistir la tentación de llamarla "Concepto de Bicicleta" (pensando que es sólo el concepto de bicicleta y

no la bicicleta). Cuando se nombran clases y propiedades, se debe pensar en cómo se van a utilizar. Suena mejor decir que "b1 es de tipo Bicicleta" que "r1 es de tipo Bicicletas" y que "r1 es de tipo Concepto de Bicicleta".

- Como última opción, especificar la correspondencia entre las ontologías. A veces la correspondencia entre ontologías se tiene que hacer cuando las ontologías se han desarrollado de forma independiente. Lo mejor es que la correspondencia se evite, ya que la construcción del conocimiento se hace mucho más complicado porque hay múltiples maneras de decir lo mismo.

### ***Clases y Conceptos***

Es tentador llamar a las clases conceptos, porque los símbolos representan conceptos: mapeo de la representación interna con los objetos o relaciones que los símbolos representan.

*Por ejemplo, puede ser tentador para llamar a la clase de los unicornios "Concepto de unicornio" ya que no existen los unicornios sino sólo el concepto de unicornio. Sin embargo, unicornios y el concepto de unicornios son muy diferentes; uno es un animal y el otro es una subclase de los conocimientos. Un unicornio tiene cuatro patas y un cuerno que sale de su cabeza. El concepto de un unicornio no tiene ni patas ni cuerno. Sorprendería mucho si un unicornio apareciera en una clase sobre ontologías, pero no debería sorprendernos que el concepto de un unicornio apareciera. No hay instancias de unicornios, pero hay muchas instancias de concepto de un unicornio. Si alguien se refiere al concepto de unicornio, se debe utilizar "concepto de unicornio". Si alguien se refiere a un unicornio, se debe utilizar "unicornio". No se debe decir que el concepto de unicornio tiene cuatro patas, porque las instancias de los conocimientos no tienen patas, sólo los animales (y los muebles) tienen patas (Poole, 2010).*

Llamar a los objetos conceptos es un error común en la construcción de ontologías. Aunque existe libertad de llamar a las cosas por cualquier nombre, sólo es útil para el intercambio de conocimientos si otras personas adoptan la ontología. Y no la van a adoptar si carece de sentido para ellos.

Las ontologías se pueden especificar en diferentes lenguajes a distintos niveles de abstracción. Para ello, existen diversos editores para implementar las ontologías. Un editor de ontologías debe proporcionar lo siguiente (Poole, 2010):

- Debe proporcionar una forma en que las personas introduzcan ontologías en el nivel de abstracción que tenga más sentido.

- Dado un concepto que un usuario quiera usar, un editor de ontologías debe facilitar la búsqueda de la terminología para ese concepto o determinar que no existe un término correspondiente.
- Debe ser sencillo para alguien determinar el significado de un término.
- Debe ser lo más fácil posible comprobar si la ontología es correcta (es decir, que se ajuste a la interpretación de los términos del usuario).
- Debe crear una ontología que otros pueden utilizar. Esto significa que se debe utilizar un lenguaje normalizado siempre que sea posible.

### ***Ontologías de alto nivel***

En informática, una ontología de alto nivel, es una ontología que describe conceptos muy generales que son los mismos en cualquier dominio del conocimiento. La función más importante de una ontología de alto nivel es la de facilitar la interoperabilidad semántica entre un gran número de ontologías accesibles “bajo” dicha ontología de alto nivel [15]. Normalmente, es una jerarquía de entidades y reglas asociadas que tratan de describir entidades generales que no pertenecen a un dominio de problema específico.

La construcción de ontologías de alto nivel genera grandes controversias en cuanto a su viabilidad y su construcción. Esto implica que no existen unos estándares aceptados mayoritariamente y que no exista la posibilidad de probar distintas ontologías de alto nivel en el mismo entorno computacional.

Dejando aparte dichos debates, se puede decir que una parte muy importante de cada ontología de alto nivel puede ser considerada como la implementación computacional de la filosofía natural, que a su vez es un método más empírico de investigación de los temas dentro de la disciplina filosófica de la ontología física.

El ejemplo que hablaba sobre apartamentos y edificios define una ontología de dominio diseñada para ser utilizada por personas que quieran escribir una base de conocimiento que se refiera a edificios de apartamentos. Cada ontología de dominio, implícita o explícitamente asume una ontología de alto nivel en la que pueda caber. Existe un interés en la construcción de una ontología coherente de alto nivel a la que otras ontologías puedan referirse y en la que puedan encajar. Encajar las ontologías de dominio en una ontología de alto nivel facilitaría la interoperación entre ellas.

Veamos a continuación los argumentos que distintos autores exponen a favor y en contra de la viabilidad de las ontologías de alto nivel y que vienen recogidos en la versión inglesa de Wikipedia [16].

### ***Argumentos en contra de la viabilidad de ontologías de alto nivel***

Desde siempre, se han producido intentos para establecer una serie de conceptos como primarios o fundamentales.

Hoy en día, los ordenadores están por todos los lados. Debido a la diversidad de puntos de investigación y a la libertad existente, existe un gran número de ontologías para las mismas áreas con diferencias realmente pequeñas. Sin embargo, a lo largo de la historia ya se ha visto que pequeñas diferencias ideológicas han dado lugar incluso a conflictos bélicos. La civilización moderna, obviamente, no permitiría en su sano juicio un conflicto armado con el trasfondo de las ontologías, sin embargo, muchas iniciativas privadas de establecer una ontología de alto nivel en pro de la mejora en la intercomunicación y la interoperabilidad no prosperan ni se asientan en el tiempo debido a la diversidad existente.

Otra gente, que se apoya en principios filosóficos, sostiene que no es objetivo establecer una ontología de propósito general y por lo tanto, sostienen que no es factible obtener un estándar donde encajen el resto de ontologías.

Otra de las objeciones es que, cuanto más generales sean los conceptos y más útil sea la interoperabilidad semántica buscada, menos probable es reducir los conceptos a símbolos y que sean aceptados por todo el mundo. No hay una forma de dividir el mundo en conceptos sin controversia y generalmente, no existe un terreno neutral donde encajar y traducir los conceptos que aparecen en las ontologías de bajo nivel.

### ***Argumentos a favor de la viabilidad de ontologías de alto nivel***

Muchos de los que dudan de la viabilidad de construir una ontología de alto nivel caen en alguna de las siguientes trampas:

- Afirman que no hay ninguna posibilidad de un acuerdo universal sobre cualquier esquema conceptual, pero ignoran que el hecho de una ontología común en la práctica no necesita ser universalmente aceptada. Basta con que un número suficientemente grande de comunidades de usuarios la utilice para que sea rentable utilizarla como un medio general de interoperabilidad y para que terceros desarrollen utilidades que faciliten su uso.
- Señalan que los desarrolladores de esquemas de datos encuentran diferentes representaciones afines a sus propios propósitos, pero no demuestran que estas diferentes representaciones son en realidad inconsistentes. Las afirmaciones acerca del mundo deben ser

consistentes y si existen dos representaciones sobre un elemento existente que no son consistentes, significa que una de ellas es errónea.

Las ontologías son perfectamente capaces de representar aproximaciones, y también situaciones en las que distintas aproximaciones tienen utilidad distinta. La objeción de que existen modelos del mundo incompatibles es cierta. Sin embargo, en una ontología de alto nivel, esos modelos pueden ser representados como distintas teorías que podrán ser usadas por los partidarios de las mismas mientras se respete la consistencia lógica de los supuestos necesarios de la ontología de alto nivel. Dichos supuestos incluyen el vocabulario necesario con el que especificar esos modelos incompatibles, los cuales, no está demostrado que no se puedan especificar con un conjunto común de conceptos básicos.

Los objetores de las ontologías de alto nivel tampoco tienen en cuenta la posibilidad de llegar a un acuerdo sobre una ontología que contenga aquellos elementos primitivos que puedan ser combinados para representar cualquier número de conceptos más especializados. Con esa táctica, se permite focalizar el esfuerzo en acordar número limitado de elementos ontológicos básicos.

Muchos de los argumentos en contra de las ontologías de alto nivel son cuestiones sobre definición, lenguaje y hechos. Por ejemplo, la gente utiliza diferentes términos para un mismo concepto. Sin embargo, eso no significa necesariamente que se refieran a conceptos distintos. Puede que esté usando simplemente otro idioma. Las ontologías típicamente utilizan etiquetas para referirse a los conceptos pero el concepto definido bajo esa etiqueta no significa ni más ni menos que lo que dictan sus axiomas. Se debe señalar que en las ontologías de alto nivel, el significado de sus elementos (clase, relaciones, reglas) depende únicamente de su forma lógica y no de sus etiquetas que son utilizadas para simplificar su manejo por parte de los humanos.

Existen diferentes palabras que etiquetan un único concepto, incluso en el habla (sinónimos), y puede que haya una sola palabra que etiquete múltiples conceptos (ambigüedad). Crear un mapeo entre el lenguaje humano y los elementos de la ontología es competencia de la comprensión del lenguaje natural. Sin embargo, la ontología funciona independientemente como una estructura lógica y computacional. Por esta razón, buscar un acuerdo en la estructura de una ontología es actualmente más fácil que desarrollar un vocabulario ya que todas las diferentes interpretaciones de una palabra pueden estar incluidas.

La principal barrera para construir este tipo de ontologías no es de cuestiones técnicas sino la reticencia de las agencias de financiación para aportar fondos para la creación de un consorcio suficientemente grande de desarrolladores y usuarios. Definitivamente, bien sean de alto nivel o no, las ontologías parecen tener, sin duda, un futuro muy prometedor donde proliferar.



### *Ejemplos de ontologías de alto nivel*

A lo largo de los años, ha habido gran cantidad de iniciativas y propuestas de desarrollo de ontologías de alto nivel. En este pequeño apartado, se puede observar un muestrario elaborado por Marcardi, Cordì y Rosso [15] de las mismas junto con una descripción de qué son y cómo y dónde funcionan.

