

Universidad de las Ciencias Informáticas

Facultad 7



**Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

**Título: Análisis y diseño de un agente semántico basado en
ontologías para el dominio de la salud**

Autor: Yanssel Urquijo Morales

Tutores: Lic. Roberto Acosta González
Ing. Yosvanys Sánchez Corales

Ciudad de La Habana, Junio del 2009
"Año del 50 aniversario del triunfo de la Revolución"



*"La inteligencia consiste no sólo en el conocimiento,
sino también en la destreza de aplicar los
conocimientos en la práctica".*

Aristóteles

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Yanssel Urquijo Morales

Autor

Lic. Roberto Acosta González

Tutor

Ing. Yosvanys Sánchez Corales

Tutor

DATOS DE CONTACTO

TUTOR: Lic. Roberto Acosta González: Graduado en la especialidad de Ciencia de la Computación en el año 2007. Imparte la asignatura de Programación. Ha presentado ponencias y trabajos en eventos científicos obteniendo diferentes reconocimientos y premios. Es líder del Proyecto “Colaboración Médica” y dirige el grupo de trabajo de base de datos del Área Temática Sistema de Apoyo a la Salud en la Facultad número siete de la Universidad de las Ciencias Informáticas. Su dirección de correo es raglz@uci.cu.

TUTOR: Ing. Yosvanys Sánchez Corales: Profesor graduado de Ingeniero en Informática en el año 2005 en la CUJAE. Ha impartido las asignaturas Programación 3 e Inteligencia Artificial, Forma parte del proyecto de Atención Primaria de Salud de la facultad 7. Pasó cursos de diplomado para la maestría GPI (Gestión de Proyectos Informáticos), actualmente cursa la maestría. Ha participado en eventos científicos. Su dirección de correo es yscorales@uci.cu.

AGRADECIMIENTOS

Antes de adentrarme a mencionar nombres quisiera agradecer de forma general a todas aquellas personas que de una forma u otra hicieron que mi estancia en la universidad fuese una bella página en mi vida. Aquellos que me ayudaron, criticaron y depositaron en mí su confianza.

Agradezco a mis padres por su apoyo incondicional, por confiar en mí y guiarme siempre por el camino correcto, por su amor, comprensión y estar orgullosos de mí. A mis hermanos por ser más que un faro de guía en el transcurso de mi vida y estar siempre ahí cuando los necesito. Al resto de mi familia, mi abuelita, mi tía y mis tíos por elogiarme tanto y darme fuerzas cuando la necesito, a mi padrastro. A mi primo por encaminarme en mi vocación como informático. A Jenny por ser esa persona con la que compartí tanto tiempo y por apoyarme siempre en mi camino hasta aquí.

Un agradecimiento especial para el cariñosamente llamado grupo de logística (Osvaldo, Ariel, Romel y Dunieski), por dejarme compartir tantos ratos con ustedes y ser siempre mis amigos. A Niurki, a Raque, a Jenny, a Sol, a Leydis y Daimy por dejarme compartir tantos secretos con ustedes, por sus consejos y por ser como hermanas para mí. A mis amigos Yoel, Montalvan, Yerandy, Leshter y Alberto por su amistad incondicional. A Yosmany por soportarme tanto tiempo.

A Celia, a quien cariñosamente llamo Ángel pues eso ha significado para mí, gracias por llegar en el momento que más lo necesitaba y compartir maravillosos días conmigo. Gracias por todo tu apoyo y por las cosas que me has enseñado.

A Adrian, Dayrel y el profesor Juan Carlos por ser en esta escuela aquellas personas que siempre me apoyaron y enseñaron gran parte de lo que sé, por su confianza en mí y ayudarme a estar hoy aquí culminando mis estudios. A mis tutores y oponente por guiarme y apoyarme en la realización de este trabajo.

A todos los que no menciono pero que forman parte de mi historia en la UCI muchas gracias por las alegrías compartidas.

DEDICATORIA

Este trabajo lo dedico a una persona que merece más que esto en la vida: mi mamá.

A ti porque te debo todo en esta vida, tú siempre estuviste ahí para mí y me guiaste hasta aquí para verme triunfar. Sin importar el pasar del tiempo este triunfo siempre será tuyo pues este sueño común ya hoy es realidad.

A mis hermanos por estar tan orgullosos de mí, por su apoyo y por todo lo que representan para mí.

RESUMEN

El crecimiento explosivo de información y sistemas que experimenta actualmente la red de informática médica, exige un mecanismo que permita localizar la información precisa que necesitan los usuarios. Encontrar lo que se busca, gestionar el conocimiento y actualizar los datos no es tarea fácil para los diferentes técnicos, especialistas e investigadores de esta rama. Los mismos, hacen uso de estos servicios en su afán de superación personal o para realizar su aporte a las ciencias médicas, en especial al área de Atención Primaria de Salud (APS).

Para dar solución a esta problemática en el presente trabajo se propone el desarrollo de un Agente Semántico Inteligente basado en ontologías, que permita consultar y aprovechar de manera eficiente, el gran caudal de información relacionada a los sistemas de APS desplegados en la red INFOMED. Lo cual brindará una mayor información sobre estas aplicaciones.

Como resultado de esta investigación se espera obtener el análisis y diseño para la posterior implementación de un agente semántico basado en ontologías. Se sigue la metodología *Rational Unified Process* (RUP). Además la implementación de una ontología que englobe los sistemas de APS, guiándose por los pasos descritos en la guía Desarrollo de ontologías 101. Se utiliza para el modelado de la misma, la herramienta *Protégé* en su versión 3.3.1.

PALABRAS CLAVES:

Agente Semántico, Conocimiento, Inteligencia Artificial, Ontología, Taxonomía.

TABLA DE CONTENIDOS

INTRODUCCIÓN	- 1 -
CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	- 5 -
1.1 Lenguajes de marcado ontológico.	- 5 -
1.2 Ontologías	- 11 -
1.2.1 Tipos de ontologías	- 13 -
1.3 Las ontologías y la inteligencia artificial.....	- 14 -
1.4 Metodologías para la construcción de ontologías.....	- 15 -
1.5 Agentes.....	- 16 -
1.5.1 Tipologías de agentes	- 17 -
1.5.2 Agentes inteligentes	- 18 -
1.6 Relación Agente inteligente - Web semántica.....	- 18 -
1.7 Analizador Sintáctico	- 18 -
1.7.1 Visión general del proceso.....	- 19 -
1.8 Trabajos relacionados	- 19 -
1.9 Herramientas para el trabajo con ontologías.....	- 21 -
1.10 Metodología y herramientas utilizadas para el diseño del agente	- 26 -
1.11 Resultados esperados.....	- 27 -
Conclusiones	- 28 -
CAPÍTULO 2. DISEÑO DE LA ONTOLOGÍA	- 29 -
2.1 Tipo de ontología seleccionada.....	- 29 -
2.2 Metodología seguida	- 29 -
2.3 Elementos de una ontología	- 30 -
2.4 Descripción de los pasos seguidos	- 31 -
2.5 Representación del Conocimiento del Registro de Fallecidos.....	- 36 -
2.6 Representación del Conocimiento del Registro de Parto y Nacimiento	- 37 -
2.7 Razonadores	- 38 -
2.7.1 Ejemplo de razonador	- 39 -
Conclusiones	- 39 -
CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL AGENTE	- 40 -
3.1 Modelo de Dominio	- 40 -
3.1.1 Conceptos fundamentales	- 40 -
3.1.2 Diagrama del Modelo de Dominio	- 41 -
3.2 Propuesta del sistema.	- 41 -

3.2.1 Especificación de requerimientos de software	- 41 -
3.2.2 Requerimientos funcionales.....	- 42 -
3.2.3 Requerimientos no funcionales	- 42 -
3.3 Definición de casos de uso	- 44 -
3.3.1 Definición de actores del sistema.....	- 45 -
3.3.2 Listado de casos de uso del sistema	- 45 -
3.3.3 Diagramas de casos de uso del sistema	- 47 -
3.3.4 Especificación de los casos de uso.....	- 47 -
3.4 Patrones de Arquitectura a utilizar	- 52 -
3.5 Análisis.....	- 52 -
3.6 Diseño.....	- 53 -
Conclusiones	- 55 -
CONCLUSIONES GENERALES.....	- 56 -
RECOMENDACIONES.....	- 57 -
REFERENCIAS BIBLIOGRÁFICAS	- 58 -
BIBLIOGRAFÍA.....	- 60 -
ANEXOS	- 62 -
GLOSARIO DE TÉRMINOS.....	- 67 -

INTRODUCCIÓN

En Cuba como parte del programa existente para el bienestar social se le otorga máxima importancia a la salud de todo el pueblo. En virtud de esto se disponen de todos los medios necesarios para el logro de dicho propósito. Dentro del programa de adelantos científicos que se promueven, se ha incrementado y perfeccionado el estudio de la informática. La que con su desarrollo ha demostrado la amplia gama de aplicaciones que tiene en la sociedad, en especial en la solución de problemáticas y apoyo del trabajo de las variadas instituciones del país. Se diseñan y organizan estrategias que permiten convertir los conocimientos y las tecnologías de la información y las comunicaciones; en instrumentos a disposición del avance de las profundas transformaciones en el sector.

Debido a este proceso surge en el sector de la salud cubana, la informatización del Sistema Nacional de Salud (SNS), dirigida al manejo de la información en esta área, donde se incrementa la efectividad y eficiencia en el entorno de la salud asistencial, donde se tiene como eslabón principal, la población, por ser el elemento beneficiado al obtener organización, calidad y consistencia de la información.

Gracias al proceso de informatización de la sociedad cubana, comenzó a desarrollarse a partir de 1994 el Portal de la Salud, INFOMED, con el objetivo de facilitar el acceso a la información relacionada con las ciencias de la salud y especialmente dar acceso a la información de salud originada en Cuba. (1)

INFOMED es la red telemática disponible para personas e instituciones que trabajan y colaboran en el sector de la salud. Facilita el acceso a la información y el conocimiento con el objetivo de mejorar la salud de los cubanos y de los pueblos del mundo; mediante el uso intensivo y creativo de las tecnologías de la información y las comunicaciones. A esta red, están conectadas todas las unidades de salud del SNS, por lo que el volumen de sistemas alojados en los servidores es cada vez mayor.

El crecimiento explosivo de información y sistemas que experimenta actualmente la red de informática médica, exige un mecanismo que permita localizar la información precisa que necesitan los usuarios. Encontrar lo que se busca, gestionar el conocimiento y actualizar los datos no es tarea fácil para los diferentes técnicos, especialistas e investigadores de esta rama, que hacen uso de estos servicios en su

afán de superación personal o aporte a las ciencias médicas, en especial al área de Atención Primaria de Salud (APS).

El hecho de que los usuarios posean una herramienta llamada Localizador de Información de Salud (LIS) (2), con la capacidad de buscar y procesar un gran caudal de textos relacionados con un tema específico no es suficiente. Puesto que se centra en el acceso a los recursos de las instituciones de información del sector de la salud y no es capaz de brindar información sobre los sistemas alojados en los servidores. Lo cual da lugar a que los usuarios desconozcan muchas de las aplicaciones que integran la red de salud.

Dada la situación anterior el **problema** a resolver radica en: *La ineficiencia del proceso de recuperación de la información sobre los sistemas alojados en la red de informática médica.* Para dar solución al problema planteado, se define como **objeto de estudio**: *El proceso de recuperación de la información sobre los sistemas para la salud alojados en INFOMED.* Y para enmarcar la investigación en un ambiente más definido se tiene como **campo de acción**: *La recuperación de la información sobre los sistemas para la salud alojados en la red de informática médica empleados en la Atención Primaria de Salud.*

Para el desarrollo de la investigación se plantea como **objetivo general**, *diseñar un agente semántico basado en ontologías que permita consultar la información sobre los sistemas para la Atención Primaria de Salud que se encuentran en la red de informática médica (INFOMED).*

Para lograr el objetivo planteado anteriormente se trazaron las siguientes tareas de la investigación:

1. Analizar el estado del arte de la gestión de la información de los sistemas para la salud hospedados en la red INFOMED.
2. Asimilar las tecnologías para la representación y gestión del conocimiento mediante ontologías.
 - 2.1. Comparar los diferentes lenguajes ontológicos RDF, RDF-S, OIL, DAML+OIL y OWL.
 - 2.2. Identificar las diferentes herramientas necesarias para el desarrollo y mantenimiento de las ontologías.

3. Crear una representación del conocimiento mediante ontologías de los sistemas de información para la Atención Primaria de Salud.
 - 3.1. Identificar los conceptos claves para el desarrollo de la ontología a partir del problema descrito.
 - 3.2. Implementar la ontología para la atención primaria de la salud.
4. Examinar acerca de la implementación de estas tecnologías en Java, utilizando el framework Jena.
 - 4.1. Comparar los diferentes lenguajes de consultas para ontologías RDQL, SeRQL, SPARQL, etc.

Como resultado de esta investigación se espera obtener el análisis y diseño para la posterior implementación de un agente semántico basado en ontologías, que permita aprovechar de manera eficiente el gran caudal de información relacionada a los sistemas de atención primaria de la salud con que cuenta la red INFOMED, así como la implementación de una ontología de los sistemas de APS, la cual será consultada por dicho agente.

La estructura del presente documento se compone de tres capítulos en los que se recoge la información obtenida desde el punto de vista investigativo. Además de lo referente al análisis y diseño del agente semántico, así como la propuesta de la ontología para el dominio de los sistemas de Atención Primaria de Salud.

En el **CAPÍTULO 1: Fundamentación Teórica:** Se aborda todo lo relacionado con la fundamentación teórica que sustenta la presente investigación: Ontología y Agentes Inteligentes. Se describen los diferentes lenguajes ontológicos, se define el concepto de "Ontología" y se indican los distintos tipos de ontologías existentes. Se enumeran diversos aspectos relacionados con el mundo de los agentes. También se exponen las tendencias, técnicas, tecnologías, métodos y herramientas usadas durante la investigación y construcción de la solución propuesta.

En el **CAPÍTULO 2: Diseño de las Ontologías:** Se abordan los temas relacionados al proceso construcción de las ontologías. Se crea una ontología para el dominio de APS, otra para el Registro de Fallecidos y por último la del Registro de Parto y Nacimiento, guiándose a través de una serie de pasos, así como un conjunto de herramientas seleccionadas para la elaboración de la misma.

En el **CAPÍTULO 3: Análisis y Diseño del Agente Semántico**: Se describe un conjunto de herramientas y tecnologías a seguir para la implementación de un agente semántico. Se realiza un análisis y diseño general de un agente semántico para su posterior implementación.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En este capítulo se definen los términos fundamentales que han sido utilizados en el presente trabajo. Se realiza un análisis del estado del arte a nivel nacional e internacional de los sistemas informáticos que hacen uso de las ontologías como representación del conocimiento. Se presentan y se justifican un conjunto de técnicas, herramientas y tecnologías utilizadas durante el desarrollo del sistema propuesto.

1.1 Lenguajes de marcado ontológico.

El acelerado crecimiento de Internet y los servicios que brinda ha permitido el acceso a grandes volúmenes de información localizados en diferentes sistemas. La información presente en la Web se encuentra estructurada mediante lenguajes de etiquetado que solo describen la manera en que debe ser mostrada al usuario pero no expresan nada relacionado al significado, o sea, su semántica.

Para conseguir esto se deberá codificar la semántica de los recursos web mediante lenguajes de metadatos y ontologías (representaciones compartidas de conocimiento en forma de taxonomías de conceptos relacionales y reglas de inferencia).

Para dar solución a esto la organización *World Wide Web Consortium (W3C)* junto a otras organizaciones se dieron la tarea de crear nuevas tecnologías, herramientas y lenguajes para representar los recursos en la red con un sentido semántico. Estos lenguajes se utilizan para expresar las descripciones de los recursos en la web con base en objetos relacionados u ontologías. A continuación se describen estos lenguajes. (3)

RDF

RDF (*Resource Description Framework*), es uno de los lenguajes desarrollados por el W3C. RDF hace uso de lenguaje de marcas ampliable (XML, siglas de *Extensible Markup Language*) como sintaxis común para la representación de metadatos. Permite el intercambio y procesamiento de los mismos, provee independencia, extensibilidad, validación, legibilidad humana, y la habilidad para representar estructuras complejas. Su sintaxis permite codificación, intercambio y procesamiento automático de los metadatos normalizados. (4)

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

RDF basa su modelo en tres partes:

- Recursos (sujeto): todos los objetos que pueden ser expresados con expresiones RDF son llamados recursos; por ejemplo, una página Web o parte de una página Web; también puede ser un conjunto de páginas Web o un documento. Es todo aquello que pueda ser referenciado por un identificador único de recursos (URI).
- Propiedades (predicado): es un aspecto específico, atributo o relación que permite describir un recurso.
- Expresión (objeto): un recurso o bien un literal específico junto con el nombre de una propiedad, más el valor de ésta formarán una expresión RDF.

En el ejemplo a continuación ilustra detalladamente la composición de una expresión RDF:

John Smith es el **autor** del artículo **<http://www.ejemplo.org/index.html>**

De la expresión anterior se tiene que:

Recurso (sujeto)	http://www.ejemplo.org/index.html
Propiedades (predicado)	autor
Expresión (objeto)	John Smith

Tabla 1. Elementos de una expresión en RDF.

Esta expresión se puede representar de la siguiente forma:

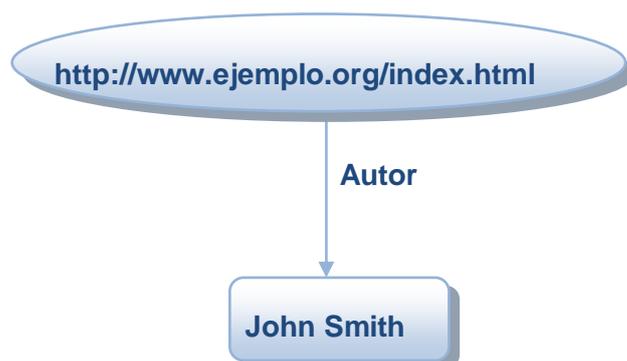


Figura 1. Representación gráfica de una expresión RDF.

Donde los arcos representan las propiedades, los recursos son representados por nodos circulares y los nodos rectangulares representan los literales, o sea, los valores. Todo esto llevado al lenguaje RDF quedaría de la siguiente forma:

```

<?xml version="1.0">
<rdf:RDF>
<rdf:Description about="http://www.ejemplo.org/index.html">
<s:autor> John Smith </s:autor>
</rdf:Description>
</rdf>

```

Tabla 2. Representación en código de una expresión RDF.

Básicamente RDF es un diagrama formado por nodos y arcos interpretados como afirmaciones sobre objetos identificados por referencias URI.

RDF Schema

RDF-S (*Resource Description Framework Schema*) proporciona un mecanismo para definir un vocabulario para datos RDF, a su vez permite definir jerarquías de clases (relación de subclase y subpropiedad), de objetos y propiedades (relaciones binarias) y admite la creación de restricciones (rango y dominio). RDF-S cuenta con un conjunto de términos que permiten construir las expresiones RDF-S válidas.