- **BFO (Basic Formal Ontology):** consiste en dos tipos de ontologías (SNAP y SPAN) e incorpora las perspectivas de tres y cuatro dimensiones (incluye el tiempo) de la realidad en un único marco de trabajo. Las interrelaciones son definidas por los dos tipos de ontologías de tal forma que la BFO facilita que se traten las características de la realidad estático-espacial y dinámico-temporal. Las ontologías SNAP son un inventario de las entidades existentes en un punto en el tiempo mientras que las ontologías SPAN son un inventario de los procesos desarrollados en un intervalo de tiempo. Ambos tipos proporcionan la base para una serie de sub-ontologías que definen porciones de la realidad a diferentes niveles de granularidad.
- **Cyc:** es una ontología muy conocida y comprensible hoy en día. Es un sistema propietario en desarrollo desde 1984 que consiste en una ontología base y muchas ontologías de dominio específicas llamadas microteorías. Es una representación formal de los hechos, las reglas y las heurísticas para razonar acerca de los objetos de la vida cotidiana.
- **DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering):** es el primer modulo de la Biblioteca de Ontologías Base de WonderWeb (Proyecto para conseguir una infraestructura ontológica para la web semántica). Tiene un claro sesgo cognitivo ya que su objetivo es capturar las categorías subyacentes del lenguaje natural y del sentido común humano. Las categorías que representa son artefactos cognitivos que dependen de la percepción humana y de las impresiones y convicciones sociales, por lo que no está comprometido con una metafísica relacionada con la naturaleza intrínseca del mundo. DOLCE trata de ser sólo descriptiva. Es una ontología de particulares, que significa que es una ontología de instancias más que de propiedades universales. Posteriormente apareció Dns que lleva la postura descriptiva aún más lejos, eliminando las restricciones sobre los tipos de las entidades y sus relaciones.
- **GFO (General Formal Ontology):** es una ontología realista que integra procesos y objetos junto al tiempo, el espacio, las propiedades, las

relaciones, los roles, las funciones, los hechos y las situaciones. Intenta incluir muchos aspectos de la filosofía reciente, la cual se ve reflejada en su taxonomía de árbol y su axiomatización. Las dos especialidades de GFO son su persistencia y su modelo de tiempo entendiendo persistencia como la distinción explícita entre objetos y procesos mediante una categoría especial. Con respecto al tiempo, los intervalos de tiempo son primitivos mientras que los instantes o puntos de tiempo son derivados.

- **PROTON (PROTo Ontology):** es una ontología básica de alto nivel que da cabida a los conceptos generales necesarios para una amplia gama de tareas, incluyendo anotación semántica, indexación y recuperación de documentos. Los principios de diseño pueden ser resumidos en independencia del dominio, poco peso de las definiciones lógicas, alineación con estándares populares y buena cobertura de las entidades nombradas y dominios concretos.
- **Sowa's Ontology:** Las categorías básicas y las distinciones son derivaciones de gran variedad de fuentes de lógica, lingüística, filosofía e inteligencia artificial. Esta ontología no se basa en una jerarquía de categorías sino en un marco de trabajo de distinciones desde el cual se genera automáticamente la jerarquía. Para cualquier aplicación particular, las categorías no son definidas dibujando líneas en un gráfico sino seleccionando un conjunto de distinciones apropiado. Estas categorías incluyen objetos, procesos, esquemas, scripts, coyunturas, participación, descripción, historia, estructura, situación, razón y propósito. Cada una de las categorías puede ser física o abstracta (y en ambos casos puede ser o continua u ocurrente) y también puede ser independientes, relativa o de mediación.
- **SUMO (Suggested Upper Merged Ontology):** es otro proyecto de ontología comprensible. SUMO y sus ontologías de dominio forman uno de las mayores ontologías formales públicas que existen en la actualidad. Son usadas en aplicaciones e investigación sobre búsqueda, lingüística y razonamiento. SUMO se extiende con numerosas ontologías de dominio y un completo conjunto de links a WordNet (base de datos o diccionario diseñado para la red semántica) y está disponible completamente gratis.

A continuación, se muestran tres tablas con diferentes datos, informaciones y características interesantes de las ontologías presentadas con anterioridad.

	<b>Página Web</b>	<b>Desarrolladores</b>
<b>BFO</b>	<a href="http://www.ifomis.org/bfo">www.ifomis.org/bfo</a>	Smith, Grenon, Stenzhorn, Spearv (IFOMIS)
<b>Cyc</b>	<a href="http://www.cyc.com/">www.cyc.com/</a>	Cycorp
<b>DOLCE</b>	<a href="http://www.loa-cnr.it/DOLCE.html">www.loa-cnr.it/DOLCE.html</a>	Guarino and other researchers of the LOA
<b>GFO</b>	<a href="http://www.onto-med.de/ontologies/gfo.html">www.onto-med.de/ontologies/gfo.html</a>	The Onto-Med Research Group
<b>PROTON</b>	<a href="http://proton.semanticweb.org/">proton.semanticweb.org/</a>	Ontotext Lab, Sirma
<b>Sowa's</b>	<a href="http://www.jfsowa.com/ontology/">www.jfsowa.com/ontology/</a>	Sowa
<b>SUMO</b>	<a href="http://www.ontologyportal.org/">www.ontologyportal.org/</a>	Niles, Pease, and Menzel

**Tabla 14. Ontologías Alto Nivel 1**

	<b>Dimensiones</b>	<b>Lenguajes</b>	<b>Modularidad</b>
<b>BFO</b>	36 clases relacionadas	OWL	Módulos SNAP y SPAN
<b>Cyc</b>	Sobre 300.000 conceptos, 3.000.000 de aserciones (hechos y reglas) y 15.000 relaciones *	OWL y CycL	Microteorías
<b>DOLCE</b>	Unos 100 conceptos y 100 axiomas	LPO, KIF y OWL	No dividido en módulos
<b>GFO</b>	79 clases, 97 relaciones de subclase y 67 propiedades	LPO, KIF y OWL	Nivel superior abstracto, nivel central abstracto y nivel básico

<b>PROTON</b>	300 conceptos y 100 propiedades	OWL Lite	3 niveles incluyendo 4 modelos
<b>Sowa's</b>	30 clases, 5 relaciones y 30 axiomas	Lenguaje modal de primer orden y KIF	No dividido en módulos
<b>SUMO</b>	20.000 términos y 60.000 axiomas**	SUO-KIF y OWL	Dividido en SUMO, MILO y ontologías de dominio

Tabla 15. Ontologías Alto Nivel 2

\*Incluidas las “microteorías”.

\*\*Incluidas las ontologías de dominio.

	<b>Aplicaciones</b>	<b>Alineación con WordNet</b>	<b>Licencia</b>
<b>BFO</b>	Principalmente en el dominio biomédico	No soportado	Disponibilidad gratuita
<b>Cyc</b>	Procesamiento de lenguaje natural y evaluación de riesgos en la red.	Mapeado con alrededor de 12.000 Wordnet synsets	Producto comercial; ResearchCyc y OpenCyc están disponibles gratuitamente (ResearchCyc sólo para investigación)
<b>DOLCE</b>	Recuperación de información multilinguaje, servicios y sistemas basados en web, e-learning	DOLCE-Lite-Plus ha ido alineado con alrededor de 100 Wordnet synsets	Disponibilidad gratuita
<b>GFO</b>	Principalmente en el dominio biomédico	No soportado	Publicado bajo la licencia BSD modificada.
<b>PROTON</b>	Anotación semántica, sistemas de gestión de conocimiento y dominios de telecomunicación, análisis e investigación multimedia, investigación inteligente y Ontología de Datos de Negocio para servicios de la web semántica	No soportado	Disponibilidad gratuita
<b>Sowa's</b>	Inspiración para múltiples ontologías de alto nivel	No soportado	Disponibilidad gratuita
<b>SUMO</b>	Lingüística, representación y razonamiento	Mapeado a mano con Wordnet V2.1	Disponibilidad gratuita

Tabla 16. Ontologías Alto Nivel 3

## La Web semántica

Actualmente, la Web es un espacio preparado para el intercambio de información diseñado para el consumo humano. Las páginas web son creadas por personas para ser entendidas por personas. No existe un formato común para mostrar la información. Por ello, los diseñadores crean las páginas dependiendo de los potenciales usuarios que van a visitarlas. Los actuales buscadores web realizan la búsqueda de información mediante palabras clave que aparecen en el código HTML de las páginas web.

Otra carencia de la situación actual es que, con los estándares web del momento, no se puede diferenciar entre información personal, información académica, información comercial, etc. Es decir, cuando un buscador web realiza una consulta con algunas palabras clave, lo más común es que aparezca información que no es útil ya que no se corresponde con lo que se busca.

Por otro lado, los agentes de búsqueda actuales no se diseñan para "comprender" la información que reside en la web, precisamente porque es prácticamente imposible conocer la representación de los datos ubicados en las diferentes páginas [17].

En los últimos años, muchos investigadores han diseñado modelos para transformar la red desde un espacio de información hasta un espacio de conocimientos. En 1999, Tim Berners-Lee, uno de los inventores de la web, defendió el desarrollo de la web con conocimiento, y organizaciones como *Semantic Web* se encargan de estandarizar lenguajes y herramientas para hacer efectiva la Web Semántica.

La web semántica es una forma de permitir que el conocimiento interpretable por una máquina sea distribuido en el World Wide Web. En vez de simplemente servir páginas HTML para que sean leídas por humanos, los sitios web también proporcionarán información que puede ser usada por computadoras [3].

La web semántica se basa en dos conceptos fundamentales [18]:

- La descripción del significado que tienen los contenidos en la web.
- La manipulación automática de estos significados.