Así, se tienen los términos *Class*, *subClassOf* y *Property* que permitan construir expresiones acerca de los recursos (clases), representar sus propiedades y a la vez representar una jerarquía de recursos. Los objetos podrán ser instanciados a partir de las clases usando la propiedad *type*. Las restricciones de las propiedades podrán ser especificadas mediante las primitivas *domain* y *range*. (5)

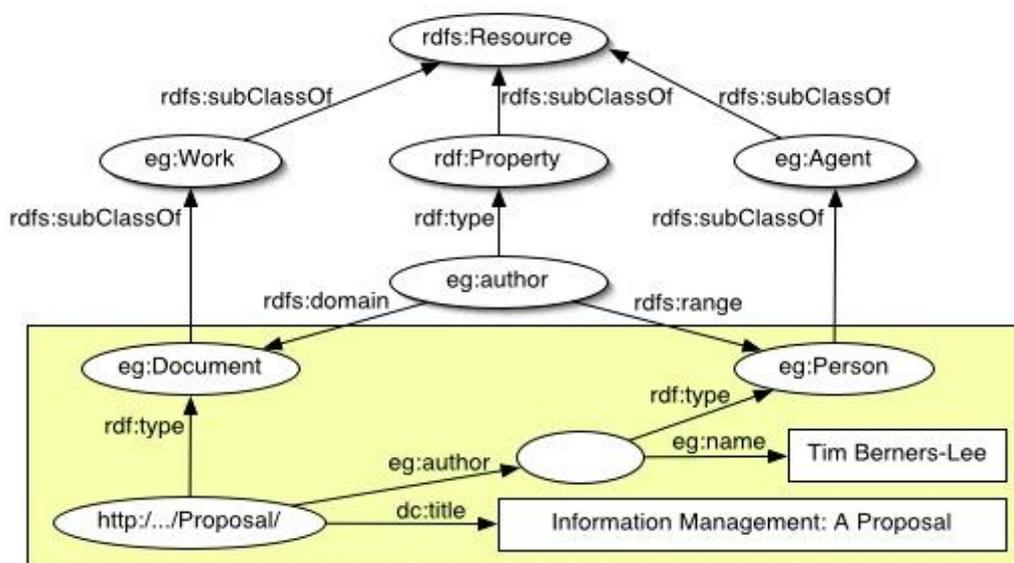


Figura 2. Representación de una ontología en RDF-S.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Este esquema se aproxima más al concepto de ontología ya que se dispone de relaciones diseñadas para especificar la jerarquía de la taxonomía de conceptos que define un conocimiento. RDF-S funciona como un modelo semántico de datos capaz de permitir preguntas referentes a su contenido y no a la estructura del documento.

OIL

OIL (*Ontology Inference Layer*) es el primer lenguaje de representación de ontologías basado en estándares W3C. Al igual que RDF tiene sintaxis XML y está definido como una extensión de RDF-S. Fue una propuesta de un lenguaje que permitiese la especificación de ontologías y que además sirviese como lenguaje de intercambio de éstas.

Para la representación del conocimiento OIL ha heredado una parte de la Lógica Descriptiva (declaración de axiomas o reglas) y, por otra, de los sistemas basados en *frames* (taxonomía de clases y atributos), donde unifica tres criterios importantes provistos por diferentes comunidades:

- La semántica formal y el soporte eficiente para razonamiento, como las provistas en descripciones lógicas.
- El enriquecimiento epistemológico de las primitivas, como el provisto por los *frames*.
- Un estándar para notaciones de intercambio sintáctico como la provista por la comunidad del Internet.

Las ontologías en OIL está organizada en tres niveles: un nivel objeto (donde residen las instancias); un primer nivel meta donde residen las definiciones y descripciones de la ontología y por último un segundo nivel meta donde reside un contenedor sobre la ontología, es decir, donde reside la información sobre las características de la ontología. OIL también soporta axiomas por lo que el razonamiento sobre ontologías de este tipo es posible. (6)

DAML+OIL

La unión de los lenguajes DAML (*Darpa Agent Markup Language*) y OIL (*Ontology Inference Layer*) surgió de la necesidad de expresar de manera más detallada los recursos disponibles en la web, extendie el nivel de expresividad de RDF-S. Aunque el lenguaje tiene muchas coincidencias con OIL, se distancia de la idea de los *frames* para acercarse más a la lógica descriptiva. Por tanto DAML+OIL es un lenguaje que

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

brinda mas expresiones para las descripciones de clasificaciones y propiedades de los recursos que poseía RDF y RDF-S por lo que representa la evolución de estos ya que posee muchas de sus descripciones e incorpora otras para enriquecer el lenguaje.

DAML+OIL divide el universo de descripción en dos partes. Una parte consiste en el valor (*datatypes*) representado en XML *Schema* la cual es llamada *datatype domain*. La otra parte consiste en objetos que son considerados miembros dentro de las clases descritas en DAML+OIL (o RDF) y a esta parte le llama *object domain*. (7)

OWL

OWL (*Web Ontology Language*) es el más completo de los lenguajes utilizados para la representación del conocimiento en la web. OWL abarca las potencialidades ofrecidas por el lenguaje DAML+OIL y al igual que este es una extensión del modelo RDF-S. También incorpora otro gran número de primitivas para representar el significado de los elementos y sus relaciones dentro de la ontología. (8)

A continuación se enumeran algunas de las características adicionales que incorpora OWL: (9)

- Definición de clases mediante restricciones sobre propiedades, valores o cardinalidad.
- Definición de clases mediante operaciones booleanas sobre otras clases: intersección, unión y complemento.
- Relaciones entre clases (ej. Inclusión, disyunción, equivalencia).
- Cardinalidad (ej. “únicamente una”).
- Igualdad y desigualdad de clases.
- Igualdad y desigualdad de instancias.
- Clases enumeradas.

Las mejoras que aporta OWL son:

- Capacidad de representar ontologías distribuidas a través de varios sistemas.
- Escalable a las necesidades de la Web.
- Compatible con los estándares Web de accesibilidad e internacionalización
- Abierto y extensible.

Dada las desventajas que surgieron tras la incorporación de estas características en cuanto a la completitud computacional (el razonador encuentre todas las conclusiones validas) y la decidibilidad computacional (se obtienen respuestas para cualquier

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

entrada en un periodo de tiempo finito), se decidió separar OWL en tres niveles o lenguajes:

- OWL Lite: la versión más simple para los programadores principiantes. Permite la jerarquía de clasificación y las restricciones simples.
- OWL DL: esta versión ya tiene todo el vocabulario OWL completo. Las limitaciones son que las clases no son instancias ni tipos y los tipos no son ni instancias ni clases. No permite restricciones de cardinalidad en propiedades transitivas. Posee gran expresividad sin perder las propiedades de completitud y decidibilidad.
- OWL Full: esta versión también incluye todo el vocabulario de OWL pero en este caso no hay limitaciones para explotar todo su potencial. Sin garantías computacionales.

Cada uno de estos sublenguajes es una extensión de su predecesor más simple, respecto a lo que puede ser expresado legalmente y a la validación de sus conclusiones. Para ello se establecen estas relaciones, pero las relaciones inversas no se permiten.

- Cada ontología legal de OWL Lite es una ontología legal de OWL DL.
- Cada ontología legal de OWL DL es una ontología legal de OWL Full.
- Cada conclusión válida de OWL Lite es una conclusión válida de OWL DL.
- Cada conclusión válida de OWL DL es una conclusión válida de OWL Full.

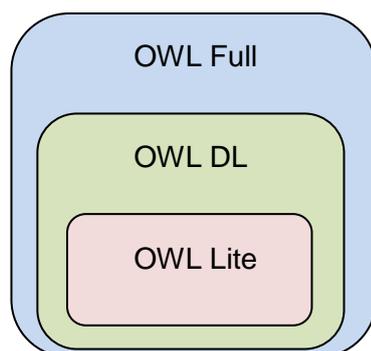


Figura 3. Niveles de OWL.

A continuación se muestra una tabla comparativa entre los lenguajes ontológicos más usados en la web para la representación del conocimiento.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

	XML DTD	XML SCHEMA	RDF(S)	DAML+OIL	RDF(S) 2002	OWL
Listas Limitadas				X	X	X
Restricciones cardinalidad	X	X		X		X
Expresiones de clases				X		X
Datatypes		X		X	X	X
Clases definidas				X		X
Enumeraciones	X	X		X		X
Equivalencias				X		X
Extensibilidad			X	X	X	X
Semántica formal				X	X	X
Herencia			X	X	X	X
Inferencia				X		X
Restricciones locales				X		X
Restricciones cualificadas				X		
Reificación			X	X	X	X

Tabla 3. Comparación de lenguajes ontológicos.

De los lenguajes expuestos en este epígrafe se decide escoger el OWL para el desarrollo de las ontologías. Esta decisión se basa en que es el más completo para la representación del conocimiento en la web. Además por ser una evolución de los anteriores e incorporar una serie de primitivas para enriquecer la expresividad de los elementos y sus relaciones.

Lo que se persigue es modelar un conocimiento que estará disperso en la red cosa que no es posible con otros lenguajes debido a las limitantes que presentan y a la poca expresividad que poseen.

1.2 Ontologías

El conocimiento se puede ver como información de la información, básicamente es una información semánticamente "rica". Con fin de darle un carácter semántico y especializado a los recursos contenidos en la web, categorizando y catalogando la información de acuerdo a un área específica, se hace inevitable el uso de las ontologías. Este concepto de ontología proviene de tiempos atrás, en la filosofía y se define como:

Campo que se ocupa de la naturaleza y organización de la realidad, es decir de lo que "existe".

La siguiente figura ilustra gráficamente un ejemplo de una ontología que recoge las clasificaciones en las que está dividido el reino animal.

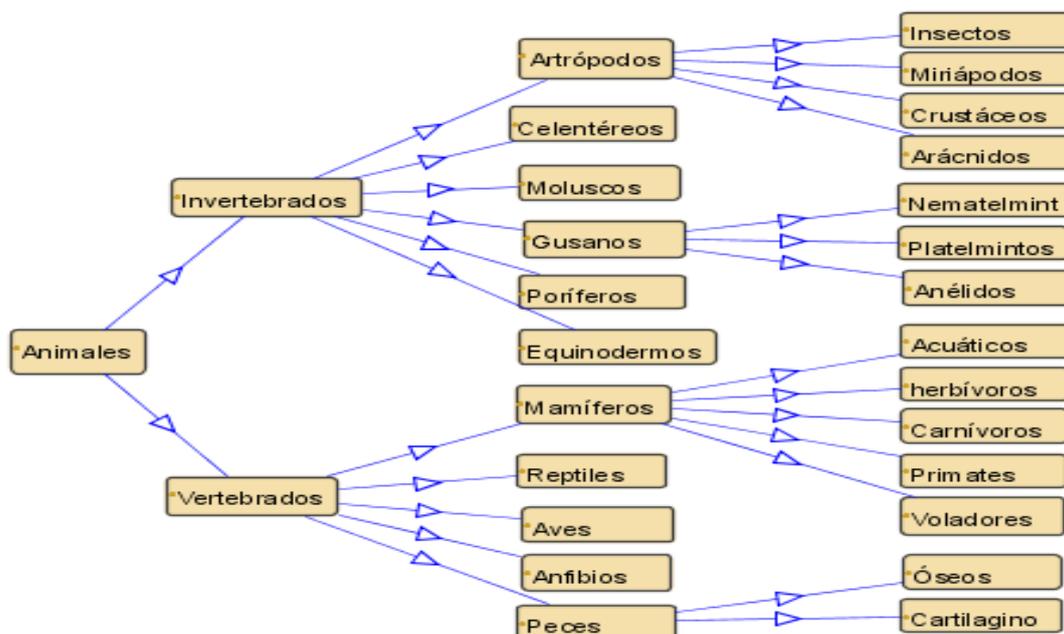


Figura 4. Ontología del mundo animal.

En la actualidad se ha retomado este concepto de ontología, aplicado en la informática para definir: vocabularios entendibles por las máquinas mediante la creación de un exhaustivo y riguroso esquema conceptual dentro de uno o varios dominios dados.

Las ontologías fueron desarrolladas en el área de Inteligencia Artificial (IA) para facilitar el intercambio y reutilización del conocimiento. Actualmente, las computadoras han dejado de ser dispositivos aislados y se han convertido en puntos de entrada en la red mundial de intercambio de información y transacciones de negocio. Se ha vuelto un asunto clave contar con apoyo para el intercambio de datos, información y conocimiento.

Emplear el poder de razonamiento automático, para guiar el acceso a orígenes de información, requiere de representación de la semántica de tales orígenes. De manera que puedan ser procesadas por la computadora, en consecuencia, se requieren metadatos que describan de una manera computable, dichos orígenes.

Los principales objetivos de las ontologías son los siguientes:

- Compartir la comprensión de estructuras de información entre personas o agentes de software. Esto debe revertir de forma positiva y casi necesaria en la extracción y recuperación de información, desde fuentes con contenidos conectados temáticamente.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Permitir la reutilización del conocimiento perteneciente a un dominio en la elaboración de una nueva ontología o su utilización en la solución de un problema.
- Permite hacer explícitos los supuestos de un dominio. Esta aseveración puede conducir a conclusiones muy interesantes para la representación del conocimiento más allá de consideraciones técnicas, operativas e informáticas.
- Separa el conocimiento de un dominio del conocimiento que se puede tener de un proceso. El conocimiento que se representa en una ontología se puede aplicar en diferentes áreas de conocimiento independientemente del proceso que se modele.
- Hace posible analizar el conocimiento de un campo, por ejemplo en lo que se refiere al estudio de los términos y relaciones que lo configuran ya sea formalmente o no.

Una ontología es una jerarquía de conceptos con atributos y relaciones, que define una terminología consensuada para definir redes semánticas de unidades de información interrelacionadas. Es un vocabulario común para compartir información dentro de un dominio concreto, está formado por clases o conceptos, propiedades o atributos de las clases y relaciones entre clases.

Las ontologías son colecciones de enunciados redactados en un lenguaje, que define las relaciones entre conceptos y especifica reglas lógicas para razonar con ellos. Los ordenadores "comprenderán" el significado de los datos semánticos de una página de la red a través de vínculos con ontologías especificadas".

Cada vez más, de forma creciente las ontologías juegan un rol significativo en la recuperación de información y en los procesos de compartir, reutilizar y adquirir conocimiento. Las ontologías hacen posible compartir una comprensión común de la estructura de la información entre personas o agentes de *software*. Reutilizar el conocimiento de dominio, hacer explícitos presupuestos de dominios, separar el conocimiento de dominio del conocimiento operacional y analizar el conocimiento de dominio.

1.2.1 Tipos de ontologías

En la literatura se puede encontrar diferentes clasificaciones de tipos de ontologías. De acuerdo a dos criterios para tales clasificaciones, se encuentran los siguientes:

- Según el tipo de conocimiento que contienen.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- Ontologías terminológicas, lingüísticas.
- Ontologías de información.
- Ontologías para modelar conocimiento.

Alternativas a estos criterios se tienen los siguientes:

- Ontologías del dominio.
- Ontologías de tarea.
- Ontologías generales.
- Según la motivación de la ontología.
 - Ontologías para la representación de conocimiento.
 - Ontologías genéricas.
 - Ontologías del dominio.
 - Ontologías de aplicación.

Como alternativas a estos criterios se encuentran los siguientes:

- Ontologías generales.
- Ontologías categóricas.
- Ontologías del dominio.
- Ontologías genéricas.
- Ontología regional.
- Ontología aplicada.

1.3 Las ontologías y la inteligencia artificial

En el campo de la inteligencia artificial, la ontología se define como una descripción formal y explícita de conceptos de un dominio (clases, a las que se hace referencia como conceptos). Además, a las propiedades de cada concepto, las cuales describen las características y atributos de estos (*slots* llamados también como propiedades o roles) y a las restricciones de las propiedades (*facets* conocidas como restricciones).

Los programas informáticos pueden utilizar así este punto de vista de la ontología para una variedad de propósitos. Lo que incluye el razonamiento inductivo, la clasificación y una variedad de técnicas de resolución de problemas.

Para usar las ontologías en la web y determinar el significado contextual de una palabra. Se necesitan agentes inteligentes que rastreen la red de forma automática a través de los vínculos entre las diferentes ontologías y localicen exclusivamente la información referente a la palabra en cuestión. Por lo tanto, para potenciar el uso de ontologías en la Web, se necesitan aplicaciones específicas de búsqueda de ontologías.

1.4 Metodologías para la construcción de ontologías

Las metodologías proporcionan un conjunto de directrices que indican cómo hay que llevar a cabo las actividades identificadas en el proceso de desarrollo, qué técnicas son las más apropiadas en cada actividad y qué produce cada una de ellas.

Para guiar el proceso de desarrollo de ontologías existen diferentes tipos de metodologías entre las que se destaca **Methontology**, es una metodología recomendada por FIPA (acrónimo en inglés de, *Foundation for Intelligent Physical Agents*). Desarrolla su función mediante las tareas de especificación, adquisición de conocimiento, conceptualización, integración, implementación, evaluación y documentación de las ontologías. (10)

Otra metodología de este tipo (orientada a proceso) es **On-To-Knowledge** (OTK), la cual contiene identificación de metas, objetivos, que son logrados con herramientas de soporte para el manejo de conocimiento. Las distintas fases que promueven son estudio de factibilidad, fase inicial, refinamiento, inferencia, evaluación, y aplicación-evolución. (11)

Por otra parte la metodología de **Grüninger y Fox** está basada en la identificación de escenarios y la formulación de preguntas de competencia (*Competency Questions*), extracción de conceptos y relaciones relevantes y la formalización en Lógica de Primer Orden. (12)

Ontology Development 101, es una guía que engloba los pasos comunes que presentan las diferentes metodologías existentes. Es una simple ingeniería del conocimiento muy convincente para el comienzo del desarrollo de una ontología. Esta metodología es la propuesta por la herramienta *Protégé* como guía inicial para el desarrollo de ontologías. (13)

La tabla 4, muestra los diferentes métodos y metodologías para la construcción de ontologías que han sido comparados de acuerdo a ciertos criterios.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Característica	CYC	Uchold & King	Grüniger & Fox	KACTUS	METHONTOLOGY	SENSUS	OTK
Ciclo de vida propuesto	Prototipos de desarrollo	No propuesta	Prototipos de desarrollo o incremental?	Prototipos de desarrollo	Prototipos de desarrollo	No propuesto	Incremental y cíclica con Prototipos de desarrollo
Estrategia con respecto a la aplicación	Independiente de aplicación	Independiente de aplicación	Semi-dependiente de aplicación	Dependiente de aplicación	Independiente de aplicación	Semi-dependiente de aplicación	Dependiente de aplicación
Estrategia para identificar conceptos	No especificada	Middle-out	Middle-out	Top-down	Middle-out	No especificada	Top-down, bottom-up, middle-out depende de la aplicación
Uso de una ontología base	Si	No	No	No	No	Si	Depende de los recursos disponibles para el Proyecto
Herramientas que dan soporte	Cyc	Ninguna específica	Ninguna específica	Ninguna específica	ODE WebODE OntoEdit Protégé2000	Ninguna específica (Usualmente Ontosaurus)	OntoEdit con sus plugins

Tabla 4. Comparación de las diferentes metodologías de construcción de ontologías.