La idea es que los datos puedan ser utilizados y comprendidos por los ordenadores sin necesidad de supervisión humana, de forma que los agentes web puedan ser diseñados para tratar la información emplazada en las páginas web de manera semiautomática. Se trata de convertir la información en conocimiento, referenciando los datos de las páginas web a metadatos con un esquema común consensuado. Los metadatos no sólo especificarán el esquema de los datos que deben aparecer en cada instancia, sino que además podrán

tener información adicional de cómo hacer deducciones con ellos, es decir, axiomas que podrán aplicarse en los diferentes dominios que trate el conocimiento almacenado. El objetivo es mejorar Internet ampliando la interoperabilidad entre los sistemas informáticos usando agentes inteligentes.

Con ello, se mejorará la búsqueda de información y se potenciará el desarrollo de aplicaciones de comercio electrónico ya que las anotaciones de información seguirán un esquema común al igual que los buscadores web pudiendo intercambiar datos siguiendo esos esquemas comunes consensuados. Los agentes web no sólo encontrarán la información de forma precisa, sino que podrán realizar inferencias automáticamente buscando información relacionada con la que encuentran situada en las páginas, y con los requisitos de la consulta planteada por el usuario [19].

Para que esto pueda llevarse a cabo, se necesita que el conocimiento de la web esté representado de una forma consensuada, que sea legible por los ordenadores y sea reutilizable. Las ontologías proporcionan la vía para representar este conocimiento a través de conceptos, relaciones, funciones, instancias y axiomas

Para poder explotar la web semántica, se necesitan lenguajes de marcado apropiados que representen el conocimiento de las ontologías. Por otra parte, se necesitan agentes y aplicaciones web que exploten este conocimiento. Estos agentes serán capaces de interpretar los esquemas ontológicos y axiomas de diferentes dominios, mantendrán la consistencia de las instancias que se inserten en las páginas web siguiendo los esquemas ontológicos definidos, realizarán una búsqueda con inferencias utilizando los axiomas situados en los esquemas, y podrán realizar ligaduras entre los árboles taxonómicos de distintas ontologías. Para potenciar el uso de ontologías en la web, se necesitan aplicaciones específicas de búsqueda de ontologías que indiquen a los usuarios las ontologías existentes y sus características para utilizarlas en su sistema [1].

## ***Implantación***

Este apartado es el resultado de una adaptación de una parte del libro de Poole y Mackworth [3].

En el nivel más básico, XML (Extensible Markup Language) proporciona una sintaxis diseñada para que sea comprensible por las máquinas, pero que también sea entendible por humanos. Es un lenguaje basado en texto, donde los ítems son etiquetados de una manera jerárquica. La sintaxis para XML puede ser un tanto complicada, pero en el más simple de los niveles, el alcance de una etiqueta o está en la forma <tag.../>, o en la forma <tag...>...</tag>.

Un URI (un identificador de recursos uniformes) es usado únicamente para identificar un recurso. Un recurso es algo que puede ser identificado únicamente. Un URI es un string que se refiere a un recurso, como una página web, una persona, o una corporación. Frecuentemente, los URIs usan la sintaxis de las direcciones web.

Actualmente, mediante anotaciones RDF - RDF Schema (Resource Description Framework) se pueden representar algunas facetas sobre conceptos de un dominio y permite, mediante relaciones taxonómicas, crear una jerarquía de conceptos. Pero se necesitan lenguajes de marcado (basados en RDF) con mayor expresividad y capacidad de razonamiento para representar los conocimientos que contienen las ontologías. De esta forma, existen ya herramientas como Protégé, OntoEdit o WebOnto para realizar anotaciones en páginas web con los lenguajes de marcado propios.

**RDF** (Resource Description Framework) es un lenguaje construido en XML, proporcionando tripletas individuo-propiedad-valor.

**RDF-S** (RDF Schema) permite definir recursos (y también las propiedades) en términos de otros recursos (por ejemplo, usando subClassOf). RDF-S también permite restringir el dominio y el rango de propiedades y proporciona contenedores (conjuntos, secuencias y alternativas, una de las cuáles debe ser cierta).

RDF permite materializar sentencias en su propio lenguaje. Esto significa que se pueden representar fórmulas lógicas arbitrarias y por eso no es decidible en general. La indecidibilidad no es necesariamente algo malo; sólo significa que no se puede poner un límite al tiempo que una computación puede llevar. Los programas de lógica simple con símbolos de función y prácticamente todos los lenguajes de programación son indecidibles.

Sin embargo, se necesitan lenguajes de marcado (basados en RDF) con mayor expresividad y capacidad de razonamiento como DAML, DAML+OIL y más actualmente OWL para representar los conocimientos que contienen las ontologías.

**OWL** (Web Ontology Language) es un lenguaje de ontologías para el World Wide Web. Define algunas clases, propiedades e individuos. Tiene mecanismos de construcción para individuos, clases y propiedades, para restringir los dominios y los rangos de las propiedades, y otras restricciones sobre las propiedades (transitividad, cardinalidad, etc.).

Se han hecho algunos esfuerzos para construir grandes ontologías universales, pero la idea de web semántica es permitir a las comunidades converger en las ontologías. Cualquiera puede construir una ontología. La gente que quiere desarrollar una base de conocimiento, puede usar una ontología existente o desarrollarla por su cuenta, por lo general basada en ontologías existentes. Debido al interés de tener interoperabilidad semántica, las compañías



y los individuos deberían tender a converger en ontologías estándar para sus dominios o desarrollar mapeos desde sus ontologías a las ontologías de los demás.

### ***Dificultades***

Según Wikipedia [20], el desarrollo y difusión masivos de la web semántica tiene algunas dificultades que no ha podido superar todavía: una de ellas es tecnológica y la otra está relacionada con la falta de interés de los propietarios de las páginas web.

Las tecnologías para expresar la información en el formato que requiere una web semántica existen desde hace años. Quizás el componente más especializado sea OWL, que existe como estándar del W3C desde 2004. Sin embargo, falta algún componente que permita convertir de forma automática el abundante contenido de texto de las páginas web en marcas OWL. Actualmente se requiere que los creadores de las páginas web traduzcan "a mano" su contenido para que pueda ser interpretado por agentes semánticos. Afortunadamente algunas de las páginas tienen la información ordenada y clasificada en su base de datos y pueden realizar la conversión de manera automática.

La otra barrera que se opone pasivamente a la web semántica es el modelo de negocio de muchas páginas web, que obtienen ingresos de la publicidad. Estos ingresos son posibles únicamente si sus páginas son visitadas por mucha gente, y se perdería si los datos quedan disponibles para que los interprete un proceso automático.

La visión de los dueños de páginas web no debería ser esa ya que es muy probable que, de proliferar masivamente la web semántica, el modelo de negocios se adaptaría. Sin embargo, mientras no haya indicios de que esto vaya a suceder de manera inevitable e inminente, es lógico que los sitios web basados en publicidad no apoyen su desarrollo poniendo la información en el formato necesario.



## 4. Protégé y OWL

Este TFG se centra en elaborar un curso virtual para la enseñanza del lenguaje de ontología OWL, orientado a la web semántica, y del editor de ontologías Protégé, que facilita la construcción de las mismas. A continuación se muestra de forma muy resumida qué es OWL y qué es Protégé, además de su relación y su la forma en la que participan en este proyecto.

### OWL

OWL (Ontology Web Language) es un lenguaje de definición de ontologías basado en RDF y RDF-S de tal forma que extiende a los dos anteriores. Está recomendado por la W3C para la definición de ontologías web. Se trata de un lenguaje que añade mayor expresividad a RDF y RDF-S con una semántica formal basada en lógica descriptiva.

Se puede decir que tanto RDF como RDF-S tienen algunas limitaciones inaceptables para un lenguaje que pretende ser la base de la web semántica. RDF únicamente permite representar predicados binarios, mientras que RDF-S se limita a representar jerarquías de clases y propiedades, pudiendo definir para estas últimas restricciones de dominio y rango. Además, RDF y RDF-S carecen de soporte para tipos de datos primitivos, así como del poder expresivo para representar axiomas puesto que no existe ni la implicación, ni la cardinalidad ni la cuantificación. A dichas carencias hay que añadir que no permite especificar condiciones necesarias ni suficientes para establecer la pertenencia a una clase. Queda claro por lo tanto que la expresividad de RDF y RDF-S está muy limitada para poder modelar ontologías completas, por lo tanto se hace necesario buscar un nuevo lenguaje que proporcione un grado mayor de expresividad.

Ese nuevo lenguaje puede ser OWL, que fue creado por el W3C y es actualizado día a día. Actualmente se trabaja con la segunda versión de este lenguaje (OWL 2). Las normas de sintaxis y el significado semántico del código están explicados de forma muy detallada en las diferentes publicaciones que el W3C publica periódicamente y que se pueden consultar en su web oficial. En el siguiente enlace se puede revisar el estado actual de OWL y se puede acceder a las publicaciones actualizadas: [http://www.w3.org/standards/techs/owl#w3c\\_all](http://www.w3.org/standards/techs/owl#w3c_all).

OWL está dividido a su vez en tres tipos o sub-lenguajes para permitir diferentes niveles de expresividad. Estos lenguajes son OWL-Lite, OWL-DL y OWL-Full, los cuales se describen brevemente a continuación.

- **OWL-Lite.** Soporta las necesidades de aquellos usuarios que solamente necesiten representar una clasificación jerárquica simple con restricciones básicas. Es el lenguaje menos expresivo de los tres pero es el más simple

de ser provisto de herramientas de soporte y su complejidad formal es menor.

- **OWL-DL.** Es para aquellos usuarios que necesitan la máxima expresividad compatible con la decidibilidad y manteniendo la completitud y eficacia computacional (se garantiza que todas las conclusiones son computables en un tiempo finito). Incluye todos los constructores de OWL pero sólo pueden ser usados bajo ciertas restricciones. Es llamado de esta forma ya que está basado en la lógica de descripción (DL – Description Logic).
- **OWL-Full.** Diseñado para aquellos usuarios que buscan la máxima expresividad junto con la libertad sintáctica de RDF pese a no tener garantías sobre la computación. Permite ampliar el significado del vocabulario predefinido y es poco probable que exista algún software de razonamiento que soporte un razonamiento completo para cada característica de OWL-Full.