1.5 Agentes

No existe una definición clara de agente debido a su utilización en diferentes áreas de las ciencias de la computación. Entre las más mencionadas esta la aportada por Russel y Norving (14):

“Un agente es cualquier cosa capaz de percibir su medioambiente con la ayuda de sensores y actuar en ese medio utilizando actuadores.”

Esta es una definición que se refiere al componente físico del término y su modo de actuar con el ambiente que lo rodea. Otra definición más cercana es la propuesta por Wooldridge y Jennings (15):

“Un agente es un sistema computarizado que está situado en algún entorno, y que es capaz de actuar de forma autónoma en este entorno para satisfacer sus objetivos de diseño.”

Esta definición destaca un poco más de las características propias de los sistemas basados en agentes.

Las principales características que debe cumplir un agente son las siguientes (no es necesario que las posean todas).

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

- **Persistencia:** El código de un agente se ejecuta continuamente, no cuando el usuario lo decide, y decide por sí mismo cuando deje ejecutar alguna acción. Los agentes siempre permanecen activos: se ejecutan en un bucle infinito y observan su entorno, actualizan su estado y determinan qué acciones realizar. Se comportan como buitres (siempre mirando, siempre esperando a que llegue el turno de actuar), pero sin sus siniestras intenciones.
- **Autonomía:** Un agente puede tomar decisiones sin la intervención humana o de otros agentes. Es decir, para cumplir sus objetivos, un agente controla sus propias acciones.
- **Reactividad:** Un agente es capaz de percibir su entorno y mantener un vínculo constante con él, de modo que puede responder adecuadamente a los cambios. Los agentes más reactivos tienen en cuenta la posibilidad de que se produzcan fallos o de que las cosas no vayan como se esperaba.
- **Pro actividad:** Un agente no actúa dirigido solamente por los sucesos que se producen en su entorno, sino que también toma iniciativas para lograr sus objetivos. Es decir, los agentes reconocen y utilizan las oportunidades que se les presentan. Las iniciativas pueden proceder de experiencias pasadas, pues los agentes aprenden del pasado.
- **Habilidad social:** Un agente constituye una entidad social integrada en una sociedad, en la cual hay tareas, recursos y papeles que se distribuyen entre los agentes. Algunos objetivos sólo se pueden conseguir mediante la colaboración con otros agentes o con humanos.

1.5.1 Tipologías de agentes

De acuerdo a la funcionalidad, estos agentes se agrupan según los siguientes criterios.

A. Según su capacidad para resolver problemas:

- Agentes Reactivos
- Agentes Deliberativos

B. Según su autonomía, aprendizaje y cooperación (atributos):

- Agentes Colaborativos o Cooperativos
- Agentes de Interfaz

C. Según su movilidad:

- Agentes Móviles
- Agentes Estáticos

D. Otros tipos:

- Agentes de Información / Internet
- Agentes Híbridos

1.5.2 Agentes inteligentes

El término de agente inteligente suele usarse para referirse a agentes que estén dotados de cierta inteligencia artificial y posean las características mínimas antes expuestas.

Los agentes inteligentes realizan continuamente tres funciones: perciben las condiciones cambiantes del entorno; actúan para modificar las condiciones del entorno. Además de que razonan para interpretar percepciones, resolver problemas, extraer inferencias y determinar acciones.

1.6 Relación Agente inteligente - Web semántica

La Web Semántica se basa en la estandarización de todos sus datos (páginas, servicios, entre otros) de modo tal que pueda ser interpretado por una nueva generación de agentes inteligentes que clasificarán la información de una manera más eficiente para retornar resultados más precisos ante una búsqueda o un pedido de información.

Los agentes web no sólo encontrarán la información de forma precisa, si no que podrán realizar inferencias automáticamente buscando información relacionada con la que se encuentra situada en las páginas, y con los requerimientos de la consulta indicada por el usuario. Se trata de convertir la información en conocimiento, referenciar los datos dentro de las páginas web a metadatos u ontologías con un esquema común consensuado sobre algún dominio.

El papel del agente inteligente en el proceso de recuperación “semántica” de información no debe confundirse con la de un buscador inteligente. El buscador inteligente basa su funcionamiento en la previa indización de los datos, no así por el agente inteligente quien recorrerá la Web a través de los enlaces entre ontologías.

1.7 Analizador Sintáctico

Un analizador sintáctico (en inglés *parser*) es una de las partes de un compilador que transforma su entrada en un árbol de derivación.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

El análisis sintáctico convierte el texto de entrada en otras estructuras (comúnmente árboles), que son más útiles para el posterior análisis y capturan la jerarquía implícita de la entrada. Un analizador léxico crea *tokens* de una secuencia de caracteres de entrada y son estos *tokens* los que son procesados por el analizador sintáctico para construir la estructura de datos, por ejemplo un árbol de análisis o árboles abstractos de sintaxis. El análisis sintáctico también es un estado inicial del análisis de frases de lenguaje natural. Es usado para generar diagramas de lenguajes que usan flexión gramatical.

La mayoría de los analizadores modernos son al menos en parte estadísticos, esto quiere decir que se basan en unos datos de entrenamiento que han sido analizados a mano. Este enfoque permite al sistema reunir información sobre la frecuencia con que ocurren ciertas construcciones en un contexto específico. (16)

1.7.1 Visión general del proceso

El primer estado es la generación de *tokens* o análisis léxico, en este proceso la cadena de entrada se parte en símbolos con significados definidos por una gramática de expresiones regulares

El siguiente estado es el análisis sintáctico lo que significa comprobar que los *tokens* forman una expresión válida, esto se hace usualmente mediante el uso de una gramática libre de contexto que define recursivamente componentes que pueden aparecer en una expresión y el orden en que estos deben aparecer. Las reglas que definen un lenguaje de programación no siempre se pueden expresar únicamente a través de una gramática libre de contexto; por ejemplo la validación de tipos y la declaración correcta de identificadores. Estas reglas pueden expresarse formalmente mediante gramáticas de atributos.

La fase final es el análisis semántico, que trabaja en las implicaciones de la expresión ya validada y realiza las actuaciones pertinentes. (16)

1.8 Trabajos relacionados

Han sido numerosas, las experiencias y los proyectos que se han desarrollado, mediante el uso ontologías. Algunos de estos proyectos se describen a continuación.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

OLID

En este trabajo se propone la creación de la ontología OLID para un laboratorio de investigación y desarrollo, la cual es una ontología que conceptualiza el conocimiento relacionado a este dominio, este el primer paso para poder implementar una aplicación semántica con la capacidad de mejorar la recuperación de la información así como inferir nuevos conocimientos.

OLID está basada en el uso de *Methodology*, una de las más destacadas metodologías para el desarrollo de ontologías. OLID se construyó mediante la herramienta *Protégé* (v 3.1.1) basado en el lenguaje ontológico OWL-DL.

Gestión documental

Esta investigación presenta una arquitectura de un sistema basado en agentes para la recuperación de recursos de información en un entorno de educación. Se construyó una ontología en la cual se representan los documentos utilizados por estudiantes y profesores. Se utilizó el lenguaje *Resource Description Framework Schema* para representar la ontología de documentos.

En la arquitectura del sistema se desarrollaron tres tipos de agentes (*UserAgent*, *SearchAgent* y *OntologyAgent*) que se encargan de la interacción con el usuario, la recuperación de la ontología y la búsqueda de meta-datos. A base de esta arquitectura se desarrolló un prototipo llamado oSEMA para recuperar información en un entorno distribuido.

OnProc

OnProc es un trabajo que presenta las características generales de una ontología que tiene como dominio el proceso jurídico y como objetivo la automatización de estos procesos y la recuperación de información sobre los mismos, incluye los procesos particulares activos que se están desarrollando en un espacio y un tiempo dados. Se describe además el procedimiento que llevan a pasar de un modelo computacional previamente existente hacía el análisis ontológico. Para este proceso utilizaron el lenguaje OWL DL para la representación del conocimiento y el editor de ontologías *Protégé*.

Información de tráfico

Esta tesis doctoral muestra el proceso de creación de una infraestructura ontológica cuyo dominio queda enmarcado en la información sobre tráfico vial, describe los diferentes servicios de información de tráfico. Se propone para esto la creación de servicios web semánticos (SWS) de manera que la información aportada pueda ser almacenada con la adición de significado lo que permite potenciar nuevas capacidades. El planteamiento del uso de SWS como medio de acceso a la información requerida por un usuario, establece que los requerimientos o consultas de éste, sean especificados mediante perfiles de servicio.

Para consultar estas ontologías de manera autónoma se hizo necesario crear un sistema multi-agente basado en las características y ventajas que estos presentan relacionado con la flexibilidad, fiabilidad y modularidad que las aplicaciones distribuidas tradicionales. Así como su habilidad para interactuar con sistemas remotos de todo tipo y, realizar tareas de forma autónoma sin la constante comunicación con el usuario.

Como lenguaje para la representación de las ontologías usaron DAML+OIL. Para la creación del sistema multi-agente usaron la plataforma JADE debido a las ventajas que esta posee para la creación e integración agentes.

1.9 Herramientas para el trabajo con ontologías

De acuerdo al estudio sobre el estado del arte se ha podido identificar una serie de herramientas (las más utilizadas) indispensables para el trabajo con ontologías. Estas herramientas se organizan en las categorías siguientes:

- Desarrollo
- Evaluación
- Combinación e integración
- Herramientas de anotación
- Almacenamiento y consulta
- Aprendizaje

A continuación se da una breve descripción de algunas de estas herramientas:

Herramientas de desarrollo de ontologías

Este grupo incluye herramientas que pueden ser usadas para construir una ontología o reutilizar una ya existente.

Ontolingua

Proporciona un entorno distribuido y colaborativo para la creación, edición, modificación, navegación y utilización de ontologías mediante la Web. También incluye una interfaz de programación de aplicaciones (API) para poder integrar las ontologías del servidor con agentes preparados para Internet. (17)

OntoStudio

Es una herramienta de edición, desarrollo y mantenimiento de ontologías que utiliza medios gráficos. Además, puede almacenarlas en una base de datos relacional. Permite la representación semántica de lenguajes conceptuales y estructuras mediante conceptos, jerarquías de conceptos, relaciones y axiomas.

WebOnto

WebOnto fue diseñada para facilitar la navegación, creación y edición cooperativa de ontologías representadas OCML sin sufrir problemas de interfaz.

Las principales características de WebOnto son:

- Gestión gráfica de ontologías
- Generación automática de instancias a partir de definiciones de clases.
- Inspección de elementos, la herencia de propiedades y el chequeo de consistencia.

OntoEdit

OntoEdit apoya el desarrollo y mantenimiento de las ontologías a través de medios gráficos en un entorno web. Permite la representación semántica de lenguajes conceptuales y estructuras mediante conceptos, jerarquías de conceptos, relaciones y axiomas. La interfaz es abierta y permite a los usuarios ajustar OntoEdit a sus necesidades. Lo que OntoEdit pretende, es ofrecer una herramienta para representar gráficamente ontologías y que además, pueda almacenarlas y posteriormente manipularlas en una base de datos relacional. Aunque OntoEdit está pensado para un entorno web, también puede aplicarse a una Intranet. De esta forma se preserva la seguridad de la información que contiene. (18)

Protégé

Es una herramienta para construir modelos de dominio y aplicaciones basadas en conocimiento con ontologías. En su núcleo, *Protégé* implementa un rico conjunto de estructuras de modelado de conocimiento y actividades que ayudan a la creación,

visualización y manipulación de ontologías en varios lenguajes de representación. La mayor ventaja que presenta *Protégé* es que puede agregar funcionalidades por medio de la arquitectura *plug-in* y de una API Java para construir herramientas y aplicaciones basadas en el conocimiento. (19)

Herramientas de evaluación de ontologías

Estas herramientas tiene como propósito asegurar la calidad tanto de las ontologías como de las tecnologías asociadas, este es un punto importante para evitar problemas de consistencia e integración de ontologías.

ONET-T de Ontolingua.

Permite la verificación de ontologías almacenadas y disponibles en cualquier servidor Ontolingua. ONET-T detecta errores de inconsistencia en taxonomías de conceptos, tales como: Errores de circularidad, errores de partición, redundancia, errores gramaticales y definición formal idéntica de clases.

OntoClean en WebODE

Es una extensión de WebODE que da soporte a las metodologías de evaluación *Methontology* y *OntoClean*. Una de las principales características de esta herramienta es que evalúa acorde a restricciones en la taxonomía, además que el criterio de evaluación de ontologías es expresado declarativamente en el módulo de conceptualización de WebODE.

Herramientas de combinación e integración de ontologías

Estas herramientas están diseñadas para combinar e integrar diferentes ontologías del mismo dominio.

Chimaera

Chimaera es un entorno de combinación y diagnóstico de ontologías basado en navegador web. Fue construido en una plataforma compatible con el sistema de representación OKBC. Inicialmente el proyecto fue para integrarse a Ontolingua, pero puede ser integrado a cualquier editor. Facilita la combinación lo que permite al usuario cargar ontologías y sugiere candidatos potenciales para combinación según el número de propiedades. Esto genera una lista de resolución de nombres que puede usarse como guía para la combinación.

PROMPT

Este es un módulo de extensión de Protege. Dirige al usuario a través del proceso de combinación, identifica posibles puntos de integración, crea sugerencias relativas a las próximas operaciones a realizar.

Además de proporcionar sugerencias al usuario, PROMPT identifica conflictos de diferentes tipos, tales como: conflictos de nombre, referencias nulas, redundancias en jerarquías de clases y restricciones sobre valores de propiedades que violan la herencia de clase. (20)

Herramientas de anotación basadas en ontologías

Este tipo de herramientas han sido diseñadas para permitir al usuario insertar y mantener semiautomáticamente marcas en páginas web basadas en ontologías.

KIM (21)

La plataforma KIM proporciona infraestructura y servicios de anotación semántica, indexado y recuperación basándose en una ontología y una masiva base de conocimiento, entre sus características se destacan:

- Indexado y recuperación.
- Consulta y explotación de conocimiento formal.
- Seguimiento de acontecimientos simultáneos y clasificación.
- Análisis de la evaluación de la población en entidades
- Instancias automáticas de una ontología y anotación dinámica.

Ontomat

Ontomat es una herramienta de anotación de páginas web interactiva y amigable. Posibilita crear y mantener ontologías basadas en OWL para crear instancias, atributos y relaciones. Incluye un navegador de ontologías para explorar la ontología y las instancias y un navegador HTML que visualiza las partes anotadas del texto.

Herramientas de almacenamiento y consulta de ontología

Este grupo permite la facilidad de uso y consulta de ontologías. Para las consultas de ontologías han surgido varios lenguajes tales como: RDQL, SeRQL o SPARQL.

Sesame

Sesame es un API para Java, es decir, un entorno para el desarrollo de aplicaciones en el lenguaje de programación Java para la web semántica. Es un marco de desarrollo para almacenamiento, consulta y razonamiento con RDF y RDF *Schema*. Puede ser usado como base de datos para RDF y RDF *Schema*, o como una librería de Java para aplicaciones que necesitan trabajar internamente con RDF. De manera más general, Sesame proporciona a los desarrolladores de aplicaciones un conjunto de herramientas muy útil para hacer cualquier cosa por uno mismo con RDF. (22)

Owlim (23)

Owlim es la abreviatura de OWLMemSchemaRepositorySAIL para Sesame (capa de almacenamiento e inferencia), el cual soporta el razonamiento parcial sobre OWL DL. Es una implementación en memoria, lo que permite una consulta y recuperación eficiente. Owlim tiene entre sus características más importantes:

- Razonamiento y soporte de lenguaje OWL.

Herramientas de gestión de ontologías

Estas herramientas incluyen varios aspectos del proceso de desarrollo de ontologías como pueden ser edición, navegación, almacenamiento y recuperación.

Jena

Jena es un API para Java, es decir, un entorno para el desarrollo de aplicaciones en el lenguaje de programación Java para la web semántica. Jena permite gestionar todo tipo de ontologías (añadir hechos, borrarlos y editarlos), almacenarlas y realizar consultas contra ellas. Soporta RDF, DAML y OWL y es independiente del lenguaje. Los recursos no están ligados estáticamente a una clase java particular. (24)

Jena considera que la abstracción Java del recurso es sólo una vista del mismo. Incluye varios componentes:

- ARP: un *parser* de RDF
- API RDF
- API de Ontologías para OWL, DAML y RDF *Schema*.
- Subsistema de razonamiento
- Soporte para persistencia
- RDQL: Lenguaje de consultas de RDF

pOWL

pOWL es un marco de trabajo en PHP (procesador de hipertexto) para el análisis sintáctico, almacenamiento, consulta, manipulación, servicio y socialización de bases de conocimiento OWL en un entorno colaborativo web. Presenta características tales como:

- Navegación y edición de ontologías RDF-S/OWL.
- Edición de datos visual.
- Extensible según la filosofía *plug-in*.
- Sistema de consulta RDQL.
- Autenticación en el modelo.

1.10 Metodología y herramientas utilizadas para el diseño del agente

Para construir el modelo de dominio y la representación del conocimiento con ontologías el autor propone la utilización de la herramienta *Protégé* (v 3.3.1). En su núcleo, *Protégé* implementa un rico conjunto de estructuras de modelado de conocimiento y actividades que ayudan a la creación, visualización y manipulación de ontologías en varios lenguajes de representación del conocimiento. La mayor ventaja que presenta *Protégé* es la de agregar funcionalidades por medio de la arquitectura *plug-in* y de una API Java para construir herramientas y aplicaciones basadas en el conocimiento.

De los diferentes lenguajes que existen para representar el conocimiento se seleccionó OWL debido a que es el más completo de los lenguajes utilizados para la representación del conocimiento en la web. OWL es una evolución de los lenguajes DAML+OIL y el RDF-S. Además de ser abierto y extensible, posee la capacidad de representar ontologías distribuidas a través de varios sistemas, es escalable a las necesidades de la web y compatible con los estándares web de accesibilidad e internacionalización.

Existen diversas metodologías para guiar el proceso de desarrollo de una ontología, de acuerdo a la similitud que presentan la mayoría de ellas, se propone la utilización de *Ontology Development 101*, la cual propone una serie de pasos necesarios para la implementación de una ontología. La decisión esta basada en que es la metodología recomendada por la herramienta *Protégé* la cual fue utilizada para la implementación, visualización y manipulación de la ontología.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

Para la gestión y consulta de la ontología se utilizó el *framework* Jena que es una API para Java, es decir, un entorno para el desarrollo de aplicaciones en el lenguaje de programación Java para la web semántica.