## Protégé

Protégé es una plataforma de código abierto y gratuita desarrollada por la Universidad de Stanford en colaboración con la Universidad de Manchester, que ofrece un conjunto de herramientas para construir modelos de dominio y aplicaciones basadas en conocimiento con ontologías.

Una ontología describe los conceptos y relaciones que son importantes en un dominio particular, proporcionando un vocabulario para dicho dominio así como el significado de los términos usados en el vocabulario. El rango de una ontología va desde simples taxonomías y clasificaciones hasta teorías axiomatizadas completas. En los últimos años, las ontologías han sido adoptadas en muchas comunidades científicas y empresariales como una forma de compartir, reutilizar y procesar conocimiento de dominio.

Protégé ofrece dos formas principales de modelar ontologías:

- **El editor Protégé-Frames:** permite a los usuarios crear y poblar ontologías basadas en marcos de acuerdo con el OKBC (Open Knowledge Based Connectivity protocol). Con este modelo, la ontología está formada por un conjunto de clases organizadas en una jerarquía para representar los conceptos más destacados del dominio, un conjunto de slots asociados a las clases para definir sus propiedades y relaciones y un conjunto de instancias de esas clases.
- **El editor Protégé-OWL:** permite a los usuarios construir ontologías para la web semántica, en particular para el lenguaje definido por el W3C (World Wide Web Consortium), OWL. Una ontología OWL debe contener

descripciones de clases, de propiedades y de instancias. La semántica de OWL permite derivar las consecuencias lógicas de la ontología, es decir, hechos no presentes en ella de forma literal.

Actualmente, existe una versión beta de Protégé llamada **WebProtégé**, que es un editor de ontologías basado en web y cuyo objetivo es soportar el proceso de desarrollo de ontologías de forma colaborativa en un entorno web.

Existen gran cantidad de plugins desarrollados para esta herramienta. Dichos plugins ofrecen mejoras, extensiones, nuevas funcionalidades y nuevas posibilidades. Se pueden descargar e instalar fácilmente para utilizarlos cuando sea preciso.

Protégé es una herramienta gratuita y multiplataforma descargable desde: [http://protege.cim3.net/download/4.2/installanywhere/Web\\_Installers/](http://protege.cim3.net/download/4.2/installanywhere/Web_Installers/)

## Protege 4.2 beta (build 276)

Recommended installer for your platform...

[Download Installer \(54.3M\) for Windows \(includes Java VM\)](#)

Enable Java in your browser to install directly from the web.

Installer created with InstallAnywhere. Copyright © 2012 Flexera Software LLC. All Rights Reserved.

### Available Installers

	Platform	includes Java VM	without Java VM	Instructions
X	MacOSX		<a href="#">Download (22.7M)</a>	<a href="#">View</a>
>	 <b>Windows 64bit</b>	<a href="#">Download (46.2M)</a>	<a href="#">Download (23.7M)</a>	<a href="#">View</a>
>	 <b>Windows</b>	<a href="#">Download (54.3M)</a>	<a href="#">Download (23.7M)</a>	<a href="#">View</a>
	 Linux 64bit	<a href="#">Download (52.1M)</a>	<a href="#">Download (23.2M)</a>	<a href="#">View</a>
	 Linux	<a href="#">Download (59.3M)</a>	<a href="#">Download (23.2M)</a>	<a href="#">View</a>
	UNIX Any Unix Platform		<a href="#">Download (23.2M)</a>	<a href="#">View</a>
	 Solaris		<a href="#">Download (23.2M)</a>	<a href="#">View</a>
	 HP-UX	<a href="#">Download (157.7M)</a>	<a href="#">Download (23.2M)</a>	<a href="#">View</a>
	 AIX	<a href="#">Download (120.2M)</a>	<a href="#">Download (23.2M)</a>	<a href="#">View</a>
	 Other Java-enabled Platforms		<a href="#">Download (23.2M)</a>	<a href="#">View</a>

Imagen 6. Página descarga Protégé

En esa dirección también se pueden consultar también los manuales de instalación de Protégé.

Se puede consultar una guía para comenzar a trabajar con Protégé en la siguiente dirección: <http://protegewiki.stanford.edu/wiki/Protege4GettingStarted>

## Relación Protégé-OWL

Este TFG se centra en mostrar el editor Protégé-OWL que permite construir ontologías para la web semántica pero a un nivel mucho más alto que el que se tiene al trabajar con OWL directamente. Este editor utiliza el tipo OWL-DL que garantiza la máxima expresividad manteniendo la completitud computacional y la decidibilidad.

El curso creado en este trabajo explica detalladamente cómo crear una ontología completa con Protégé. Sin embargo, es necesario resumir los pasos básicos para crear una ontología utilizando el editor Protégé.

El proceso de construcción de una ontología es un proceso iterativo que conduce desde una sencilla ontología inicial hasta la ontología completa tras varias revisiones en las que se refina poco a poco. Sin embargo, existen metodologías que establecen una serie de pasos a seguir en la construcción de ontologías. Se detalla a continuación:

- Determinación del ámbito: El dominio que cubre la ontología creada.
- Enumeración de los términos y definición de las clases en Protégé (pestaña *Classes* de Protégé).
- Definición de las propiedades de las propiedades. En OWL existen diferentes tipos de propiedades: Propiedades de objeto, de valor de tipos de datos y de anotación (pestañas *Object Properties* y *Datatype Properties* de Protégé).
- Definición de las restricciones sobre las propiedades que describen las clases.
- Creación de las instancias, que son los individuos que pertenecen a las clases. Son los datos actuales de la base de conocimiento representada por la ontología creada (pestaña *Individuals* de Protégé).

Al finalizar este proceso, la ontología estaría disponible para ser utilizada por algún sistema de manejo de conocimientos.

## 5. Curso de Ontologías en Moodle

En este capítulo se explica la parte más pedagógica de este TFG, haciendo referencia a los objetivos que se pretenden alcanzar con los usuarios del curso, la metodología utilizada y los contenidos tratados.

### Objetivos generales

Se espera que los usuarios de este curso, tras completarlo, sean capaces de:

- Comprender el concepto de ontología.
- Desarrollar una ontología y definir la jerarquía de clases adecuada.
- Manejar los distintos tipos de propiedades: de objeto, de tipos de datos y de anotación.
- Utilizar las propiedades para crear las restricciones que se utilizan para describir las clases: restricciones existenciales, universales, de cardinalidad y de valor.
- Conocer el mecanismo de razonamiento que se lleva a cabo en OWL y ser capaz de aprovechar todas sus utilidades: clasificación automática de clases e individuos y comprobación de la consistencia.
- Conocer y aprovechar las facilidades que brinda Protégé a la hora de construir ontologías.

### Metodología

La primera cuestión metodológica que se va a tratar es la elección de la plataforma Moodle para llevar a cabo la implementación del curso de ontologías. Los requisitos impuestos para la elección de la herramienta fueron los siguientes:

- Debía permitir una asistencia presencial, semi-presencial y no presencial.
- Debía ser fácil de utilizar por el usuario.
- Debía ser sencillo para los usuarios comprobar las fechas de entrega de las tareas.
- Debía ser fácil de manejar por el profesor.
- Debía ser fácil el mostrar y ocultar la información que el profesor considerase oportuna en cada momento.
- Debía ser fácil para el profesor corregir y evaluar las tareas.

Muchos de los entornos virtuales de aprendizaje analizados cumplían estos requisitos pero finalmente nos decantamos por Moodle por ser la plataforma con

la que se trabaja más habitualmente en la Universidad de Valladolid y, más en concreto, en la Escuela Técnica de Ingeniería Informática.

El siguiente aspecto metodológico analizado fue la estructuración del curso. Con el motivo de no abrumar a los posibles alumnos del curso, se pensó en dividir los contenidos en diferentes puntos de tal forma que se pudieran abordar en diferentes etapas. La forma de trabajar con la plataforma Moodle es a través de bloques, lo cual se ajusta perfectamente a este requisito.

El curso debía ofrecer la posibilidad de ser utilizado y seguido de manera presencial, no presencial y semi-presencial. Además, debía ser capaz de atender a las personas que quisieran profundizar más en los contenidos. De esta forma, se decidió dividir cada uno de los bloques en cuatro secciones:

- **Material de trabajo:** Presenta el material teórico básico del bloque. Para poder realizar las diferentes actividades propuestas es imprescindible leer, estudiar y consultar este material y es compatible con una posible explicación teórica del profesor.
- **Lección:** Es un compendio de actividades que están diseñadas para adquirir la habilidad necesaria para llevar a la práctica los conceptos teóricos de cada bloque. Las lecciones están pensadas para ser realizadas en un periodo limitado de tiempo (una o dos horas según el nivel del alumno) y, en principio, pueden completarse tanto de forma presencial como no presencial. Se recomienda la forma presencial para que el profesor pueda resolver las dudas que surjan, aunque están diseñadas para que puedan ser completadas de manera autónoma por cualquier persona que haya estudiado los conceptos teóricos. Hay que tener en cuenta que en el momento en que se realiza la lección se tiene acceso al material de trabajo del bloque y a Internet.
- **Material complementario:** Proporciona una serie de enlaces para que los alumnos más interesados puedan ampliar el horizonte en cada uno de los bloques tratados.
- **Tarea:** Es una actividad que debe ser enviada a través de Moodle antes de una fecha determinada. De esta forma, se “obliga” a los alumnos a repasar los conceptos aprendidos en cada bloque y a llevarlos a la práctica por sí mismos. Los alumnos deberán entregar los ejercicios prácticos propuestos resueltos junto con una memoria donde justifiquen las decisiones tomadas y expliquen cómo han llevado a cabo la resolución de los ejercicios.

Todos estos bloques con contenidos debían estar relacionados de alguna forma. Por esta razón, y tras analizar la ontología “Pizza Ontology”, que es una ontología utilizada en un manual de Protégé creado por la Universidad de Manchester, se decidió que cada bloque estuviera dedicado a avanzar un poco en el desarrollo de dicha ontología, siendo ésta el hilo conductor del curso. De



esta forma, los alumnos encontrarían más atractivo y motivador el tener que realizar la tarea encargada.