Jena permite gestionar todo tipo de ontologías (añadir hechos, borrarlos y editarlos), almacenarlas y realizar consultas sobre ellas. Soporta RDF, DAML y OWL y es independiente del lenguaje. Los recursos no están ligados estáticamente a una clase java particular.

Como metodología de desarrollo para la construcción del agente se propone el empleo de el Proceso Unificado de Desarrollo (RUP siglas de *Rational Unified Process*) que junto con el Lenguaje Unificado de Modelado (UML siglas de *Unified Modeling Language*), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas.

El Proceso Unificado es un proceso de desarrollo de software (conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de *software*). Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de *software*, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. (25)

Para el modelado de los artefactos de software que requiere la metodología de desarrollo empleada, se determinó el uso del *Enterprise Architect* (v 7.0), herramientas de diseño guiado por computadora (CASE, siglas de *Computer Aided Software Engineering*) que combina el poder de la última especificación de UML versión 2.1 con un alto rendimiento y una interfaz intuitiva, para traer un modelado avanzado para el equipo completo de desarrollo e implementación.

Enterprise Architect es una herramienta comprensible de diseño y análisis UML, cubre el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. Es también una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad.

1.11 Resultados esperados

Obtener un agente que interactúe con una o varias ontologías. El agente debe ser capaz de permitir a los usuarios realizar consultas sobre los negocios de los servicios informáticos de APS alojados en INFOMED. El agente debe facilitar a los especialistas

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

la incorporación de nuevo conocimiento o incorporar ontologías sobre nuevos negocios que surjan posteriormente.

Conclusiones

En este primer capítulo se han descrito las principales características, definiciones y aspectos relacionados con el proceso de representación semántica de los recursos en la web. Se han tenido en cuenta las herramientas y lenguajes apropiados para el modelado e implementación del mismo.

CAPÍTULO 2. DISEÑO DE LA ONTOLOGÍA

En este capítulo se describe el proceso de desarrollo de la ontología propuesta para el dominio de salud, se detallan los pasos seguidos así como las características que presenta dicha ontología.

2.1 Tipo de ontología seleccionada

En el capítulo 1 del presente trabajo se relacionaron una serie de criterios que definían los tipos de ontologías. Esto es algo muy importante a tener en cuenta antes de comenzar a elaborar cualquier ontología. Primero se debe definir qué tipo es conveniente utilizar de acuerdo a lo que se pretende desarrollar. De acuerdo a los criterios mencionados anteriormente se determina lo siguiente:

Ya que la presente investigación se centra en modelar un dominio, el dominio de salud, donde se representa un conocimiento específico acerca de los sistemas informáticos alojados en INFOMED, se puede decir que:

Según el tipo de conocimiento que contiene, la ontología propuesta responde a la clasificación de **ontologías para modelar conocimiento** ya que especifican conceptualizaciones de conocimiento y son estas ontologías las más utilizadas en sistemas basados en conocimiento. Una clasificación alternativa sería **ontologías del dominio** debido a que contiene conceptos asociados a un dominio particular.

De igual forma se puede exponer que según la motivación de la ontología, las clasificaciones más acordes serían **ontologías para la representación de conocimiento** pues permiten explicar las conceptualizaciones que subyacen en los formalismos de representación de conocimiento y como alternativa las **ontologías del dominio** ya que se definen conceptualizaciones específicas del dominio.

2.2 Metodología seguida

Como se plantea en el capítulo 1, existen diversas metodologías para guiar el proceso de desarrollo de ontologías, pero no existe ni una sola forma o metodología “correcta” para modelar el conocimiento, por lo que se decide seguir una serie de pasos que son generales y han de tomarse en consideración a la hora de desarrollar una ontología.

CAPÍTULO 2. DISEÑO DE LA ONTOLOGÍA.

Antes hay que tener en cuenta una serie de reglas fundamentales que pueden ayudar mucho en la toma de decisiones respecto al diseño de la ontología propuesta y estas son:

1. No hay una forma correcta de modelar un dominio - siempre hay alternativas viables.
2. La mejor solución casi siempre depende de la aplicación que tienes en mente y las extensiones que anticipas.
3. El desarrollo de ontologías es un proceso necesariamente iterativo.
4. Los conceptos en la ontología deben ser cercanos a los objetos (físicos o lógicos) y relaciones en tu dominio de interés. Esos son muy probablemente sustantivos (objetos) o verbos (relaciones) en oraciones que describen tu dominio.

Es necesario definir para qué va a ser utilizada la ontología y cuán detallada o general será, tener en cuenta las alternativas viables y determinar cuál es la más acertada, sin dejar de pensar en que la ontología no es más que un modelo de la realidad del mundo y los conceptos definidos deben reflejar esta realidad tanto como sea posible.

Después de que se defina una versión inicial de la ontología, esta puede ser evaluada para corroborar su consistencia. Este proceso de diseño iterativo probablemente continuará a través del ciclo de vida entero de la ontología.

2.3 Elementos de una ontología

Antes de pasar a describir cada uno de los pasos es bueno aclarar que en términos prácticos, desarrollar una ontología incluye:

- Definir clases en la ontología.
- Organizar las clases en una jerarquía taxonómica (subclase-superclase).
- Definir slots y describir valores permitidos para esos slots.
- Llenar los valores de los slots para las instancias.

Las clases son el centro de la mayoría de las ontologías pues describen los conceptos de un dominio. Una clase puede tener subclases que representan conceptos que son más específicos que la superclase. Los slots (llamados *properties* en OWL) describen propiedades de clases e instancias. Las instancias (*individuals* en OWL) contendrían los valores que formaría la base de conocimiento.

2.4 Descripción de los pasos seguidos

A continuación se detallan los pasos seguidos para la implementación de la ontología propuesta.

Paso 1. Determinar el dominio y alcance de la ontología

La bibliografía plantea que para comenzar ha de definirse un dominio y alcance que no es más que darle respuesta a las siguientes preguntas:

- ¿Cuál es el dominio que la ontología cubrirá?
- ¿Para qué usar la ontología?
- ¿Para qué tipos de preguntas la información en la ontología debería proveer respuestas?
- ¿Quién usará y mantendrá la ontología?

Se ha definido como dominio para la ontología propuesta, el dominio de la salud y será usada para brindar información sobre los sistemas y negocios que componen el nivel de APS. Brinda respuestas a preguntas relacionadas con que servicios y negocios que estén vinculados a determinadas aplicaciones. También provee quienes son los autores de esos servicios, que describen esos negocios o las restricciones que estos poseen.

Tal es el caso del Registro de Fallecidos, que gestiona los certificados de defunción y estos certificados están relacionados con determinados modelos. Esta ontología será usada por todo el personal de salud y la mantendrán los ingenieros del conocimiento de cada aplicación.

Preguntas de competencia

Una forma de determinar el alcance de una ontología es elaborar una serie de preguntas que la base de conocimientos creada debe ser capaz de responder. Estas preguntas servirán después, más tarde, de control e inicialmente no tienen porque ser tan exhaustivas.

En el dominio que se intenta modelar las posibles preguntas de competencia serían las siguientes.

- ¿Quiénes son los responsables del servicio **rfac**?
- ¿A qué sistemas responde el servicio de **parto**?
- ¿Con qué servicios está relacionado el sistema Registro de Fallecidos?

- ¿A qué tipo de certificado pertenece el modelo x?
- ¿Qué grupo de edades corresponde al certificado de defunción neonatal?

Acorde a estas preguntas, la ontología recogerá información sobre quiénes son los responsables de la implementación de los servicios. A que negocio responden cada uno. Que gestiona dicho negocio; así como los datos relacionados con el negocio en sí.

Paso 2. Considerar la reutilización de ontologías existentes

A pesar de que se debe considerar la reutilización de ontologías existentes con el fin de refinar y extender los recursos que estas poseen con relación al dominio, en esta investigación se decide en parte obviar un poco este paso, pues en Cuba el sistema de salud es muy diferente a otros existentes. No obstante es bueno destacar que se revisaron las bibliotecas de ontologías reusables en la Web, tales como, Ontolingua (<http://www.ksl.stanford.edu/software/ontolingua/>) o la biblioteca de ontologías DAML (<http://www.daml.org/ontologies/>).

También hay un cierto número de ontologías comerciales públicamente disponibles (UNSPSC (<http://www.unspsc.org>), RosettaNet (<http://www.rosettanet.org>), DMOZ (<http://www.dmoz.org>). Es importante destacar que a pesar de no reutilizarse ninguna ontología existente, si brindaron estas un gran aporte a la investigación en lo que respecta a la visión del modelado de dominios, así como el diseño e implementación de la misma.

Paso 3. Enumerar términos importantes para la ontología

En este paso es útil escribir una lista de los términos que tratará la ontología para dar explicación a un usuario. ¿Qué términos debe tratar?, ¿Qué propiedades deben tener? Los términos importantes relacionados a los negocios descritos incluirán **niveles**, diferentes tipos de niveles tales como Atención Primaria de Salud (**APS**), Atención Secundaria de Salud (**ASS**) y Atención Terciaria de Salud (**ATS**), **servicios**, **autores**, **datos de contacto**, **url del servicio**, etc. No importa en estos momentos tener la lista integral de todos los términos ni sus relaciones o propiedades que presenten, eso se determinará en pasos posteriores a esta etapa de diseño.

Paso 4. Definir las clases y la jerarquía de clases

Este paso guarda relación con la definición de las propiedades de las clases, es muy difícil realizar primero uno sin tener presente el otro. Estos dos pasos son los más importantes en el proceso de diseño de la ontología.

Hay varios puntos de partida para crear una jerarquía de clases:

- Un proceso de desarrollo **top-down** comienza con la definición de los conceptos más generales en el dominio y la subsiguiente especialización de los conceptos.
- Un proceso de desarrollo **bottom-up** comienza con la definición de las clases más específicas, las hojas de la jerarquía, con el subsiguiente agrupamiento de esas clases en conceptos más generales.
- Un proceso de desarrollo **combinado** es el resultado de una combinación de los enfoques **top-down** y **bottom-up**: primero se definen los conceptos más sobresalientes y luego los generalizan y especializan apropiadamente.

Ninguno de estos métodos es mejor que cualquiera de los otros. El método que se determine depende de la visión personal de quien modela el dominio. En ocasiones es más conveniente utilizar el método combinado, ya que los conceptos del medio tienden a ser más descriptivos.

Sea cual sea el enfoque elegido, se debe comenzar a definir las clases. Para ello se selecciona de la lista creada en el paso 3, los términos que describen a los objetos que tienen existencia independiente, los cuales serán las clases. Se organizan las clases jerárquicamente, se determina si una instancia de una clase es instancia de alguna otra en particular. O sea, si una clase A es una superclase de la clase B, entonces cada instancia de B lo es también de A.

Lo que refiere el dominio de salud es que, cada sistema informático de salud pertenece necesariamente a un nivel determinado, o sea, el Registro de Fallecidos pertenece a la atención primaria que a su vez es uno de los niveles de salud, y dentro del Registro de Fallecidos están los certificados de defunción así como los reparos y modelos utilizados.

La figura 5, muestra una parte de la jerarquía de clases de la ontología propuesta para el dominio de la salud. Esta ha sido diseñada basándose en el criterio **bottom-up** ya

que se comenzó por el modelado de las clases más específicas de los negocios, en este caso el Registro de Fallecido y luego lo más general como son los niveles de salud.



Figura 5. Jerarquía de clases de la ontología OSalud.

Para una vista más completa Ver Anexo 1.

Paso 5. Definir las propiedades de las clases: *properties*

Las clases aisladas no brindan suficiente información para dar respuesta a las preguntas de competencia listadas en el paso 1, debido a esto después de definir las clases se pasa a crear la estructura interna de los conceptos.

Una vez que se definieron las clases de la lista de términos antes creada en el paso 3, el resto serán propiedades de esas clases, estos son por ejemplo, el correo electrónico de los autores de los servicios, donde radica, la url donde está publicado el servicio, también lo serán los modelos que se utilizan en los certificados de defunción, así como los datos que son recogidos en un parto. De esta forma la clase **Servicio** tendrá una propiedad llamada **url** y otra **tiene_autor**, de igual forma la clase **Persona** contendrá las propiedades **nombre**, **e-mail** y **ubicación** al igual que la clase **CertificadoDefunción** tendrá una propiedad llamada **tiene_modelo** y la clase **Nacimiento** tendrá la propiedad **tiene_parto**.

De esta forma las propiedades describen las clases a las cuales pertenecen así como a las subclases de esta última.

Paso 6. Definir las facetas de las properties

Las propiedades tienen diferentes facetas que describen el tipo de valor, valores admitidos, la cardinalidad entre otras. En diseño propuesto en esta investigación el valor de la propiedad **nombre** (el nombre de un autor de determinado servicio) es una cadena de caracteres, o sea, de tipo *string*. La propiedad **tiene_autor** de la clase **Servicio** puede tener más de un valor que en este caso serán instancias de la clase **Persona**.

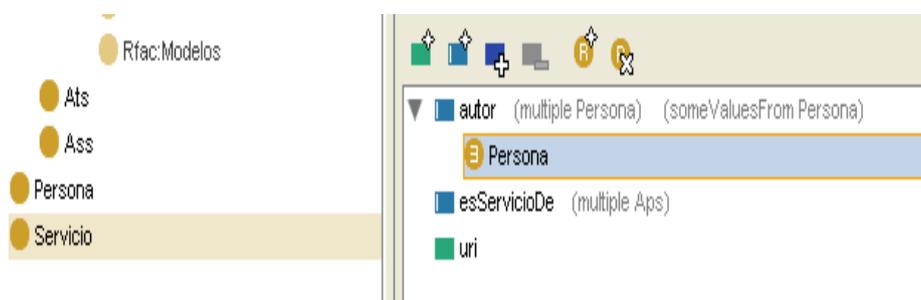


Figura 6. Propiedades de la clase Servicio.

Paso 7. Crear instancias

El último paso consiste en crear instancias de las clases de la jerarquía propuesta. Para la clase **Servicio** se creó una instancia llamada **rfac** que tiene definido los siguientes valores:

- Autor: Yanssel Urquijo Morales
- Url: <http://www.rfac.cu>
- esServicioDe: Registro_Fallecido



Figura 7. Instancia Rfac de la clase Servicio.

Nótese que “*Yanssel Urquijo Morales*” es una instancia de la clase **Persona** que contiene otros datos como **e-mail** y **ubicación**.

The screenshot shows a web-based editor for an ontology instance. At the top, it says 'INDIVIDUAL EDITOR' and '(instance of Persona)'. Below this, there are several property editors:

- nombre:** Yanssel Urquijo Morales
- ubicacion:** Universidad de las Ciencias Informaticas
- e-mail:** A table with two columns: 'Value' and 'Lang'. It contains two entries: 'yurquijo@estudiantes.uci.cu' and 'yurquijo2005@gmail.com'.

Figura 8. Instancia persona_16 de la clase Persona.

2.5 Representación del Conocimiento del Registro de Fallecidos

El sistema Registro de Fallecidos es el encargado de gestionar los certificados médicos de defunción, así como los reparos hechos a estos certificados. Cada certificado de defunción tiene un modelo determinado que corresponde a determinados grupos de edades. También tiene la facilidad de emitir un grupo de reportes estadísticos en relación con los datos obtenidos de los certificados de defunción y sus reparos.

Para el modelado de este negocio, en lo adelante llamado Rfac, se definieron un conjunto de clases y propiedades las cuales se listan a continuación.

Clases	Propiedades
Certificado_Defunción	tiene_modelo
Reparo	tiene_certificado
Modelo	grupo_edad, es_modelo_de

Tabla 5. Clases y propiedades de la ontología Rfac.

Para una mejor vista de las clases y sus relaciones Ver Anexo 2.

A esta ontología se le podrían realizar en Protégé preguntas tales como:

¿Qué modelo pertenece al certificado Fetal?



Figura 9. Consulta a la ontología Rfac.

O también ¿Qué modelo contiene "0 a 7 días" en su grupo de edades?

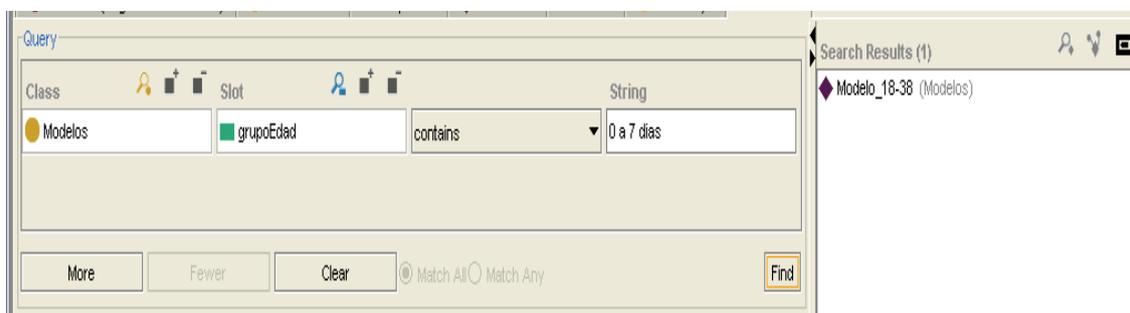


Figura 10. Consulta a la ontología Rfac.

2.6 Representación del Conocimiento del Registro de Parto y Nacimiento

El sistema Registro de Parto y Nacimiento es el encargado de gestionar la información referente al momento del parto, dígame datos propios del parto así como del o los nacidos, ya que en un mismo parto puede dar lugar a varios nacimientos. El parto puede tener lugar en una unidad hospitalaria (Institucional) o extra hospitalaria. También tiene la facilidad de emitir un grupo de reportes estadísticos en relación con los nacidos vivos y los datos recogidos en el parto.

Al igual que en el epígrafe anterior, en el proceso de creación de la ontología para este negocio, la cual se nombre RPN, se determinaron los elementos claves que pasaran a formar las clases así como sus propiedades, a continuación se relacionan las más importantes.

CAPÍTULO 2. DISEÑO DE LA ONTOLOGÍA.

Clases	Propiedades
Dato	---
Nacimiento	tiene_dato
Parto	tiene_nacimiento, tiene_dato

Tabla 6. Clases y propiedades más importantes de la ontología RPN.

Para una mejor vista de las clases y sus relaciones Ver Anexo 2.

2.7 Razonadores

A través del presente epígrafe se pretende ofrecer un visión general de los razonadores, herramientas empleadas en el trabajo con ontologías.

Un razonador es, de forma básica, una herramienta que aprovecha el gran poder semántico de las ontologías, de forma que, aplicándose sobre éstas los razonadores o máquinas de inferencia pueda extraerse conocimiento no expresado de forma explícita en su construcción. Estas herramientas son las que finalmente las que propician en gran parte el potencial de las ontologías como herramientas de gestión de la información.