## **Contenidos**

Una vez que se tenía claro que la forma de distribuir los contenidos sería a través de diferentes bloques y que la implementación se llevaría a cabo en la plataforma Moodle, se trató el tema de los contenidos a incluir en el curso.

Se decidió que, como se iba a utilizar la ontología “Pizza Ontology” como hilo conductor del curso, se dividiría la construcción de esta ontología en diferentes etapas. Para llevar a cabo cada una de esas etapas, es necesario controlar una serie de conocimientos teóricos sobre ontologías y prácticos sobre cómo se usa Protégé. De esta forma, se identificaron los contenidos requeridos para poder implementar cada fase de la ontología de la Pizza.

Cada una de las fases en las que se dividió la construcción de la ontología de la Pizza se transformó en cada uno de los bloques que conforman la ontología. Se tuvo especial cuidado con que los contenidos que debía cubrir el curso, se distribuyeran de forma uniforme a lo largo de los bloques en los que estaba dividido y se correspondieran con las actividades prácticas planteadas para cada una de las etapas en las que se dividió la construcción de la ontología de la Pizza.

Con estas premisas, se estructuró el curso en trece bloques. Los contenidos tratados en cada uno de ellos, así como los resultados de aprendizaje que se espera que alcancen los alumnos se detallan a continuación.

### ***Bloque 1. Introducción***

Este primer bloque está destinado a poner en contacto al alumno con el curso y con los contenidos que se van a abordar. Sirve para recordar las nociones básicas sobre las ontologías y para introducir el lenguaje OWL y la herramienta Protégé. Además, se explica cuál será la forma de trabajar a lo largo del curso.

#### ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Ontologías y su problemática.
- Definición de ontología.
- Aplicaciones y futuro de las ontologías.
- Introducción a OWL.
- Tipos de OWL.
- Introducción a Protégé

### ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:

- Conocer los conceptos básicos sobre ontologías.
- Conocer qué es OWL y sus tipos.
- Conocer qué es Protégé y sus aplicaciones para la construcción de ontologías.
- Crear una ontología nueva definiendo los aspectos iniciales más importantes de la misma (nombre, ubicación, lenguaje).
- Conocer la web oficial del W3C y cómo navegar por los documentos publicados en ella.

### ***Bloque 2. Clases en OWL***

En este segundo bloque se aborda el concepto de clase en OWL y cómo se pueden implementar en Protégé. Se construye una ontología sencilla en la que se practican los conceptos aprendidos y se asientan las bases de la ontología de la Pizza, que será el hilo conductor del curso.

### ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Introducción a los conceptos.
- Clases en OWL. Definición, construcción, axiomas, jerarquía.
- Definición de clases en Protégé.
  - Clases con identificador.
  - Jerarquía de clases.

### ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:

- Conocer qué relación existe entre los conceptos de un dominio y las clases de OWL.
- Conocer las formas que permite OWL para definir clases.
- Conocer cómo se organizan las clases dentro de la jerarquía.
- Conocer qué significa que dos clases sean disjuntas entre sí.
- Conocer la forma de proceder a la hora de seleccionar una nomenclatura para las clases de la ontología.
- Construir manualmente la jerarquía de clases (con super-clases y sub-clases) de una ontología en Protégé.

- Utilizar el asistente de Protégé para la definición de una jerarquía de clases, incorporando prefijos y sufijos al identificador de las mismas cuando sea necesario.
- Declarar clases como disjuntas en Protégé.

### ***Bloque 3. Propiedades. Propiedades de objeto***

En este bloque se introducen las propiedades en OWL, haciendo especial hincapié en las propiedades de objeto, que son las que relacionan dos individuos. Además, se explican detalladamente las características que se pueden definir en las propiedades.

#### ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Definición y tipos de propiedades.
- Creación de propiedades de objeto.
  - Definición de las propiedades
  - Características de las propiedades
- Creación y definición de clases en Protégé.

#### ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:

- Conocer la función de las propiedades y los tipos de propiedades que existen.
- Crear propiedades de objeto en Protégé, estructurando correctamente la jerarquía de propiedades.
- Conocer cómo se definen las propiedades de objeto y definir con Protégé el dominio, el rango y las propiedades inversas de las propiedades creadas.
- Conocer las características que se pueden aplicar a las propiedades y caracterizar correctamente con Protégé las propiedades creadas.

### ***Bloque 4. Propiedades. Definiendo clases (Restricciones existenciales)***

En este bloque se aborda la definición de las clases creadas en la ontología de una forma mucho más completa. Para ello, se crean clases anónimas que son súper-clases de las clases definidas y que dotan a éstas de características

concretas. Para definir esas clases anónimas se utilizan restricciones sobre las propiedades. Este bloque se centra en las restricciones existenciales.

### ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Clases anónimas.
- Restricciones sobre propiedades.
  - Tipos de restricciones. Cuantitativas, de cardinalidad y de valor.
  - Restricciones cuantitativas. Existenciales y universales.
- Creación de restricciones existenciales en Protégé.

### ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:

- Conocer qué son las clases anónimas.
- Conocer las restricciones de propiedades y distinguir entre sus diferentes tipos.
- Conocer detalladamente las restricciones cuantitativas, distinguiendo entre restricciones existenciales y universales.
- Crear restricciones existenciales en Protégé que actúen como súper-clases anónimas para dotar de más significado a las clases de la ontología.

## ***Bloque 5. Condiciones necesarias y suficientes***

En este bloque se profundiza un poco más en la definición de clases. Se verá que, en función de dónde se describan las restricciones, se convierten en condiciones necesarias o suficientes que los individuos deben cumplir para pertenecer a dicha clase.

### ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Condiciones necesarias y suficientes.
  - Súper-clases y Clases equivalentes.
  - Clases primitivas y clases definidas.
- Creación de condiciones necesarias y suficientes en Protégé.

### ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:

- Conocer las diferencias entre condiciones necesarias y condiciones necesarias y suficientes en la descripción de una clase.
- Conocer qué relación existe entre el tipo de condiciones utilizado y cómo se denomina la clase, clase primitiva o clase definida.
- Conocer dónde se deben colocar las restricciones para que actúen como condiciones necesarias o como condiciones necesarias y suficientes.
- Saber crear con Protégé condiciones necesarias y condiciones necesarias y suficientes para los individuos de una clase.
- Saber transformar en Protégé las condiciones necesarias de una clase en condiciones necesarias y suficientes y viceversa.

### ***Bloque 6. El razonador***

En este bloque se introduce el concepto de razonador en OWL. Se verá que Protégé ofrece a los usuarios diferentes herramientas que actúan como razonadores y se explicarán las principales utilidades y cómo usarlas en un razonador concreto, el FaCT++. Además, se aprende a crear individuos para las ontologías.

#### ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Definición y funciones de un razonador.
  - Clasificación y comprobación de la consistencia.
- Diferencias entre “Asserted hierarchy” y “Inferred Hierarchy”.
- Invocación del razonador en Protégé.
- Creación de individuos en Protégé.

### ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:

- Conocer qué es un razonador en OWL y cuáles son sus funciones principales.
- Conocer qué es la clasificación automática y las diferencias entre la jerarquía de clases introducida manualmente y la jerarquía inferida.

- Conocer qué significa que una clase sea inconsistente.
- Invocar el razonador en Protégé para aplicar los conocimientos sobre clasificación automática y comprobación de la consistencia de la ontología.
- Crear individuos en Protégé y utilizar el razonador para aplicar los conocimientos sobre clasificación automática y comprobación de la consistencia de la ontología.

### ***Bloque 7. Propiedades. Definiendo clases (Restricciones universales)***

En este bloque se aborda la definición de clases anónimas utilizando un nuevo tipo de restricción cuantitativa sobre las propiedades, las restricciones universales, en las que está centrado este bloque.

#### ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Restricciones universales.
  - Diferencias entre restricciones existenciales y universales.
  - Características y definición de las restricciones universales.
- Creación de restricciones universales en Protégé.

#### ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:

- Conocer qué son las restricciones universales y sus diferencias con las restricciones existenciales.
- Determinar los individuos que pertenecen a una súper-clase anónima descrita con una restricción universal de forma teórica.
- Crear restricciones universales con el editor Protégé.
- Distinguir en qué situaciones se debe utilizar una restricción existencial y en qué otras una restricción universal en un contexto real de construcción de una ontología.

### ***Bloque 8. El razonador (2ª parte)***

En este bloque, se retoma una de las funciones principales del razonador, la clasificación automática, para indagar más en profundidad en ella.

## ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Importancia de la clasificación automática en las ontologías construidas en OWL-DL
  - Riesgo de utilizar el dominio y rango de las propiedades como medio para clasificar automáticamente clases.
- Open World Reasoning. La forma de razonar en OWL-DL.
- Axiomas de cierre.
- Plugin OWL Viz

## ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:

- Valorar la importancia de la clasificación automática en el mantenimiento de grandes ontologías y el riesgo de utilizar otra cosa que no sea las restricciones sobre propiedades para efectuar esa clasificación.
- Conocer la forma de razonar que se lleva a cabo cuando se trabaja con OWL-DL para poder resolver.
- Resolver el origen de los problemas que puedan surgir cuando se construyen ontologías por no haber tenido en cuenta correctamente que se razona con OWR.
- Conocer la utilidad de los axiomas de cierre y saber aplicarlos en contextos reales de construcción de ontologías.
- Llevar a la práctica los puntos anteriores utilizando el editor Protégé.
- Utilizar el plugin OWL Viz de Protégé para visualizar la jerarquía de clases en una ontología.

## ***Bloque 9. Particiones de valor***

Este bloque muestra nuevas posibilidades para describir las clases de una ontología de tal forma que sea posible añadir más información sobre los individuos que pertenecen a las mismas. Se explican las particiones de valor, que son una forma de categorizar a los individuos en distintos grupos según una determinada característica.

## ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Particiones de valor
  - Definición.
  - Partición de valor como patrón de diseño.
- Creación de particiones de valor en OWL.
- Axiomas de recubrimiento.
  - Definición.
  - Creación con Protégé.

### ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:

- Conocer qué son las particiones de valor.
- Aplicar el patrón de diseño de las particiones de valor en ontologías creadas con OWL-DL.
- Conocer qué es un axioma de recubrimiento.
- Crear particiones de valor en Protégé utilizando axiomas de recubrimiento.
- Determinar la necesidad de utilizar una partición de valor en un contexto real de construcción de una ontología.

### ***Bloque 10. Propiedades. Definiendo clases (Restricciones de cardinalidad)***

En este bloque se explica un nuevo tipo de restricciones, las restricciones de cardinalidad, que, al igual que los otros tipos de restricciones, se utilizan para describir de forma mucho más específica las clases de la ontología a través de la creación de súper-clases anónimas.

### ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Definición de restricción de cardinalidad en OWL.
- Tipos de restricción de cardinalidad. Exacta, de máximo y de mínimo.
- Características de las restricciones de cardinalidad.
- Restricciones de cardinalidad cualificadas
- Creación de restricciones de cardinalidad en Protégé.

### ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:



- Conocer qué son las restricciones de cardinalidad y sus diferentes tipos, así como las restricciones de cardinalidad cualificadas.
- Comprender las características que tienen que cumplir las relaciones y los individuos envueltos en ellas para que sean contadas en una restricción de cardinalidad.
- Crear restricciones de cardinalidad de diferentes tipos y cualificadas utilizando el editor Protégé.
- Identificar en un contexto real la necesidad de utilizar los diferentes tipos de restricciones de cardinalidad.
- Anticipar si un individuo de la ontología cumple con las condiciones impuestas por la restricción de cardinalidad para pertenecer a la clase que se describe.

### ***Bloque 11. Propiedades. Datatype Properties.***

En este bloque se introduce un nuevo tipo de propiedades en Protégé-OWL así como todas las posibles características que se pueden definir en ellas. Estas propiedades son las datatype properties, que relacionan a individuos pertenecientes a alguna clase con un valor de un tipo de datos predefinido.

#### ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Definición y características de las Datatype Properties en OWL.
- Utilidad de las Datatype Properties.
- Creación de Datatype Properties en Protégé.
- Tipos de datos disponibles.

#### ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:

- Conocer qué son las Datatype Properties en OWL y las características que se pueden definir en ellas.
- Conocer las diferentes utilidades de las Datatype Properties.
- Identificar en qué casos se deben utilizar Datatype Properties e implementarlas con el editor Protégé.
- Conocer los tipos de datos que se pueden utilizar en las Datatype Properties y saber seleccionar cuál es el adecuado en un contexto real, evitando inconsistencias en la ontología construida.

## ***Bloque 12. Definiendo clases. Aspectos avanzados***

En este bloque se avanza en el conocimiento del OWR (Open World Reasoning), es decir, la forma de razonar en Protégé-OWL. Además, se explica un nuevo tipo de restricciones, las restricciones de valor, que aumentan las posibilidades del alumno a la hora de describir clases.

En este bloque no se realiza encargo práctico y en la lección se trabaja sobre la ontología de la Pizza, hilo conductor del curso. El motivo es permitir a los alumnos dedicar más tiempo a depurar esta ontología ya que en el siguiente bloque el curso finaliza y deberán entregar la ontología completa y documentada.

### ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Complementario de una clase utilizando OWR.
- Definición de restricción de valor.
- Características de las restricciones de valor.
  - Recomendaciones en su utilización.
- Construcción de restricciones de valor en Protégé.

### ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:

- Conocer qué es el complementario de una clase, identificar en un contexto real cuándo es necesario utilizarlo e implementarlo correctamente con el editor Protégé.
- Conocer qué son las restricciones de valor, identificar en un contexto real la necesidad de su utilización e implementarlas correctamente con el editor Protégé.
- Anticipar los individuos que pertenecen a las clases afectadas por restricciones de valor y a las que son el complementario de una clase ya descrita.

## ***Bloque 13. Otros aspectos***

En este bloque se tratan algunos aspectos menores sobre la construcción de ontologías OWL en Protégé como por ejemplo la definición de clases por enumeración y por operadores lógicos y las propiedades de anotación y se realiza un repaso completo de todos los tipos de restricciones vistas a lo largo del curso.

En este bloque no hay lección debido a la sencillez de algunos contenidos tratados y la naturaleza de repaso de otros. En el encargo práctico, se incluyen los nuevos contenidos y se solicita la entrega de la ontología de la Pizza completa.

### ***Contenidos***

Los contenidos que se tratarán en este bloque son:

- Definición de clases por enumeración.
- Definición de clases por operadores lógicos.
- Annotation Properties (Propiedades de anotación).
- Restricciones en OWL.
- Creación de clases por enumeración, por operadores lógicos y de propiedades de anotación utilizando el editor Protégé.

### ***Resultados del aprendizaje***

Al finalizar este bloque, se espera que el alumno sea capaz de:

- Conocer qué son las clases definidas por enumeración y construirlas con Protégé.
- Conocer los diferentes operadores lógicos que se utilizan en la descripción de clases y saber aplicarlos en un contexto real.
- Conocer las propiedades de anotación y utilizarlas en Protégé para complementar la descripción de las ontologías construidas.
- Conocer los diferentes tipos de restricciones existentes en OWL y saber identificar la necesidad de utilizar cada una de ellas en un contexto real de construcción de una ontología con el editor Protégé.

## Guía de usuario básica

Esta sección pretende orientar a los usuarios de este curso, bien sean profesores o alumnos, en el funcionamiento básico de la plataforma. Para complementar las nociones básicas que se presentarán, se mostrarán enlaces a manuales de la herramienta.

En primer lugar, para acceder al curso hay que dirigirse a la página web donde se encuentra alojado: <http://jair.lab.fi.uva.es/~cartrav/moodle/>

También se puede llegar a esa dirección a través del siguiente código QR.



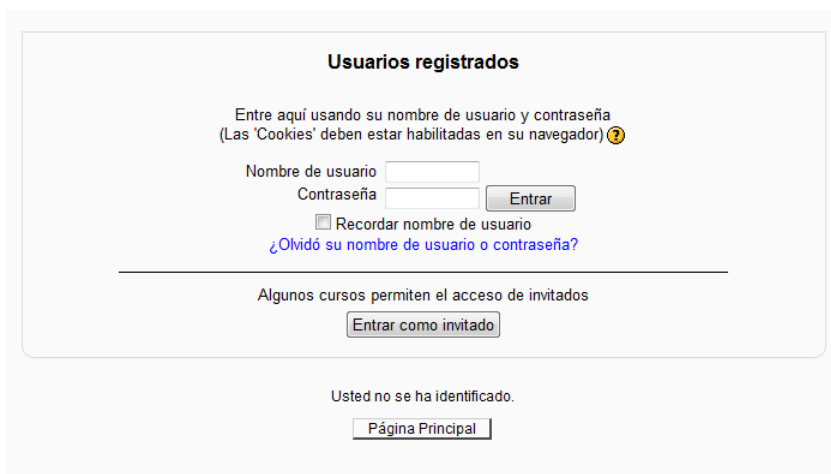
Imagen 7. Código QR del curso en Moodle

Una vez en la página se podrá ver una pantalla como la que se muestra en la siguiente imagen.



Imagen 8. Página principal del Moodle del curso

Para acceder al curso, basta con hacer un solo clic sobre el título del curso: Curso virtual para el desarrollo de ontologías en OWL con el editor Protégé. Y aparecerá una pantalla como la siguiente:



**Usuarios registrados**

Entre aquí usando su nombre de usuario y contraseña  
(Las 'Cookies' deben estar habilitadas en su navegador) ?

Nombre de usuario

Contraseña

☐ Recordar nombre de usuario

[¿Olvidó su nombre de usuario o contraseña?](#)

---

Algunos cursos permiten el acceso de invitados

Usted no se ha identificado.

**Imagen 9. Registro en el curso en Moodle**

En ese momento, es necesario introducir el Nombre de usuario y la contraseña en los campos destinados para ello y pulsar el botón Entrar. Si todo ha ido bien, se abrirá el curso y podrá ser utilizado de una forma u otra según el rol del usuario. La forma de utilizar el curso según dicho rol se explicará a continuación, pero antes se muestran algunos usuarios con rol de alumno creados para permitir que se pruebe el curso:

**Nombre de usuario:** presidente

**Contraseña:** Presidente\_1

**Nombre de usuario:** vocal1

**Contraseña:** Vocal\_01

**Nombre de usuario:** vocal2

**Contraseña:** Vocal\_02

### ***Guía profesor***

Una vez que se ha entrado en el curso con rol de profesor, se muestran todas las actividades y recursos creados, estén o no ocultos para el alumno. Para editar los contenidos, hay que pulsar el botón “Activar edición” de la parte superior derecha de la pantalla como se muestra en la siguiente imagen.

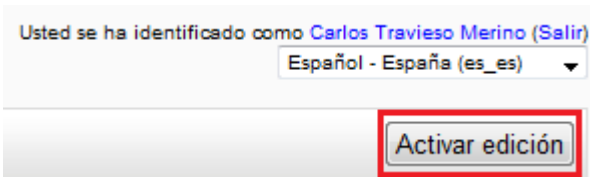


Imagen 10. Botón de activar edición en Moodle

Con la edición activada, aparecerán los siguientes iconos al lado de todos los recursos del curso:



Imagen 11. Opciones para cada recurso

Y en la siguiente tabla se explica básicamente el significado de cada uno de ellos:

Símbolo	Significado
	Desplazan el recurso a otra posición dentro del curso.
	Duplica el recurso.
	Elimina el recurso.
	Oculto o muestra el recurso al alumno e indica su estado al profesor: Si el ojo está abierto, está visible por los alumnos.
	Permite asignar los roles que pueden acceder a este recurso.
	Permite editar el recurso.