A la hora de definir una ontología se encuentran tres partes principales componentes en la misma:

- Clases e instancias, que son de forma básica los objetos o elementos que la componen
- Propiedades, que representan las relaciones entre los anteriores
- Reglas, que se emplean para modelar el conocimiento que no puede recogerse a través de los elementos anteriores.

Si no se emplearan estas últimas se puede considerar como una ontología ligera, que no es más que una taxonomía o clasificación de elementos. La no utilización de reglas origina el problema de la poca capacidad expresiva del conocimiento.

En el caso particular de las ontologías, cuanto mayor sea el trabajo de definición de axiomas y relaciones, mayor será su potencial semántico. De esta forma una ontología se convierte en algo más que en un establecimiento de categorías o subcategorías de conceptos. Es en este momento cuando el empleo de razonadores cobra gran interés. Tal y como se perfila en los párrafos anteriores el empleo de razonadores está

motivado por el gran poder lógico o de deducción de conocimiento que presentan las ontologías.

De esta forma se podría decir que recogen más conocimiento del que realmente se presenta en un principio y éste puede extraerse, haciendo deducciones o respondiendo a consultas lógicas, mediante el empleo de razonadores. (26)

2.7.1 Ejemplo de razonador

Pellet es un razonador basado en Java. Unas de las funcionalidades principales de este razonador son las de: (27)

- **Comprobación de la consistencia:** Comprueba que no existen contradicciones en la ontología. La semántica de OWL define una especificación formal para la definición de la consistencia en una ontología mediante el uso Pellet.
- **Corrección de los conceptos:** Verifica si es posible que se definan instancias para una clase; si no es así, definir una instancia para una clase provocaría que toda la ontología fuera inconsistente.
- **Clasificación:** Observa la relación entre cada clase y crea la jerarquía de clases completa.
- **Realización:** Encuentra las clases más específicas a las que pertenece una instancia; en otras palabras, determina la clase a la que pertenece cada uno de los individuos.

Conclusiones

En este capítulo se presentaron los pasos seguidos para la construcción de las ontologías. Así como una vista detallada de cada una, donde se explican los términos empleados en la representación del conocimiento que se modela. Se expone una muestra de las preguntas que se le pueden hacer y cómo responden estas. Además se dan ejemplos de la importancia de los razonadores para incrementar el conocimiento y la consistencia de las ontologías.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL AGENTE

El presente capítulo trata los temas relacionados con la arquitectura, el análisis y el diseño del agente propuesto. Se exponen además, los diagramas de clases del análisis, los diagramas de clases del diseño y los diagramas de interacción tanto del análisis como del diseño.

3.1 Modelo de Dominio

Un sistema, por pequeño que sea, generalmente es complicado. Por lo que se hace necesario dividirlo en piezas si se pretende comprender y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales.

Una técnica para la especificación de los requisitos más importantes del sistema, que da soporte al negocio, es el modelo del negocio, con lo cual se refuerza la idea de que sea el propio negocio lo que determine los requisitos. Como un subconjunto del modelo de negocio se encuentra el modelo de dominio o modelo conceptual el cual se realiza cuando no están bien definidos los procesos del negocio.

La presente investigación se centra en la creación de un agente semántico inteligente para la salud, basado en ontologías. En este, solo se representará el conocimiento de las aplicaciones informáticas del nivel de APS. Lo cual da solución solo a una fracción del proceso, por lo que debe realizarse un modelo de dominio, que permita obtener conceptos sobre la realidad física donde se enmarca la solución propuesta.

El modelo de dominio es la representación de conceptos de importancia, constituido por reglas de modelado, no por componentes propios de un sistema de software. Es posible realizar un elevado grado de detalles en un modelo de dominio, ya que tiene como objetivo fundamental capturar lo necesario para comprender el funcionamiento del sistema; bien sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir.

3.1.1 Conceptos fundamentales

A continuación se proporciona un marco conceptual donde se identifican los conceptos fundamentales dentro del dominio; que permitirán una mejor comprensión del diagrama del Modelo de Dominio que será presentado en el próximo epígrafe.

Usuario: Todo aquel usuario de INFOMED que solicite información sobre las aplicaciones software que informatizan un determinado servicio.

Administrador: Persona encargada de administrar la ontología OSalud, incorpora nuevo conocimiento.

Ingeniero del conocimiento: Persona especialista en Ciencias de la Información encargada de representar el conocimiento de los distintos negocios.

Base de conocimiento: Está constituida por las distintas ontologías que describen los diferentes negocios que se representan.

Agente: Sistema que se encargará de interactuar con la base de conocimiento para gestionar las preguntas del usuario.

3.1.2 Diagrama del Modelo de Dominio

El diagrama del Modelo de Dominio es un diagrama de clases de UML que permite representar de manera visual los conceptos fundamentales que forman parte del dominio del problema, así como la relación existente entre ellos.

Diagrama del Modelo de Dominio

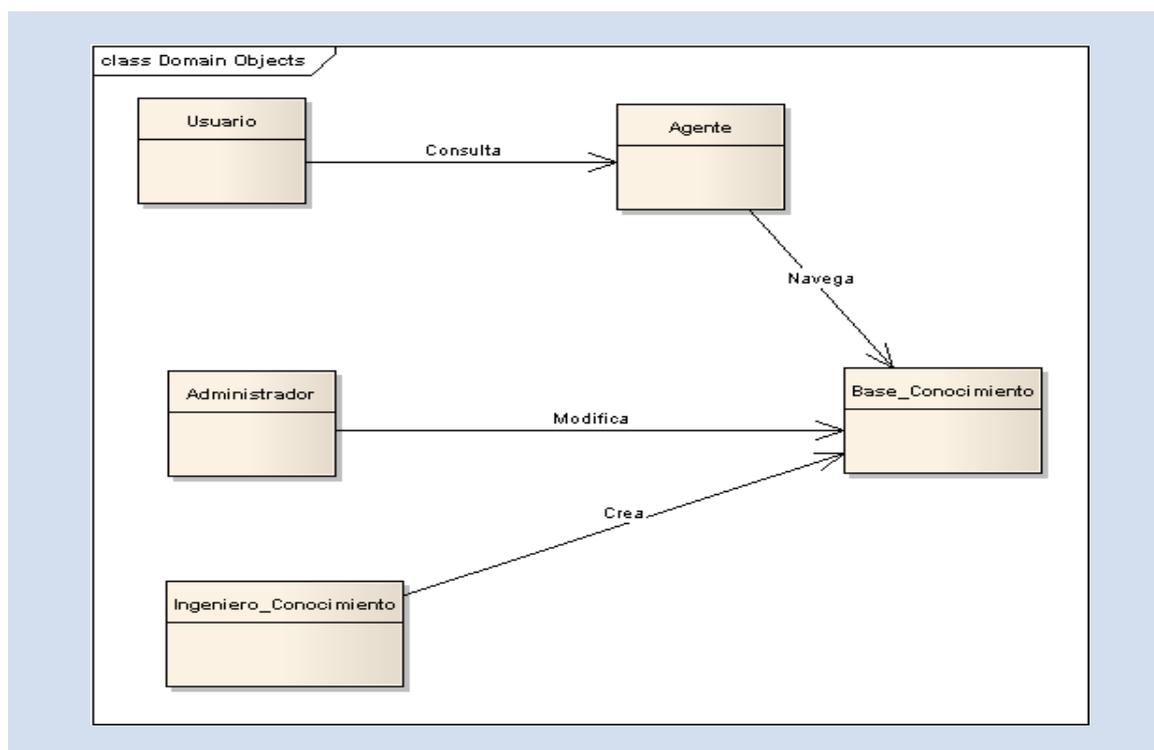


Figura 11. Modelo de Dominio.

3.2 Propuesta del sistema.

3.2.1 Especificación de requerimientos de software

El modelo de dominio brinda una vía para determinar los requerimientos del sistema. Un requerimiento es una necesidad documentada sobre el contenido, forma o

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL AGENTE.

funcionalidad de un producto o servicio. Se usa en un sentido formal en la ingeniería de sistemas o la ingeniería de software.

Todas las ideas que los clientes, usuarios o miembros del equipo de desarrollo tengan acerca de lo que debe hacer el sistema, deben ser analizados como candidatos a requerimientos. Estos se utilizan como datos de entrada en la etapa de diseño del producto; establecen **QUÉ** debe hacer el sistema, pero **NO CÓMO** hacerlo. Los requerimientos bien formulados deben satisfacer varias características. Si no lo hacen, deben ser reformulados hasta hacerlo. Los requerimientos pueden ser clasificados en funcionales y no funcionales.

3.2.2 Requerimientos funcionales

Un requerimiento funcional puede ser una descripción de lo que un sistema debe hacer. Este tipo de requerimiento especifica algo que el sistema entregado debe ser capaz de realizar. Para el sistema propuesto se determinaron los siguientes requisitos funcionales:

RF1. Agregar ontología: El sistema debe incorporar una ontología a la base de conocimiento que exista hasta el momento.

RF2. Evaluar consistencia de la Ontología: Poder evaluar el estado de las relaciones entre las clases e instancias de las ontologías, de la base de conocimiento.

RF3. Inferir conocimiento: Poder inferir nuevas relaciones entre las clases e instancias a partir de las ontologías existentes.

RF4. Consultar ontología: Poder dar respuesta a las preguntas realizadas por los usuarios.

3.2.3 Requerimientos no funcionales

Un requerimiento no funcional especifica cómo debe realizar sus funciones. Normalmente están vinculados a requisitos funcionales, una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser. Algunos ejemplos de aspectos son la disponibilidad, la prueba, el mantenimiento, la facilidad de uso, entre otros. El levantamiento de requerimientos para el sistema propuesto arrojó como requisitos no funcionales los siguientes:

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL AGENTE.

Usabilidad: estos requerimientos describen los niveles apropiados de usabilidad, dado los usuarios finales del producto, para ello debe tenerse en cuenta las especificaciones de los perfiles de usuarios.

RNF1 El usuario que se autentique como administrador del sistema tendrá acceso a la configuración, ya sea de idioma, base de conocimiento, etc.

Soporte: abarca todas las acciones a tomar una vez que se ha terminado el desarrollo del sistema.

RNF2 El personal que trabaja en la configuración del sistema, entiéndase modificar la base de conocimiento, debe contar con el nivel técnico requerido mediante adiestramiento de servicio.

Portabilidad: este tipo de requerimiento especifica los atributos que debe presentar el sistema para facilitar su montaje en diferentes plataformas o entornos.

RNF 3 Permitir que el sistema se ejecute sobre Sistema Operativo Windows 98 o superior o Linux.

Seguridad: este es uno de los requerimientos más importantes.

RNF 4 Seguridad de acceso: Asignación de perfiles.

RNF 5 Velar por la integridad de los datos de la base de conocimiento, a través de la consistencia de la información.

Rendimiento: especifican los requerimientos relacionados con la carga que se espera que deba soportar el sistema. Con el fin de aumentar este rendimiento se debe tener en cuenta lo siguiente:

RNF 6 El sistema debe soportar un tiempo de respuesta menor o igual a 20 segundos.

RNF 7 El sistema debe soportar una conexión simultánea alrededor de 500 usuarios.

Software: describen los programas con los que se debe contar para que el sistema funcione correctamente.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL AGENTE.

RNF 8 Los clientes tendrán acceso al Agente Semántico Inteligente para la Salud a través de cualquier navegador web, recomendado Mozilla Firefox 1.5 e Internet Explorer 5.0 o superior.

Hardware: El hardware que será empleado en el despliegue del módulo juega un papel importante en la construcción del mismo. Para el despliegue del sistema propuesto se identifican los siguientes requerimientos de hardware.

RNF 9 El servidor donde se encontrará montado el sistema requiere de buenas prestaciones en procesador, memoria y redes.

RNF 9.1 Procesador: Mínimo Intel P4 o superior o equivalente

RNF 9.2 Frecuencia: preferentemente 3GHz o superior

RNF 9.3 Memoria: Preferentemente 2 GB de DIMM DDR2 800 MHz.

RNF 9.4 Almacenamiento: Mínimo 80 GB de capacidad, SATA II o SCSI.

RNF 9.5 Red: Soporte para 512Mbps Ethernet o superior.

Además se debe tener en cuenta que las máquinas clientes las siguientes especificaciones:

RNF 10 PC Cliente: el sistema no hacen un uso intensivo de los recursos en el cliente por tal razón entre los requerimientos se incluyen:

RNF 10.1 Procesador de 800mHz de velocidad o superior.

RNF 10.2 Memoria RAM 256mb o superior.

3.3 Definición de casos de uso

La arquitectura de un sistema está concretada entre otras cosas por los casos de uso. Los casos de uso definen la manera en que se comportará el sistema.

Un actor es algo que interactúa con el sistema, puede definirse como cualquier persona, individuo, grupo, entidad, organización, máquina o sistema de información externo (identificados por un rol); que tiene objetivos o necesidades y utiliza el sistema para beneficiarse de sus resultados. Todo caso de uso es iniciado por un actor del sistema; en la fase de Inicio del desarrollo del sistema se detectaron los siguientes actores:

3.3.1 Definición de actores del sistema

El sistema interactúa con dos actores, los cuales se muestran a continuación.

Actores del Sistema	Justificación
<p>The diagram shows a UML Actor symbol (a stick figure with a yellow circle head) inside a rectangular box. The box has a tab at the top left labeled 'uc Actors'. Below the actor, the text 'Usuario INFOMED' is written.</p>	<p>Representa todo usuario de INFOMED que interactúa con el agente.</p> <p>Inicia el caso de uso Consultar Ontología.</p>
<p>The diagram shows a UML Actor symbol (a stick figure with a yellow circle head) inside a rectangular box. The box has a tab at the top left labeled 'uc Actors'. Below the actor, the text 'Administrador' is written.</p>	<p>Es la persona con privilegio suficiente para modificar la ontología, incorpora nuevo conocimiento.</p> <p>Inicia el caso de uso Agregar Ontología.</p>

Tabla 7. Actores del Sistema.

3.3.2 Listado de casos de uso del sistema

Los casos de uso que se identificaron en la confección del agente son los siguientes:

CU-1	Consultar Ontología
Actor	Usuario INFOMED
Descripción	El usuario de INFOMED realiza una consulta determinada mediante el agente.

Tabla 8. CU Consultar Ontología.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL AGENTE.

CU-2	Agregar Ontología
Actor	Administrador
Descripción	El administrador agrega nuevo conocimiento a la ontología OSalud.

Tabla 9. CU Agregar Ontología.

CU-3	Evaluar Consistencia
Actor	Administrador
Descripción	El administrador después de agregar nuevo conocimiento a la ontología OSalud puede chequear su consistencia.

Tabla 10. CU Evaluar Consistencia.

CU-4	Inferir Conocimiento
Actor	Administrador
Descripción	El administrador después de agregar nuevo conocimiento a la ontología OSalud y chequear su consistencia, infiere nuevo conocimiento a partir del ya existente.

Tabla 11. CU Inferir Conocimiento.

3.3.3 Diagramas de casos de uso del sistema

A continuación se presenta el diagrama de caso de uso del sistema.

Diagramas de casos de uso del sistema.

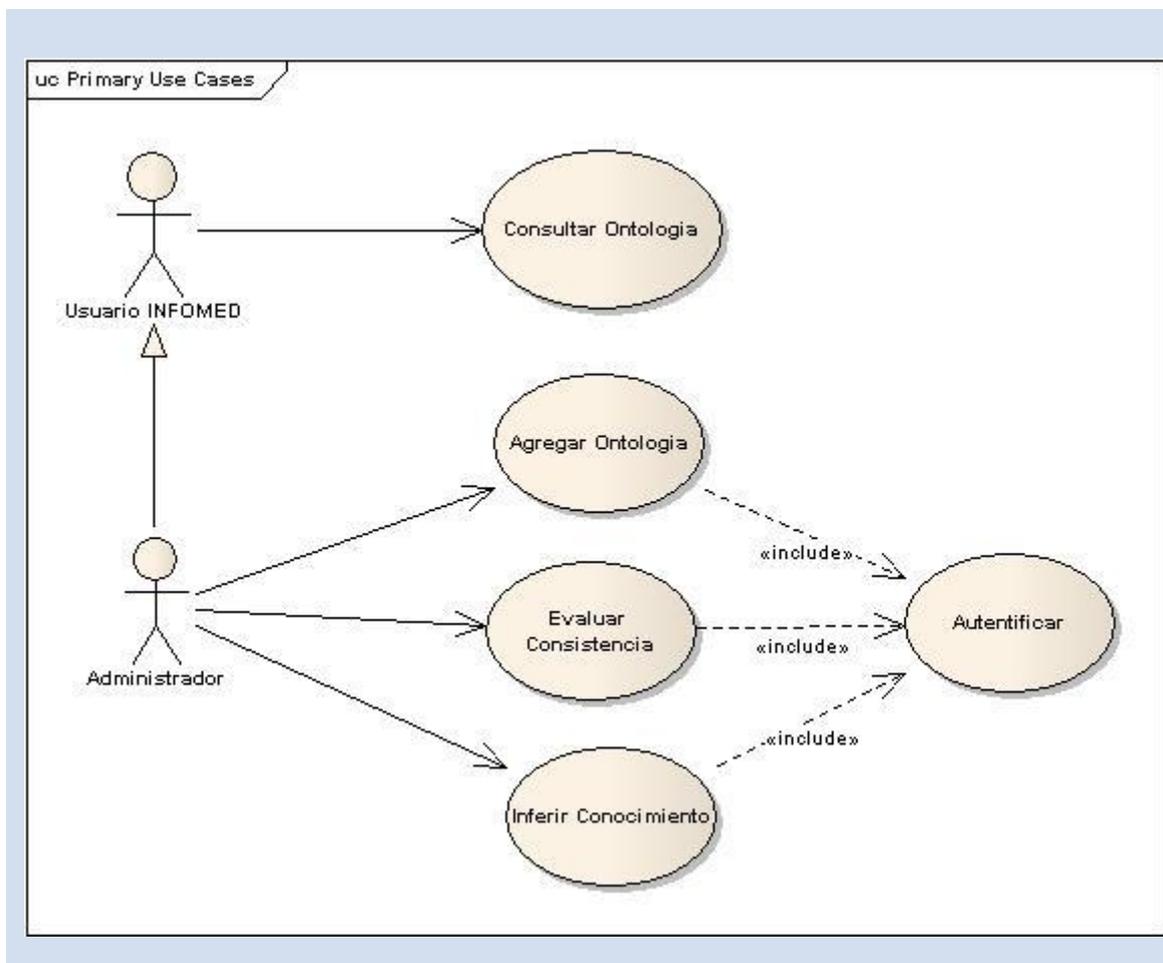


Figura 12. Diagrama de CU del Sistema.

3.3.4 Especificación de los casos de uso

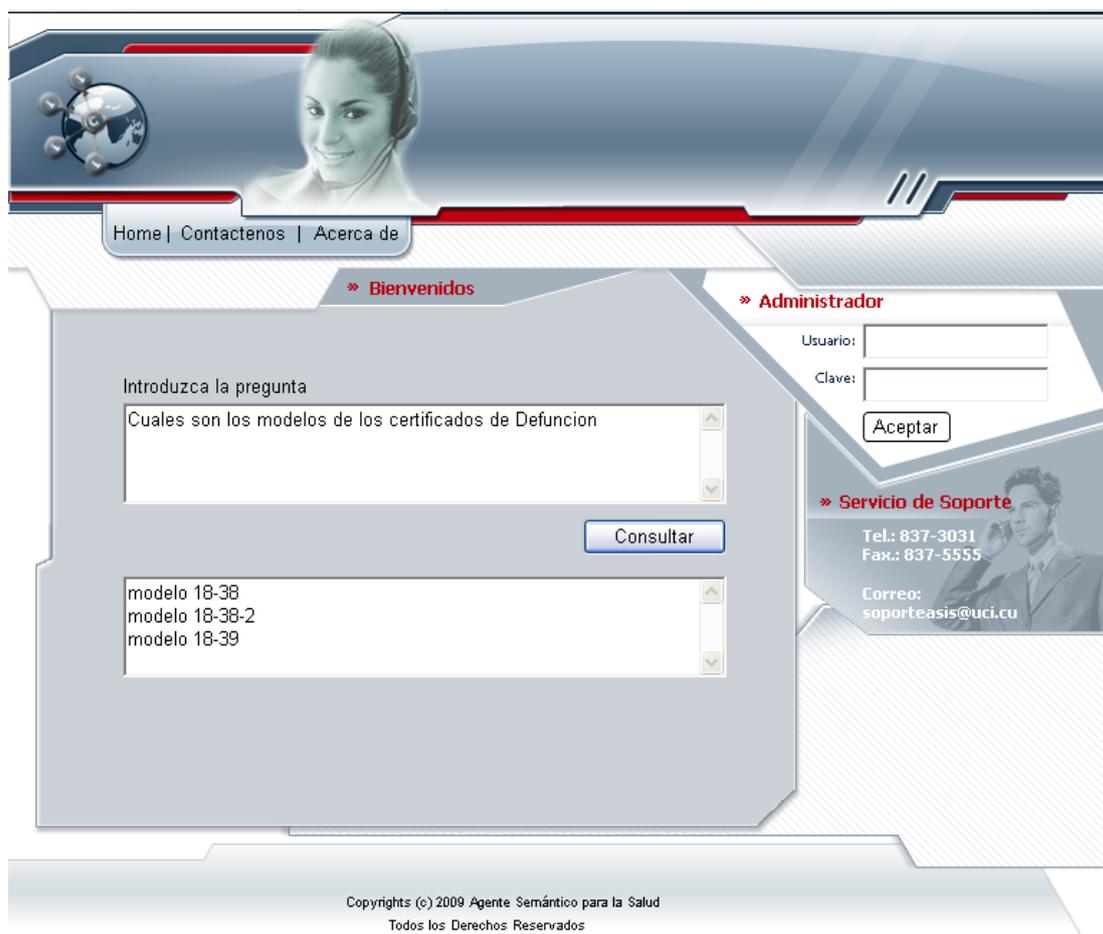
En el siguiente epígrafe se muestra la descripción detallada de los casos de usos del sistema.

CASO DE USO:	Consultar Ontología
Resumen:	El caso de uso inicia cuando el Usuario introduce en el sistema una pregunta en lenguaje natural, luego el sistema envía al analizador sintáctico estos datos. Luego de extraer del lenguaje natural las expresiones necesarias para formular la consulta en SPARQL, estas son enviadas al <i>framework</i> Jena, donde se

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL AGENTE.

	elabora la consulta y se recoge la respuesta obtenida. El caso de uso finaliza cuando la respuesta a la búsqueda realizada es mostrada al usuario.	
Complejidad:	Alta	
Prioridad:	Primaria	
Precondiciones:	Que se tenga una base de conocimiento.	
REFERENCIAS		
Actores:	Usuario INFOMED (Inicia)	
Referencias	RF4	
Entidades:	Ontologías	
Casos de Uso:		
Poscondiciones:	Se obtienen los datos de respuesta a la pregunta formulada.	
FLUJO NORMAL DE EVENTOS		
Acción del Actor	Respuesta del Sistema	
1. El usuario introduce una pregunta en lenguaje natural.	2. El sistema le envía al analizador sintáctico la expresión en lenguaje natural introducida por el usuario. 3. El sistema recibe del analizador sintáctico, la información necesaria para construir la consulta. 4. Se construye la consulta en SPARQL. 5. Muestra el resultado de la búsqueda realizada. <ul style="list-style-type: none"> • Si no encuentra el resultado ver flujo alterno 1. 	

- 6. El usuario visualiza el resultado.
- 7. El caso de uso termina.



Flujos Alternos

Flujo Alterno 1

Acción del Actor

Respuesta del Sistema

5.1 El sistema informa que no se encontró ningún resultado.

Prototipo de Interfaz

Tabla 12. Descripción del CUS Consultar Ontología.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL AGENTE.

CASO DE USO:	Agregar Ontología	
Resumen:	Inicia cuando el Administrador desea incorporar una nueva ontología a la base de conocimiento que existe hasta el momento.	
Complejidad:	Baja	
Prioridad:	Primaria	
Precondiciones:	<ol style="list-style-type: none"> 1. Debe tenerse una nueva representación del conocimiento de alguno de los servicios informáticos del nivel de APS. 2. Debe estar autenticado con privilegios de administrador. 	
REFERENCIAS		
Actores:	Administrador	
Referencias	RF1	
Entidades:	Ontologías	
Casos de Uso:	Autenticar	
Poscondiciones:	Se aumenta la base de conocimientos.	
FLUJO NORMAL DE EVENTOS		
Acción del Actor	Respuesta del Sistema	
1. El administrador se autentica en el sistema.	2. El sistema valida los datos introducidos. <ul style="list-style-type: none"> • Si los datos introducidos no son válidos ver flujo alterno 1. 3. Muestra interfaz y permite acceder al módulo de configuración.	
4. Selecciona agregar ontología.	5. Permite introducir la URI donde se encuentra y el nivel al cual pertenece.	

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL AGENTE.

6. Inserta datos de la ontología.

7. El sistema valida la URI introducida.
- Si detecta alguna dificultad ver flujo alternativo 2.
8. Incorpora a la base de conocimiento la nueva ontología.
9. Los datos son registrados y se muestra un mensaje donde se informa el éxito de la operación.

Flujos Alternos

Flujo Alterno 1

Acción del Actor

Respuesta del Sistema

2.1 El sistema muestra mensaje donde se solicita introducir los datos correctos.

Flujo Alterno 2

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL AGENTE.

Acción del Actor	Respuesta del Sistema
	7.1 El sistema muestra mensaje donde se solicita introducir los datos correctos.
Prototipo de Interfaz	

Tabla 13. Descripción del CUS Agregar Ontología.

3.4 Patrones de Arquitectura a utilizar

La arquitectura propuesta está separada en tres capas para aprovechar así las ventajas de la centralización de los aspectos de seguridad y transaccionalidad. Así como, la no replicación de la lógica de negocio, además de ser una arquitectura que les brinda mayor sencillez a los clientes. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido.

Está basada en componentes para garantizar la modularidad, la reusabilidad y compatibilidad de la lógica de negocio. Además es sustentada en un modelo cliente-servidor lo cual permite que la capacidad de proceso esté repartida entre los clientes y los servidores. También reduce los costos de producción de software y el costo del hardware requerido, lo que permite llevar las aplicaciones a plataformas más baratas, donde se aprovecha el poder de cómputo de los diferentes elementos de la red.

También usa estándares abiertos como SOAP (siglas de *Simple Object Access Protocol*) y XML para garantizar interoperabilidad con sistemas externos y legados (Ver Anexo 3).

3.5 Análisis

El análisis posibilita un conocimiento razonable del sistema o producto. La intención que se percibe es definir todas las clases que son relevantes al problema que se va a resolver, las operaciones y atributos asociados, así como sus relaciones y comportamientos. El análisis suaviza la transición al Diseño.

Para tener una vista gráfica de la representación del modelo de análisis se puede consultar los diagramas de clases del análisis para cada uno de los casos de uso (ver

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL AGENTE.

figuras 13 y 14). Se puede consultar además los diagramas de interacción de cada uno de los casos de uso (Ver Anexo 4).

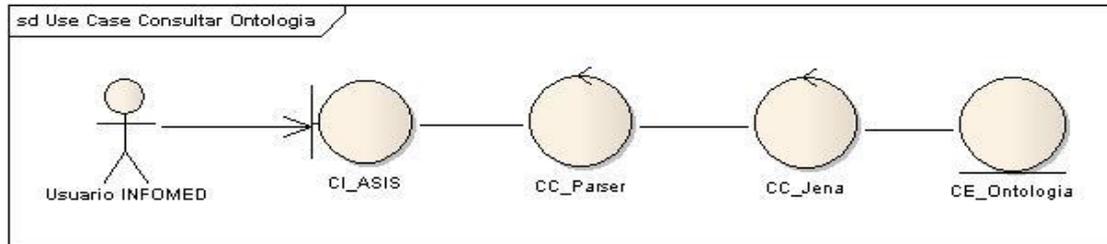


Figura 13. Diagramas de clase del análisis, caso de uso consultar ontología.

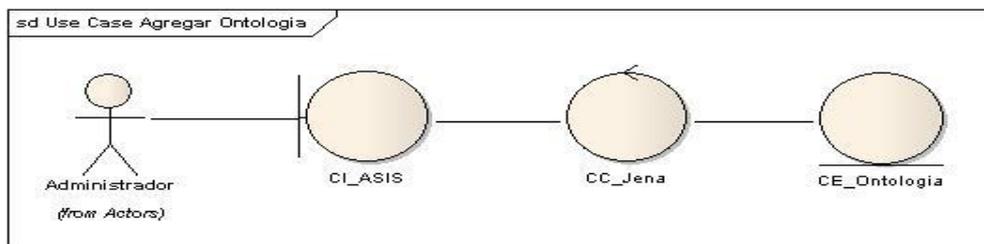


Figura 14. Diagramas de clase del análisis, caso de uso agregar ontología.

3.6 Diseño

El Modelo de Diseño es otro de los tantos modelos que se tienen en cuenta en el ciclo de vida del desarrollo de un software, consumiendo una elevada cantidad de esfuerzo para su realización. El diseño es el centro de atención al final de la fase de elaboración y el comienzo de la etapa de construcción.

Este modelo es de vital importancia para los especialistas y desarrolladores ya que sirve de abstracción de la implementación y es utilizado fundamentalmente como esbozo para la construcción del software. En él se identifican las clases que modelarán el problema, las interfaces y jerarquía de herencia, además de establecer las relaciones claves entre las mismas.

El propósito del diseño es formar la base para la implementación, desarrollar la arquitectura, dándole soporte a todos los requisitos, incluir los no funcionales y las restricciones que se le suponen, además de descomponer los trabajos a desarrollar en partes más manejables que puedan ser llevadas a cabo por diferentes equipos de desarrollo, sin descuidar la posible concurrencia.

CAPÍTULO 3. ANÁLISIS Y DISEÑO DEL AGENTE.

Para un mayor entendimiento se presentan los diagramas de clases de diseño, donde se describen gráficamente las especificaciones de las clases de software que intervienen en cada caso de uso (ver figuras 15 y 16), así como los diagramas de interacción correspondientes a cada uno de estos (Ver Anexo 5).

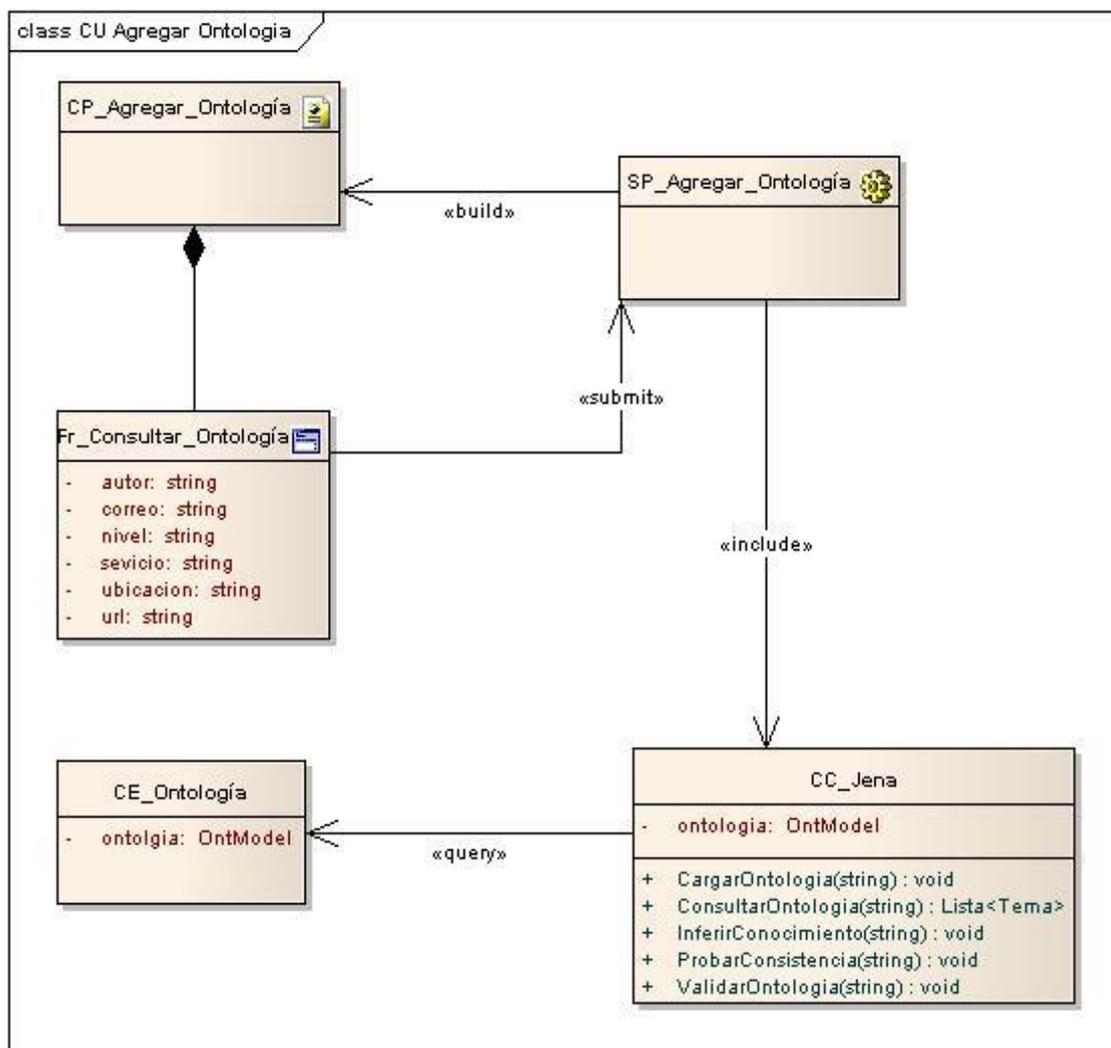


Figura 15. Diagrama de clase del diseño, caso de uso agregar ontología.

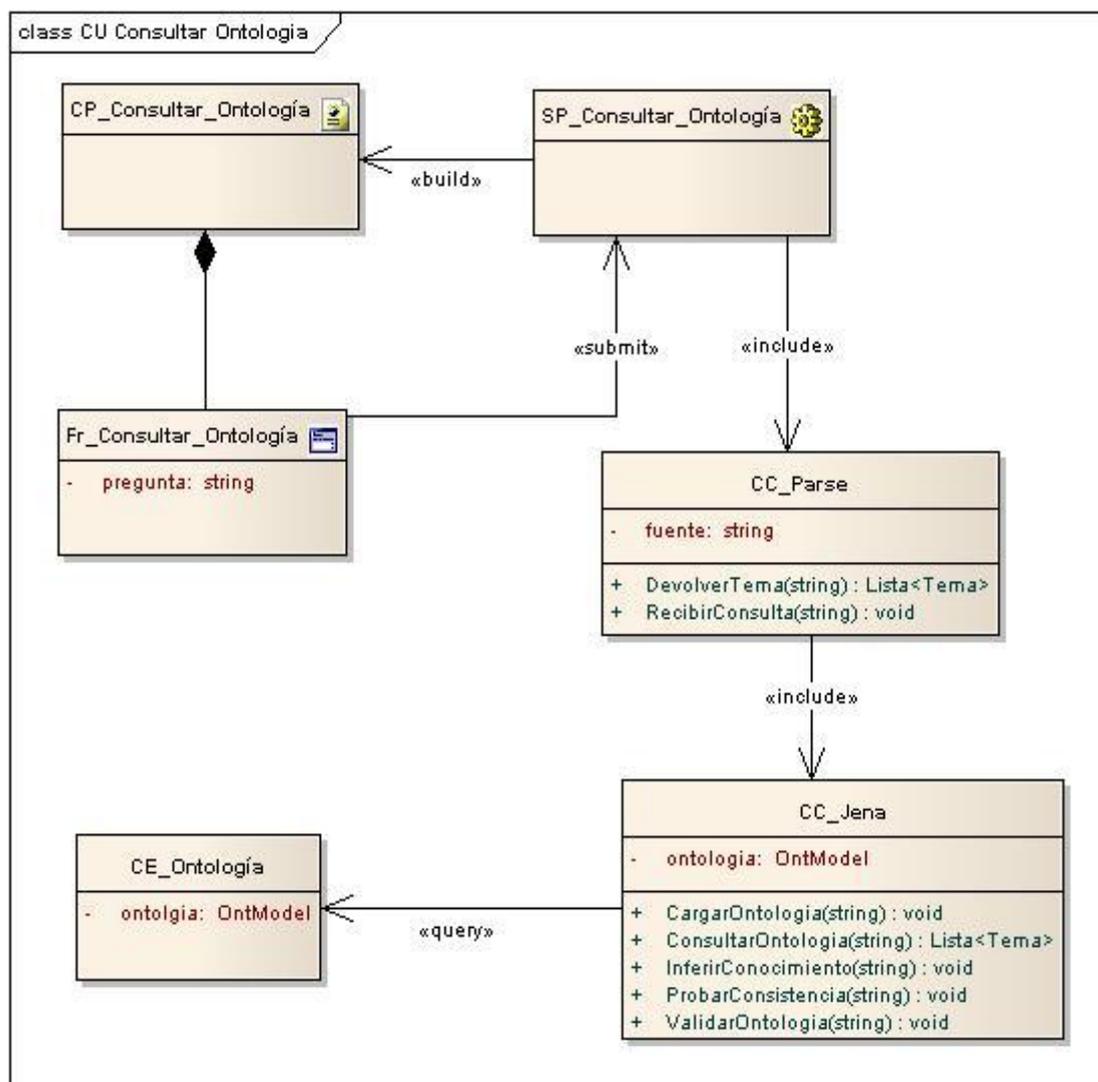


Figura 16. Diagrama de clase del diseño, caso de uso consultar ontología.