Imagen 12. Significado de los iconos

Además, con la edición activada, aparecen en cada bloque los dos menús que se muestran a continuación que permiten añadir nuevos recursos y actividades.



Imagen 13. Botones para agregar actividades o recursos

La plataforma Moodle permite agregar al curso recursos de múltiples tipos como se observa en la siguiente imagen.

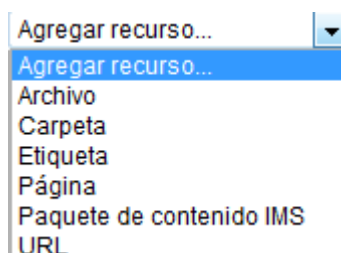


Imagen 14. Tipos de recursos

Lo mismo sucede con las actividades. Se permite añadir actividades con características muy distintas como se muestra a continuación.

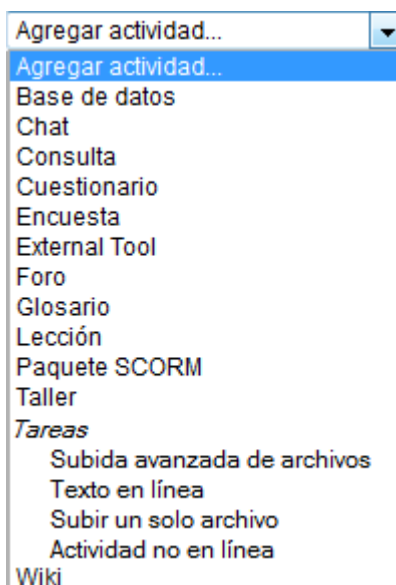


Imagen 15. Tipos de actividades

Para conocer más acerca de la forma de trabajar con Moodle, las diferentes posibilidades que ofrece de creación de recursos y actividades, la manera de editarlos y cómo se califican las actividades realizadas por los alumnos, se recomienda consultar los manuales publicados en la web oficial de Moodle: [http://docs.moodle.org/all/es/Manuales\\_de\\_Moodle](http://docs.moodle.org/all/es/Manuales_de_Moodle).

## Curso en Moodle de construcción de ontologías con Protégé

Se recomienda especialmente, por su completitud y rigor, el seguimiento de alguno de los manuales siguientes:

- [download.moodle.org/docs/es/1.9.4\\_usuario\\_profesor.pdf](http://download.moodle.org/docs/es/1.9.4_usuario_profesor.pdf)
- [recursos.cepindalo.es/mod/resource/view.php?id=13104](http://recursos.cepindalo.es/mod/resource/view.php?id=13104)

### Guía alumno

Cuando se accede al curso con rol de alumno, aparece la pantalla que se muestra en la siguiente imagen:

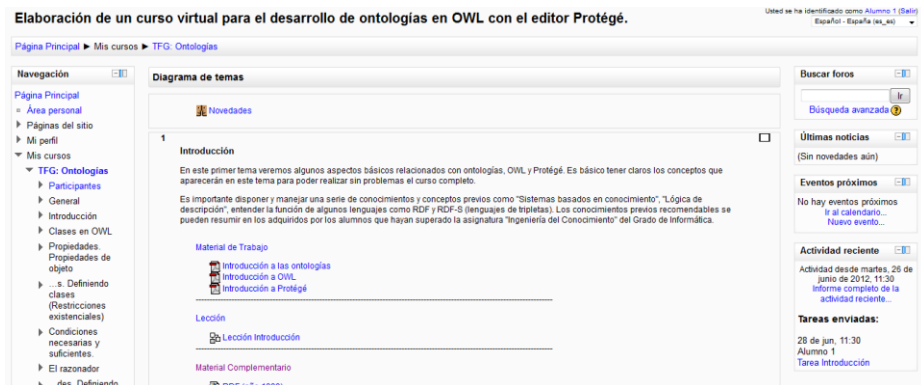






Imagen 16. Página principal del curso en Moodle

Esta pantalla es muy similar a la que se muestra a los profesores y aparecen los contenidos y actividades que el profesor responsable del curso establece como visibles en cada momento.

En este curso en concreto, se mostrarán básicamente cuatro tipos de recursos y actividades:

-  **Archivos PDF.** Contienen los materiales teóricos de cada bloque. Para acceder a ellos basta con hacer un clic sobre el icono o el nombre del archivo.
-  **Enlace URL.** Contienen enlaces a páginas web de interés para ampliar los contenidos de cada bloque. Para acceder basta con hacer un clic sobre el icono o el nombre del enlace.
-  **Lección.** Contiene una serie de ejercicios para ser realizados en un periodo concreto de tiempo con la finalidad de asentar y aplicar de forma práctica los contenidos de cada bloque. Para acceder basta con hacer un clic sobre el icono o el nombre de la lección. Posteriormente, el seguimiento de cada lección es muy intuitivo, mostrándose los ejercicios a realizar uno tras otro.



 **Tarea.** Contiene la descripción de una actividad práctica que debe ser entregada antes de una determinada fecha con la finalidad de que se practique de manera autónoma lo aprendido en cada lección, construyendo al mismo tiempo la ontología de la Pizza, hilo conductor del curso. Para acceder basta con hacer un clic sobre el icono o el nombre de la tarea. La entrega de los archivos solicitados es muy intuitiva una vez dentro de la tarea correspondiente.

Para conocer más acerca de la forma de trabajar con Moodle desde la perspectiva del alumno y para solucionar las dudas que puedan surgir, se recomienda consultar los manuales publicados en la web oficial de Moodle: [http://docs.moodle.org/all/es/Manuales\\_de\\_Moodle](http://docs.moodle.org/all/es/Manuales_de_Moodle).

Se recomienda especialmente, por su completitud y rigor, el seguimiento del siguiente manual:

- [download.moodle.org/docs/es/1.9.4\\_usuario\\_alumno.pdf](http://download.moodle.org/docs/es/1.9.4_usuario_alumno.pdf)



## 6. Conclusiones y líneas de trabajo futuras

Las conclusiones principales de este proyecto son las que se presentan a continuación.

En primer lugar, se ha cumplido el principal objetivo marcado, que es la creación de una herramienta de apoyo para la enseñanza de la construcción de ontologías en OWL utilizando el editor Protégé. Esta herramienta cumple finalmente todos los requisitos formales establecidos previamente.

Con respecto a la consecución de los objetivos pedagógicos planteados, desde mi perspectiva y desde la del tutor parece que el curso está encaminado correctamente a la adquisición de los mismos por parte de los alumnos que lo completan. Sin embargo, la fase de transición del software que permita evaluar y depurar el curso no se ha podido llevar a cabo.

Esto último, establece una línea de trabajo futuro que se antoja inmediata, el probar el curso con usuarios semi-reales que completen un informe que permita depurar los errores y modificar lo necesario hasta conseguir una herramienta de calidad. Una vez finalizada esa fase, sería el momento de llevar a cabo una labor institucional para poder organizar este curso en el marco del Centro BuenDía de la UVA, la UVA en curso, u otras vías dentro del contexto de los cursos de formación complementaria ofrecidos por la Universidad de Valladolid. Otra de las formas viables de utilización del curso sería con los alumnos del Grado de Informática de la UVA que cursen la asignatura Ingeniería del Conocimiento o con los alumnos de un posible Máster de Informática.

En cuanto a lo personal, con la realización de este trabajo, he conocido mucho más a fondo el tema de las ontologías, su importancia y sus aplicaciones, entre las que destaca la Web 3.0 o Web Semántica y que ha despertado mi interés en gran medida. Por otro lado, también he adquirido cierto grado de experiencia en el manejo de la herramienta de edición de ontologías en OWL Protégé, que era otro de los objetivos perseguidos.

Paralelamente y como consecuencia directa del desarrollo de este TFG he adquirido soltura en la administración de un sistema informático como es la plataforma de aprendizaje virtual Moodle, desde la instalación, puesta en marcha y configuración inicial, hasta la creación, gestión y mantenimiento de cuentas de usuario, cursos, etc.



## 7. Bibliografía, referencias y recomendaciones

En este capítulo se muestra la bibliografía consultada a la hora de desarrollar este TFG. También se exponen las páginas web consultadas para la elaboración del trabajo de búsqueda y procesado de información que aparece en esta memoria. Por último, también se presentan una serie de documentos interesantes que completan este Trabajo Fin de Grado por su relación directa con el mismo.

### Bibliografía y referencias web

A continuación, se presentan en orden de aparición todas las referencias bibliográficas y web utilizadas en este proyecto:

**[1]** Laura Grela, Elena Sauri, Alicia Sellés. Ontologías en documentación, 2002. URL <http://personales.upv.es/ccarrasc/doc/2001-2002/ontologias/INICIO.htm>

**[2]** Marek Obitko. Introduction to Ontologies and Semantic Web, 2007. URL <http://www.obitko.com/tutorials/ontologies-semantic-web/introduction.html>

**[3]** David Poole, Alan Mackworth. Artificial Intelligence. Foundations of computational agents. Cambridge University Press, 2010.

**[4]** Benjamin, Perakath C.; Menzel, Christopher P.; Mayer, Richard J.; Fillion, Florence; Futrell, Michael T.; deWitte, Paula S.; Lingineni, Madhavi. Information Integration for Concurrent Engineering (IICE), IDEF5 Method Report. September 21, 1994. URL <http://www.idef.com/pdf/Idef5.pdf>

**[5]** L. Liu and T. Özsu . Human–Computer Interaction. Entry in Encyclopedia of Database Systems. Springer, 2009.

**[6]** Thomas R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human-Computer Studies, Vol. 43, Issues 4-, pp. 907-928, November 1995.

**[7]** Thomas R. Gruber. A translation approach to portable ontology specifications, Knowledge acquisition, 5(2): 199-220, 1993.

**[8]** Herbert B. Enderton. A Mathematical Introduction to Logic. Academic Press, 1972.

**[9]** Wikipedia inglés. Ontology (information science). URL [http://en.wikipedia.org/wiki/Ontology\\_\(information\\_science\)](http://en.wikipedia.org/wiki/Ontology_(information_science))

**[10]** Eva García Hernández. Aplicación de las ontologías para la representación del conocimiento 7º Congreso de ISKO en Barcelona, pp. 224-237, 2005.

**[11]** Van Heist. Using explicit ontologies in KBS development. International Journal of Human-Computer studies Vol. 47 pp.183 – 292, 1997.

**[12]** Nicola Guarino. Formal Ontology and Information Systems. International Conference on Formal Ontology Systems, Trento, Italy, 6-8 June 1998. IOS Press, pp. 3-15, 1998. URL <http://www.loa.istc.cnr.it/Papers/FOIS98.pdf>

**[13]** Brian Cantwell Smith. On the Origin of Objects. MIT Press, Cambridge, MA, 1996.

**[14]** Frank van Harmelen, Vladimir Lifschitz, Bruce Porter. Handbook of Knowledge Representation. Elsevier, 2008.

**[15]** Viviana Mascardi, Valentina Cordi, Paolo Rosso. A Comparison of Upper Ontologies. DISI Technical Report, 2006. URL <http://www.disi.unige.it/person/MascardiV/Download/DISI-TR-06-21.pdf>

**[16]** Wikipedia inglés. Upper ontology (information science). URL [http://en.wikipedia.org/wiki/Upper\\_ontology\\_%28information\\_science%29](http://en.wikipedia.org/wiki/Upper_ontology_%28information_science%29)

**[17]** Adolfo Lozano Tello. Ontologías en la Web Semántica. I Jornadas de Ingeniería Web' 01 2001. URL [http://www.anobium.es/docs/gc\\_fichas/doc/68ERfhjkmv.pdf](http://www.anobium.es/docs/gc_fichas/doc/68ERfhjkmv.pdf)

**[18]** P. Castells. La Web Semántica. C. Bravo, M. A. Redondo (Eds.), Sistemas Interactivos y Colaborativos en la Web. Ediciones de la Universidad de Castilla - La Mancha, pp. 195-212, 2003.

**[19]** Jaime Blanco. Lecturas en Ciencias de la Computación. Notas sobre Web Semántica, 2007. URL <http://www.ciens.ucv.ve/escueladecomputacion/documentos/archivo/48>

**[20]** Wikipedia español. Web Semántica. URL [http://es.wikipedia.org/wiki/Web\\_sem%C3%A1ntica](http://es.wikipedia.org/wiki/Web_sem%C3%A1ntica)

**[21]** Mark Stefik. Introduction to Knowledge Systems. Morgan Kaufmann, 1995.

**[22]** J.T. Palma y R. Marín (Coordinadores). Inteligencia Artificial: técnicas, métodos y aplicaciones, McGrawHill, 2008.

**[23]** Stuart Russell, Peter Norvig. Inteligencia Artificial: Un Enfoque Moderno. 2ª Edición. Prentice Hall, 2004.

**[24]** Apuntes asignatura Ingeniería del Conocimiento. Curso de Adaptación al Grado de Informática. Universidad de Valladolid, 2010-2011.

## Recomendaciones

Antes de construir el curso, se llevó a cabo una labor de formación en la que se recopiló y procesó gran cantidad de información. De ese trabajo, surgieron una serie de documentos que se ponen a disposición del lector que esté interesado en profundizar en los temas que tratan. Se puede acceder a ellos a través de los siguientes enlaces:

**Agentes y el entorno. URL**

<http://jair.lab.fi.uva.es/~cartrav/AgentesYElEntorno.pdf>

**Sistemas basados en conocimiento. URL**

<http://jair.lab.fi.uva.es/~cartrav/SistemasBasadosEnConocimiento.pdf>

**Representación del conocimiento. URL**

<http://jair.lab.fi.uva.es/~cartrav/RepresentacionDelConocimiento.pdf>

Además, para completar la información con respecto a los lenguajes web de ontologías, se recomienda consultar el siguiente TFG:

Enrique Calleja Hoyos. TFG del Curso de Adaptación al Grado de Informática de la UVa (Universidad de Valladolid): Edición de ontologías con *Protégé*, Ejemplos básicos y *Gene Ontology*, 2011.





## Anexo I. Encuesta usuarios de prueba

En este anexo se presenta el diseño preliminar de la encuesta o informe que deberán rellenar los usuarios de prueba que completen el curso en la fase de transición.

Existen dos tipos de preguntas, las puntuables en una escala y las de respuesta escrita. En las preguntas que hay que responder con respecto a una escala, el número más alto se corresponde con “Excelente” mientras que el número más bajo se corresponde con “Muy mal”. Además, se proporcionará un espacio para justificar la respuesta si se desea.

Al comienzo del curso deberán determinar cuál es su nivel de conocimientos previo sobre el tema tratado y sobre la herramienta Protégé.

Por cada bloque, deberán responder a las siguientes cuestiones:

- Erratas encontradas
- Con respecto al material de trabajo:
  - Está bien explicado (de 1 a 5)
  - Se entiende lo explicado (de 1 a 5)
  - El material es completo (de 1 a 5)
  - Hay suficientes ejemplos (de 1 a 5)
  - Los ejemplos ayudan a entender la teoría (de 1 a 5)
  - La extensión es adecuada (de 1 a 5)
- Con respecto al material complementario:
  - Aporta información significativa (de 1 a 5)
  - Completa lo que ya sabía (de 1 a 5)
  - Se corresponde con lo que se explica en el bloque (de 1 a 5)
  - Hay suficiente material complementario (de 1 a 5)
  - El material complementario no es excesivo (de 1 a 5)
- Con respecto a la lección:
  - La extensión es adecuada (de 1 a 5)
  - La dificultad de las preguntas es adecuada (de 1 a 5)
  - Los ejercicios planteados son coherentes (de 1 a 5)
  - La lección es atractiva e interesante (de 1 a 5)
  - Ayuda a comprender los conceptos teóricos (de 1 a 5)
  - Se aprende a implementar en Protégé los conceptos teóricos del bloque (de 1 a 5)
- Con respecto a la tarea:
  - Se entiende lo que se pide (de 1 a 5)
  - La extensión es adecuada (de 1 a 5)

## Curso en Moodle de construcción de ontologías con Protégé

- La dificultad es adecuada (de 1 a 5)
- Es factible completar la tarea a tiempo (de 1 a 5)
- La tarea es atractiva e interesante (de 1 a 5)
- Es factible realizarla de manera autónoma con lo aprendido en el bloque (de 1 a 5)

A la conclusión del bloque deberán responder a los siguientes puntos:

- He aprendido los conceptos teóricos planteados (de 1 a 5)
- He adquirido las habilidades prácticas en Protégé para implementar los conceptos tratados en el bloque (de 1 a 5)
- El nivel de los contenidos ha sido (de 1 a 5)
- La asimilación de los contenidos ha sido (de 1 a 5)
- Aspectos a mejorar y sugerencias
- La lección ha sido atractiva y entretenida (de 1 a 5)
- Valoración general (nota de 1 a 10)

A la conclusión del curso deberán responder a los siguientes puntos:

- Nivel global de los contenidos del curso (de 1 a 5)
- Asimilación de los contenidos del curso (de 1 a 5)
- Aprendizaje de los conceptos teóricos (de 1 a 5)
- Adquisición de las habilidades prácticas (de 1 a 5)
- Eficacia del curso (de 1 a 5)
- Coherencia en la estructuración de los contenidos (de 1 a 5)
- Coherencia en la evolución de los contenidos (de 1 a 5)
- Coherencia en el hilo conductor del curso, la ontología de la Pizza (de 1 a 5)
- Interés del curso (de 1 a 5)
- Aspectos a mejorar y sugerencias (de 1 a 5)
- Valoración global (nota de 1 a 10)

## Anexo II. Contenido del CD

Este anexo detalla el contenido del CD que se adjunta con el presente documento. A continuación se muestra la jerarquía de carpetas o directorios (negrita) junto con los archivos que contienen.

memoria.pdf

### **Software**

#### **Ontologías por bloques**

##### **Bloque 1**

Primera\_Ontologia.owl

##### **Bloque 2**

Leccion\_Clases.owl

##### **Bloque 3**

Leccion\_Propiedades\_Objeto.owl

##### **Bloque 4**

Leccion\_Restricciones.owl

##### **Bloque 5**

Leccion\_Condiciones.owl

##### **Bloque 6**

Leccion\_Razonador.owl

Leccion\_Condiciones.owl

##### **Bloque 7**

Leccion\_Restricciones\_Universales.owl

Leccion\_Razonador.owl

Leccion\_Condiciones.owl

##### **Bloque 8**

Leccion\_Razonador\_2.owl

##### **Bloque 9**

Leccion\_Particiones\_Valor.owl

##### **Bloque 10**

Leccion\_Restricciones\_Cardinalidad.owl

Leccion\_Particiones\_Valor.owl

Leccion\_Razonador\_2.owl

##### **Bloque 11**

Leccion\_Datatype\_Properties.owl

Leccion\_Restricciones\_Cardinalidad.owl

Leccion\_Particiones\_Valor.owl

Leccion\_Razonador\_2.owl

Pizza\_Ontologia.owl

install\_protege\_4.2\_beta.exe

ProtegeOWLTutorialP4\_v1\_3.pdf

Guia\_Usuario.pdf

## Curso en Moodle de construcción de ontologías con Protégé

En las siguientes líneas, se detalla la utilidad de cada archivo:

- memoria.pdf – Versión electrónica de este documento.
- **Ontologías por bloques** – Cada bloque contiene la/s ontología/s que queda como resultado de desarrollar la lección correspondiente.
- Pizza\_Ontologia.owl – Ontología de la Pizza desarrollada a lo largo del curso.
- install\_protege\_4.2\_beta.exe – Instalador de la herramienta Protégé.
- ProtegeOWLTutorialP4\_v1\_3.pdf – Tutorial oficial de Protégé OWL.
- Guia\_Usuario.pdf – Manual de uso del curso desarrollado en Moodle.