Conclusiones

En este capítulo se presentó el modelo de domino, se elaboraron los diagramas de casos de uso del sistema y la especificación de estos. Se mostraron los elementos referentes a las etapas del flujo de trabajo de análisis y diseño: la descripción de la arquitectura, el modelo de clases del análisis y del diseño. Así como los respectivos diagramas de interacción lo que permitió tener una idea más precisa de los elementos del agente propuesto.

CONCLUSIONES GENERALES

Con la realización del presente trabajo de diploma se arribaron a las siguientes conclusiones:

Las soluciones informáticas existentes para la gestión de información basada en ontologías, a pesar de no constituir una solución al problema a resolver, sirvieron como fuente de conocimiento para el desarrollo del sistema.

La utilización de ontologías para la representación del conocimiento de los sistemas informáticos alojados en INFOMED permitirá consultar estos temas de manera semántica sin la necesidad de entrevistarse con un especialista.

La ontología resultante engloba el conocimiento sobre el dominio que abordan los diferentes sistemas informáticos desarrollados para la atención al paciente, los cuales se encuentran separados por los distintos niveles en los que se encuentra estructurado el Sistema Nacional de Salud en Cuba.

El análisis y diseño del agente semántico ofrece una visión general de la estructura y componentes a tener en cuenta en la futura implementación del agente.

RECOMENDACIONES

Por las experiencias alcanzadas durante la realización de este trabajo y con vista a enriquecer la solución propuesta, el autor sugiere:

- Representar el conocimiento de los diferentes sistemas informáticos que alojados en INFOMED vinculados con la Atención Primaria, Secundaria o Terciaria de la Salud con el fin de incrementar la base de conocimiento propuesta.
- Realizar un estudio detallado de las técnicas de inferencia de conocimiento sobre la ontología OSalud.
- Implementar el agente semántico que permita la consultar la información disponible en la ontología OSalud.

REFERENCIAS BIBLIOGRÁFICAS

1. INFOMED, Portal de Salud de Cuba. [En línea] <http://www.sld.cu/>.
2. Localizador de Información en Salud. [En línea] [http://bvs.sld.cu/xml2html/xmlListT.php?xml\[\]=lis-Regional/E/define.xml&xml\[\]=lis-Regional/E/defineContent.xml&xsl=lis-Regional/about.xsl](http://bvs.sld.cu/xml2html/xmlListT.php?xml[]=lis-Regional/E/define.xml&xml[]=lis-Regional/E/defineContent.xml&xsl=lis-Regional/about.xsl).
3. **Consortium, World Wide Web.** World Wide Web Consortium. [En línea] 2009. <http://www.w3c.es>.
4. World Wide Web Consortium. [En línea] <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>.
5. W3C Candidate Recommendation. [En línea] 27 de 03 de 2000. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>.
6. Ontoknowledge. [En línea] <http://www.ontoknowledge.org/oil/>.
7. **van Harmelen, Frank, F. Patel-Schneider, Peter y Horrocks, Ian.** The DARPA Agent Markup Language Homepage. [En línea] 03 de 2001. <http://www.daml.org/2001/03/daml+oil-walkthru>.
8. World Wide Web Consortium. [En línea] <http://www.w3.org/TR/owl-features/>.
9. **McGuinness, D y van Harmelen, F.** World Wide Web Consortium. [En línea] <http://www.w3.org/TR/2003/WD-owl-features-20030331/>.
10. **Fernández López, M, Gómez Pérez, A y Rojas Amaya, M D.** *Ontologies crossed life*. Francia : Springer Verlag. Jean Les Pins, 2000.
11. **Staab, S, y otros.** *Knowledge Processes and Ontologies*. s.l. : Intelligent Systems, 2001.
12. CiteSeer. [En línea] <http://citeseer.ist.psu.edu/old/fox97ontologies.html>.
13. **F. Noy, Natalya y L. McGuinness, Deborah.** *Desarrollo de Ontologías 101*. California, EEUU : s.n., 2005.
14. **Russel, Stuart y Norvig, Peter.** *Inteligencia Artificial. Un enfoque moderno*. s.l. : Pearson Prentice Hall, 2004.
15. **Wooldridge, M y Jennings N., R.** *Intelligent Agents: theory an practice. The Knowledge Engineering Review*. 1995.
16. Wikipedia. [En línea] http://es.wikipedia.org/wiki/Analizador_sint%C3%A1ctico.
17. Semantic Web Annotation and Authoring . [En línea] <http://annotation.semanticweb.org/>.
18. Ontoknowledge. [En línea] <http://www.ontoknowledge.org/tools/ontoedit.shtml>.
19. Protégé. [En línea] <http://protege.stanford.edu/>.

REFERENCIAS BIBLIOGRÁFICAS

20. Protege. [En línea] <http://protege.stanford.edu/plugins/>.
21. Ontotext. [En línea] <http://www.ontotext.com/kim/>.
22. Sourceforge.net. [En línea] <http://sourceforge.net/projects/sesame>.
23. Ontotext. [En línea] <http://www.ontotext.com>.
24. Sourceforge.net. [En línea] <http://jena.sourceforge.net>.
25. **Basado en la documentación online para RUP, proporcionada por Rational Corporation.**
The Rational Unified Process Product.
26. Information Engineering. [En línea] <http://163.117.147.74/ie/doku.php?id=razonadores>.
27. Information Engineering. [En línea]
http://163.117.147.74/ie/doku.php?id=ejemplos_de_razonadores.

BIBLIOGRAFÍA

Basado en la documentación online para RUP, proporcionada por Rational Corporation. *The Rational Unified Process Product.*

CiteSeer. [Online] <http://citeseer.ist.psu.edu/old/fox97ontologies.html>.

Consortium, World Wide Web. World Wide Web Consortium. [En línea] 2009. <http://www.w3c.es>.

Daml. [Online] <http://www.daml.org/ontologies/>.

Dinos Rojas, Juan Larry. *Tesis de Maestría: Arquitectura de un sistema basado en agentes para la recuperación de metadatos RDF en base a una ontología de documentos.* Mayagüez, Puerto Rico : s.n., 2004.

Dmoz. [Online] <http://www.dmoz.org>.

F. Noy, Natalya and L. McGuinness, Deborah. *Desarrollo de Ontologías 101.* California, EEUU : s.n., 2005.

Fernández López, M, Gómez Pérez, A and Rojas Amaya, M D. *Ontologies crossed life.* Francia : Springer Verlag. Jean Les Pins, 2000.

García Sánchez, Francisco. *Tesis Doctoral: Sistema Basado en tecnologías del Conocimiento para Entornos de Servicios Web Semánticos.* Murcia, España : s.n., 2007.

Hurtado Bustamante, Diana Paola and Sequeda Sanclemente, Juan Federico. *Propuesta del Uso de Ontologías para la Búsqueda Semántica en Laboratorios de Investigación y desarrollo: OLID.* Cali, Colombia : s.n.

INFOMED, Portal de Salud de Cuba. [Online] <http://www.sld.cu/>.

Information Engineering. [Online] http://163.117.147.74/ie/doku.php?id=ejemplos_de_razonadores.

Information Engineering. [Online] <http://163.117.147.74/ie/doku.php?id=razonadores>.

Localizador de Información en Salud. [Online] [http://bvs.sld.cu/xml2html/xmlListT.php?xml\[\]=lis-Regional/E/define.xml&xml\[\]=lis-Regional/E/defineContent.xml&xsl=lis-Regional/about.xsl](http://bvs.sld.cu/xml2html/xmlListT.php?xml[]=lis-Regional/E/define.xml&xml[]=lis-Regional/E/defineContent.xml&xsl=lis-Regional/about.xsl).

McGuinness, D and van Harmelen, F. World Wide Web Consortium. [Online] <http://www.w3.org/TR/2003/WD-owl-features-20030331/>.

Navarro Galindo, José L. and Samos Jiménez, José. *Una panorámica actual de software para el trabajo con ontologías.* Granada, España : s.n.

Ontoknowledge. [Online] <http://www.ontoknowledge.org/oil/>.

Ontoknowledge. [Online] <http://www.ontoknowledge.org/tools/ontoedit.shtml>.

Ontolingua. [Online] <http://www.ksl.stanford.edu/software/ontolingua/>.

Ontotext. [Online] <http://www.ontotext.com>.

Ontotext. [Online] <http://www.ontotext.com/kim/>.

Pérez Romero, Yoenny and Sánchez Corales, Yovannys. *Tesis: Registro de Partos y Nacimientos para el Sistema Integral de Salud.* Ciudad de La Habana, Cuba : s.n., 2005.

Protégé. [Online] <http://protege.stanford.edu/>.

Protege. [Online] <http://protege.stanford.edu/plugins/>.

Renté Vázquez, Yunaisi and García de La Paz, Alain. *Tesis: Análisis del Registro de Fallecidos para el Sistema de Información para la Salud.* Ciudad de La Habana, Cuba : s.n., 2007.

Rosettanet. [Online] <http://www.rosettanet.org>.

Russel, Stuart and Norvig, Peter. *Inteligencia Artificial. Un enfoque moderno.* s.l. : Pearson Prentice Hall, 2004.

Samper Zapater, José Javier. *Tesis Doctoral: Ontologías para Servicios Web Semántico de Información de Tráfico.* Valencia, España : s.n., 2005.

Semantic Web Annotation and Authoring . [Online] <http://annotation.semanticweb.org/>.

Sorceforge.net. [Online] <http://jena.sorceforge.net>.

Sourceforge.net. [Online] <http://sourceforge.net/projects/sesame>.

Staab, S, et al. *Knowledge Processes and Ontologies.* s.l. : Intelligent Systems, 2001.

Unspsc. [Online] <http://www.unspsc.org>.

Valencia García, Rafael. *Tesis Doctoral. Un Entorno para la Extracción Incremental.* Murcia, España : s.n., 2005.

van Harmelen, Frank, F. Patel-Schneider, Peter and Horrocks, Ian. The DARPA Agent Markup Language Homepage. [Online] 03 2001. <http://www.daml.org/2001/03/daml+oil-walkthru>.

W3C Candidate Recommendation. [Online] 03 27, 2000. <http://www.w3.org/TR/2000/CR-rdf-schema-20000327/>.

Wikipedia. [Online] http://es.wikipedia.org/wiki/Analizador_sint%C3%A1ctico.

Wooldridge, M and Jennings N., R. *Intelligent Agents: theory an practice. The Knowledge Engineering Review.* 1995.

World Wide Web Consortium. [Online] <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>.

World Wide Web Consortium. [Online] <http://www.w3.org/TR/owl-features/>.

ANEXOS

Anexo 1. Vista general de la ontología.

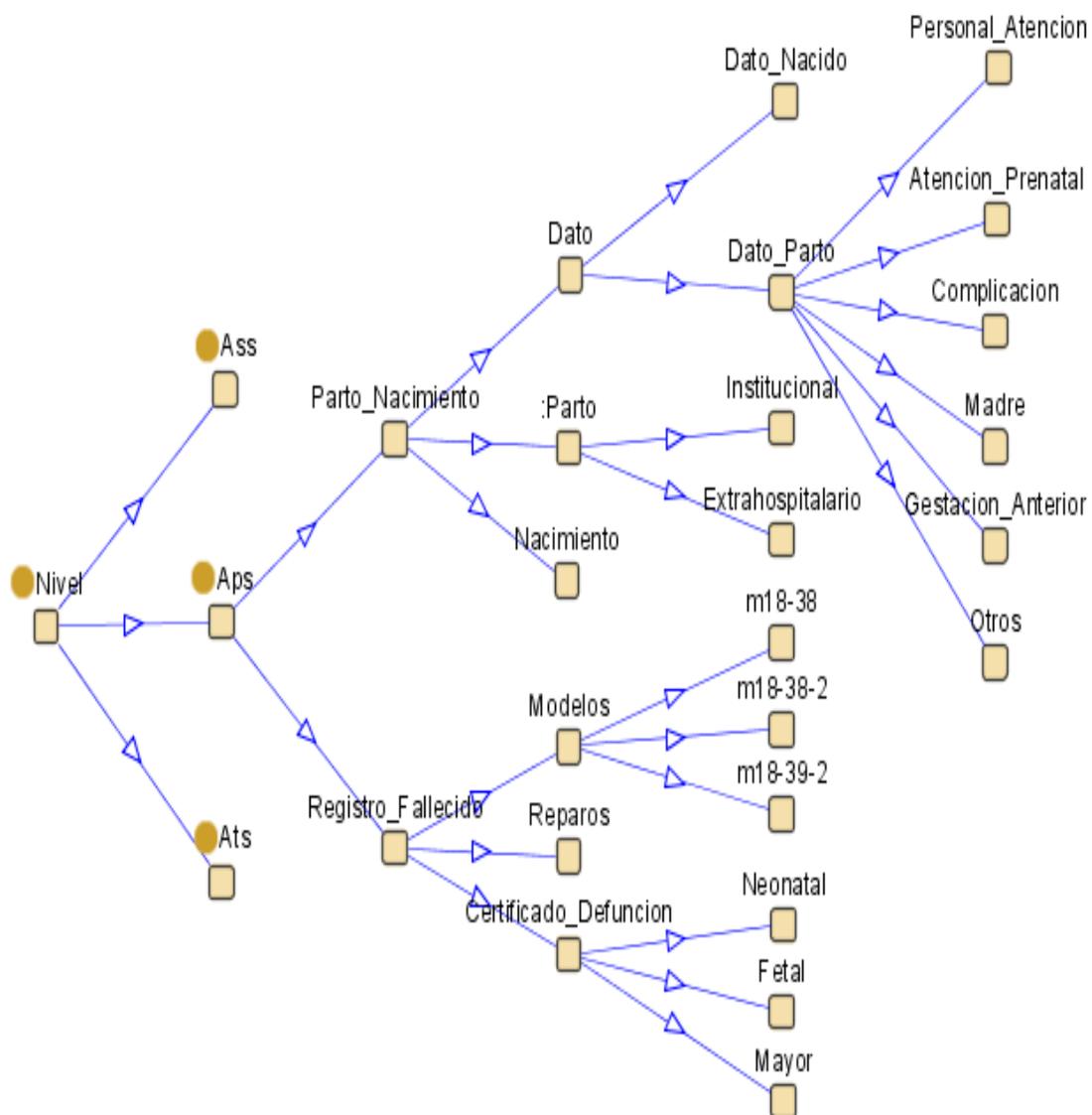


Figura 17. Jerarquía de clases de la ontología OSalud.

Anexo 2. Relaciones entre clases e instancias de las ontologías.

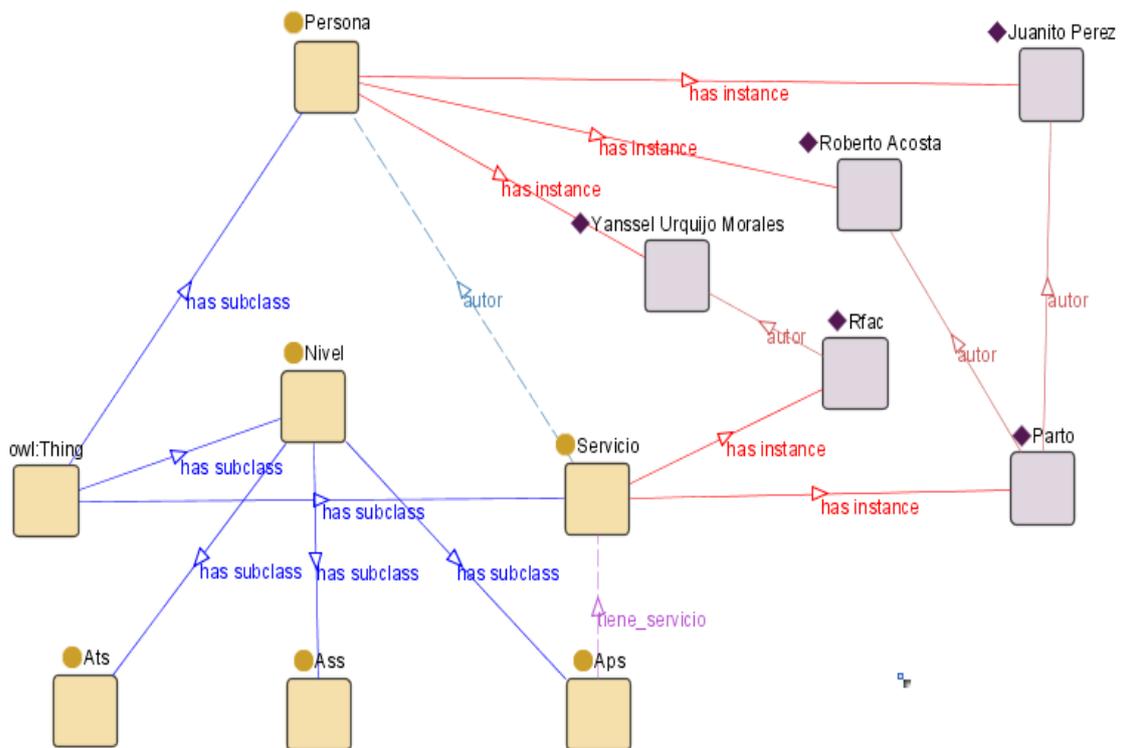


Figura 18. Relaciones entre clases e instancias de la ontología OSalud.

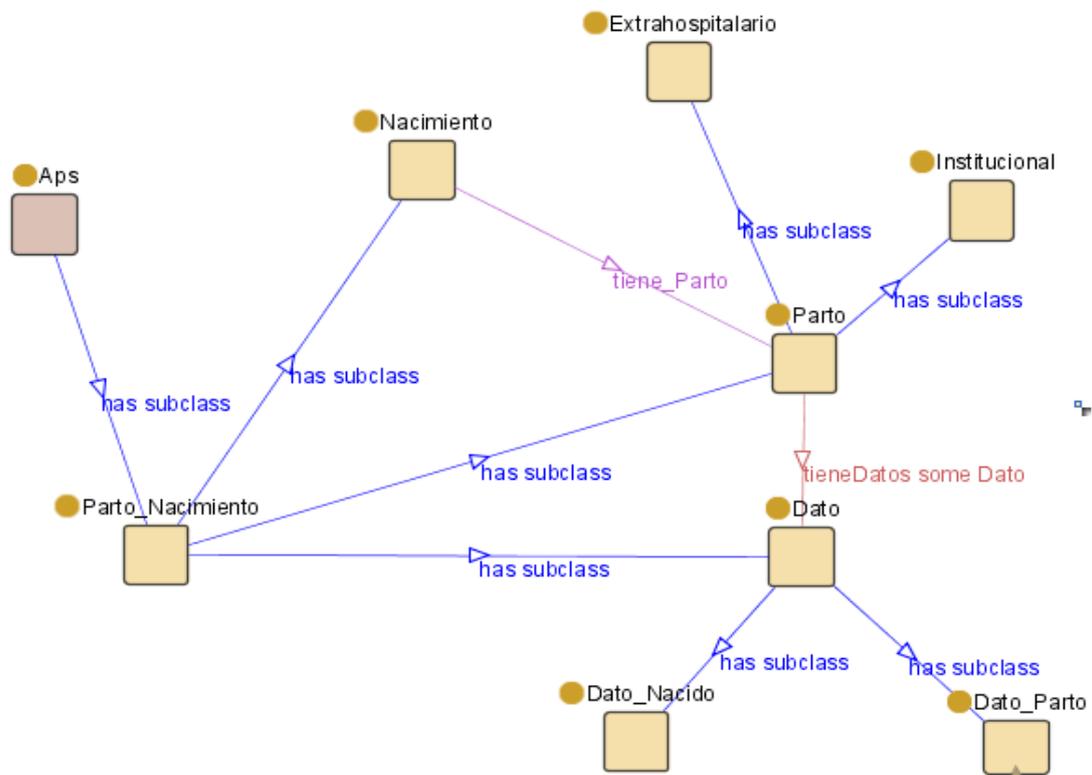


Figura 19. Relación entre clases de la ontología Registro de Parto y Nacimiento.

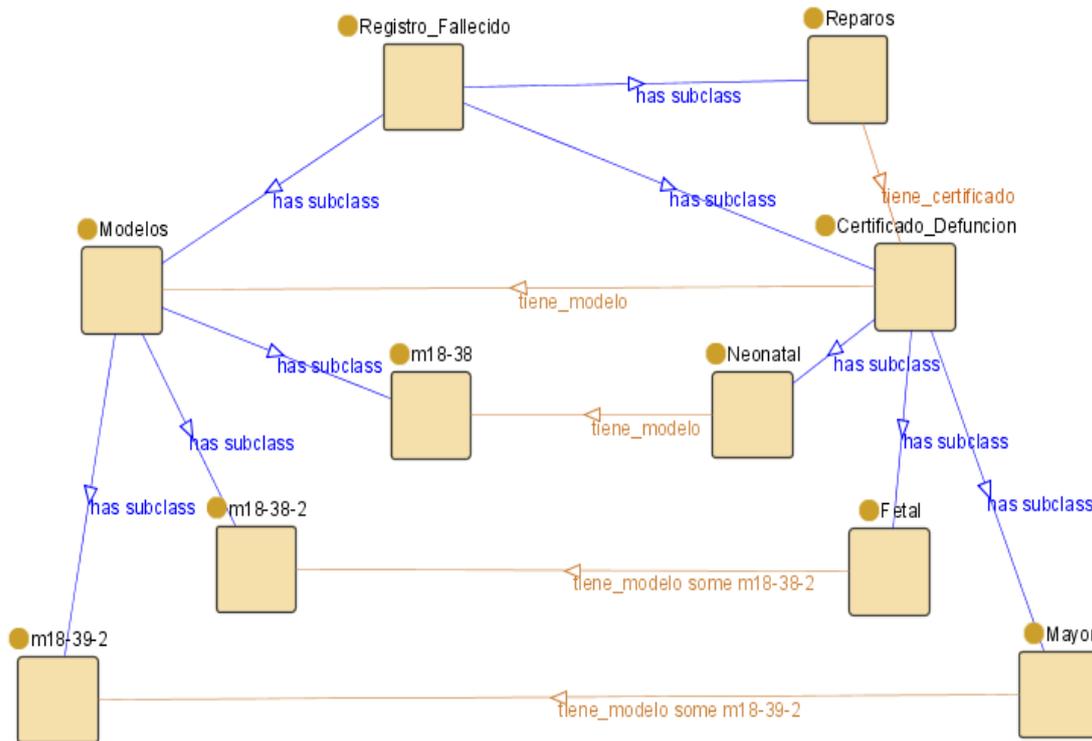


Figura 20. Relación entre clases de la ontología Registro de Fallecidos.

Anexo 3. Propuesta de Arquitectura.

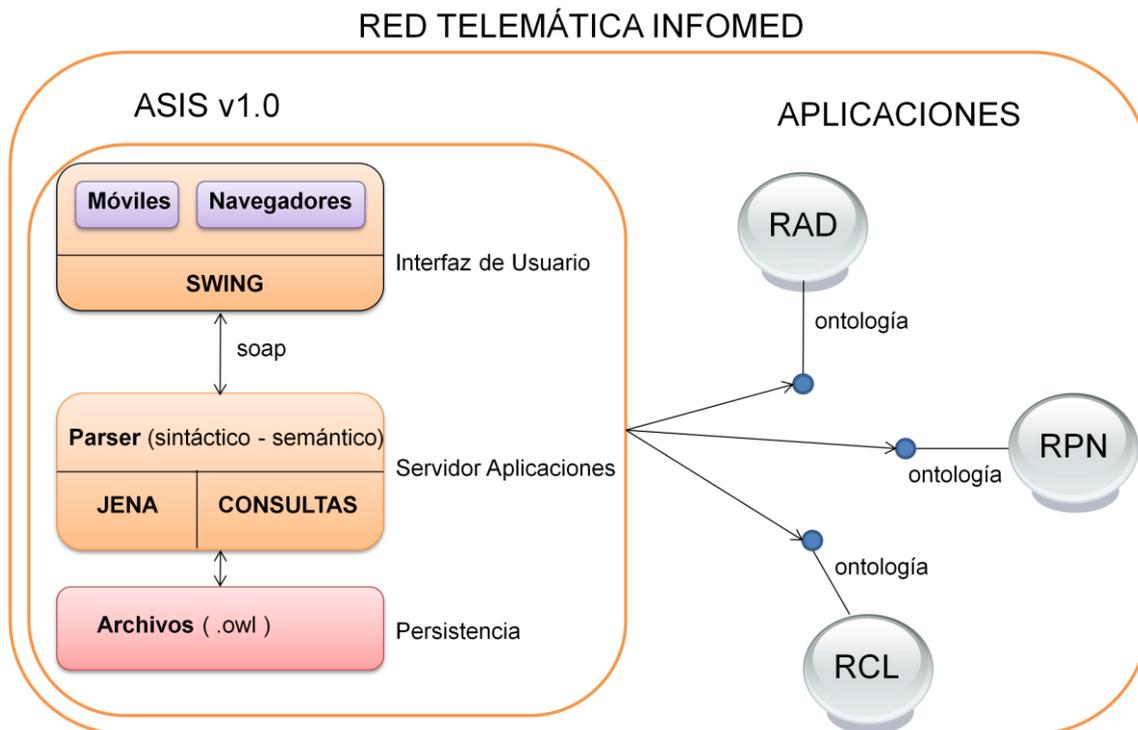
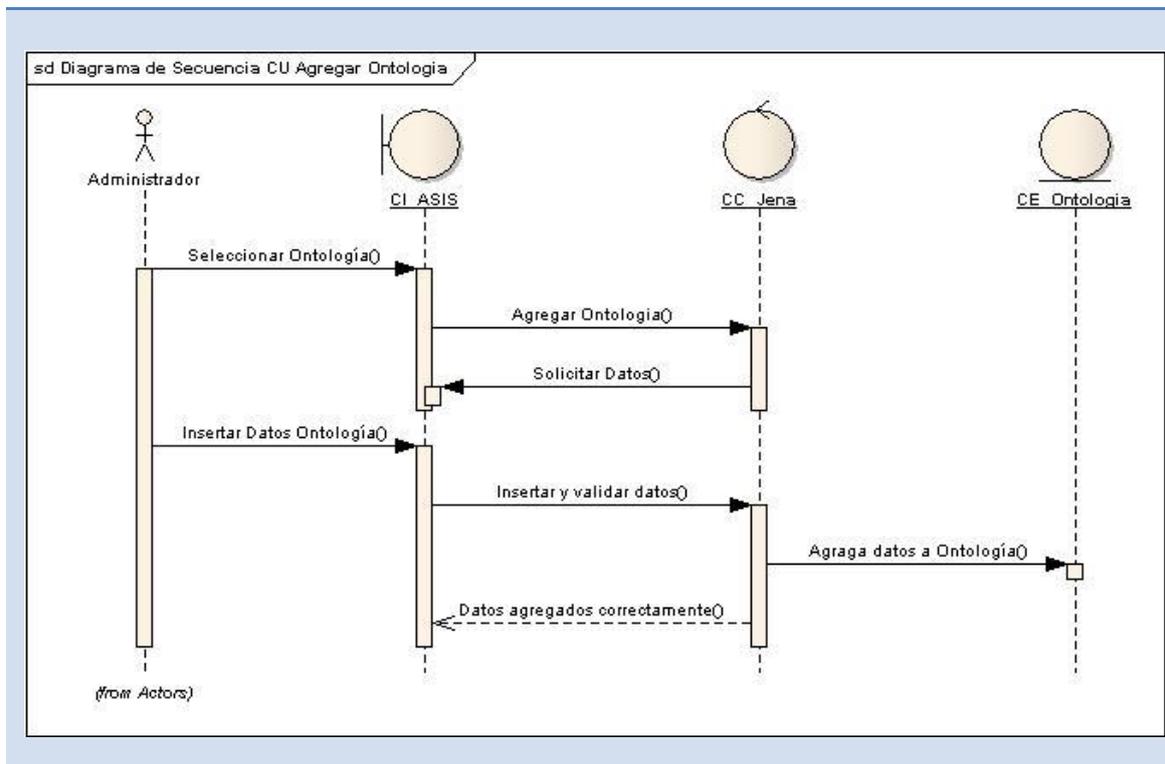


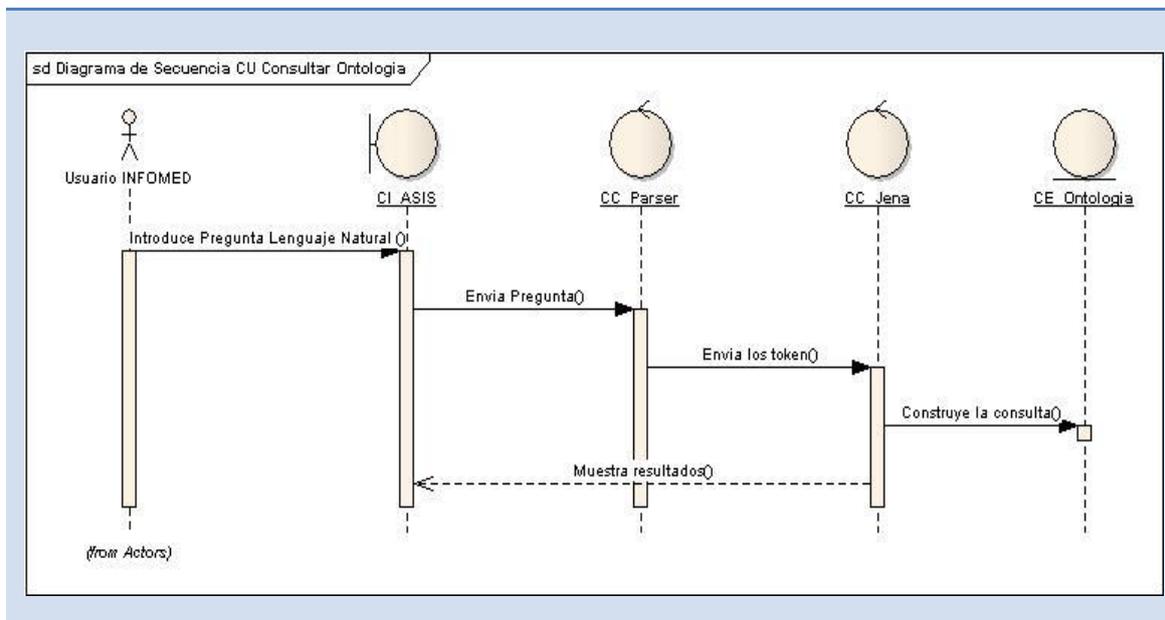
Figura 21. Propuesta de arquitectura.

Anexo 4. Diagramas de secuencia del análisis.

Caso de Uso Agregar Ontología

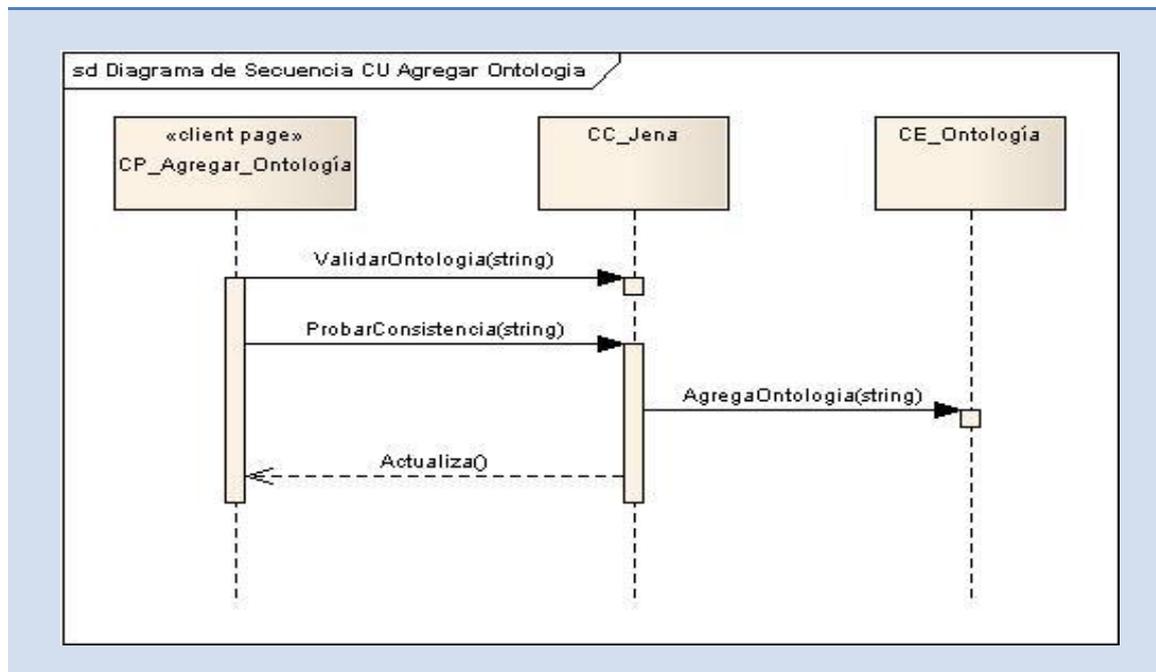


Caso de Uso Consultar Ontología

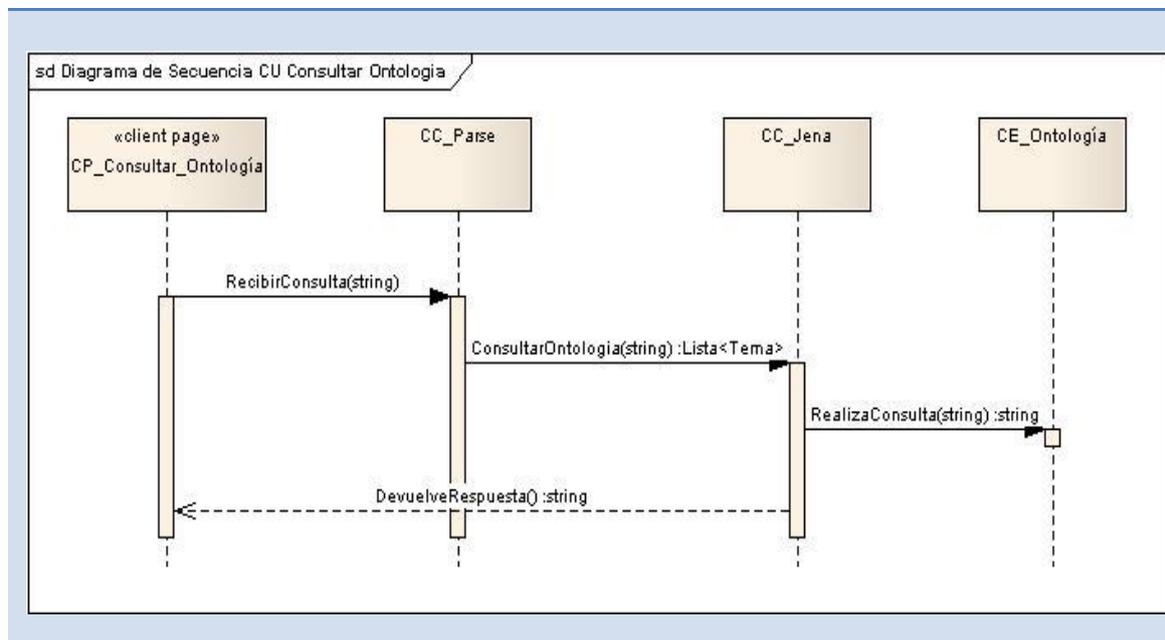


Anexo 5. Diagramas de secuencia del diseño.

Caso de Uso Agregar Ontología



Caso de Uso Consultar Ontología



GLOSARIO DE TÉRMINOS

Términos	Definiciones
Artefactos de software	Objeto fabricado durante el desarrollo de software que cumple una función técnica específica.
Axioma	La palabra 'axioma' viene de la palabra latina 'axioma', que significa "lo que parece justo". Cada una de las proposiciones compatibles entre sí a partir de las cuales se construye una teoría.
Lenguaje ontológico	Lenguaje de marcado para publicar y compartir datos usando ontologías en la Web.
Metadatos	Son datos sobre los datos, esto es, información sobre la información misma.
RUP	Proceso Unificado Racional (Rational Unified Process), proceso de desarrollo de software, es la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.
SOAP	Protocolo Simple de Acceso a Objetos (Simple Object Access Protocol). Es un protocolo para intercambiar mensajes, basado en XML, y de extendido uso en servicios Web.
SWS	Los Servicios Web Semánticos son una nueva tecnología resultante de la combinación de la Web Semántica y los Servicios Web lo cual permiten especificar formalmente los servicios, facilitando su localización y uso automatizados y orientados a metas.

GLOSARIO DE TÉRMINOS

Taxonomía	Es la ciencia que se encarga de la categorización o clasificación de cosas basado en un sistema predeterminado.
Tokens	Son los elementos más básicos sobre los cuales se desarrolla toda traducción de un programa, surgen en la primera fase, llamada análisis léxico, sin embargo se siguen utilizando en las siguientes fases antes de perderse en la fase de síntesis.
Web	World Wide Web o simplemente Web, es un sistema de documentos enlazados unos con otros, accesibles a través de Internet.
Web semántica	La Web semántica (del inglés <i>semantic web</i>) es la "Web de los datos". Se basa en la idea de añadir metadatos semánticos y ontológicos a la <i>World Wide Web</i> . Esas informaciones adicionales —que describen el contenido, el significado y la relación de los datos— se deben proporcionar de manera formal, para que así sea posible evaluarlas automáticamente por máquinas de procesamiento.
XML	Siglas en inglés de Extensible Markup Language (lenguaje de marcas ampliable), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades.