

**Universidad Central de Venezuela  
Facultad de Ciencias  
Escuela de Computación**

***Lecturas en Ciencias de la Computación***  
*ISSN 1316-6239*

**ONTOLOGÍAS: componentes,  
metodologías, lenguajes,  
herramientas y aplicaciones.**

Esmeralda Ramos y Haydemar Nuñez

**RT 2007-12**

Centro de Ingeniería de Software y Sistemas -ISYS-  
Laboratorio de Inteligencia Artificial -LIA-  
Caracas, Julio, 2007.

Universidad Central de Venezuela  
Facultad de Ciencias  
Escuela de Computación

Lecturas en Ciencias de la Computación  
ISSN 1316-6239

**ONTOLOGÍAS: componentes, metodologías, lenguajes,  
herramientas y aplicaciones.**

Ramos Esmeralda Núñez Haydemar

RT 2007-12

Centro de Ingeniería de Software y Sistemas -ISYS-  
Laboratorio de Inteligencia Artificial -LIA-

Caracas, Julio, 2007

RT 2007-  
Lecturas en Ciencias de la Computación  
ISSN 1316-6239

# ONTOLOGIAS: componentes, metodologías, lenguajes, herramientas y aplicaciones

Ramos Esmeralda Núñez Haydemar  
Julio 2007

## Resumen

Las ontologías intentan formular un esquema conceptual exhaustivo y riguroso de un dominio particular, con la finalidad de facilitar la comunicación, reusar y compartir información entre organizaciones, computadores y humanos. Una ontología define un vocabulario común que incluye además la interpretación de los conceptos básicos del dominio y sus relaciones.

¿Qué es una ontología?, ¿Para qué se desarrollan?, ¿Cómo se construyen?, ¿Qué lenguajes se utilizan para su formalización?, ¿Qué tipo de aplicaciones ontológicas se han desarrollado?, ¿Cuál es su utilidad en la Web semántica?. Éstas y otras preguntas son respondidas en este documento, el cual presenta una revisión detallada de la literatura especializada en la ingeniería ontológica. Se presentan definiciones, componentes de las ontologías, tipos de ontologías, lenguajes y métodos de desarrollo, así como algunos desarrollos ontológicos que revisten importancia. Además, se describe detalladamente el desarrollo de una ontología para el dominio de los insectos acuáticos con la metodología Methontology y utilizando el editor Protégé para su codificación.

Palabras Claves: Ontología, Conocimiento, Métodos de construcción, Methontology, Protégé.

---

Centro de Ingeniería de Software y Sistemas -ISYS-  
Laboratorio de Inteligencia Artificial -LIA-  
Escuela de Computación Facultad de Ciencias Universidad Central de Venezuela  
Los Chaguaramos. Apartado 47002Caracas 1041-A Venezuela  
ramos@kuaimare.ciencs.ucv.ve, hnunez@ciens.ucv.ve

## CONTENIDO

<b>INTRODUCCIÓN.....</b>	<b>2</b>
<b>1. ¿QUÉ ES UNA ONTOLOGÍA?.....</b>	<b>3</b>
1.1. ONTOLOGÍA EN INTELIGENCIA ARTIFICIAL.....	3
1.2. OBJETIVOS DE LAS ONTOLOGÍAS.....	4
1.3. USOS DE LAS ONTOLOGÍAS .....	4
1.4. COMPONENTES DE LAS ONTOLOGÍAS.....	6
1.5. CLASIFICACIÓN DE LAS ONTOLOGÍAS.....	6
1.6. TIPOS DE APLICACIONES ONTOLÓGICAS.....	8
<b>2. PROCESO DE DESARROLLO DE ONTOLOGÍAS .....</b>	<b>9</b>
2.1. METODOLOGÍA METHONTOLOGY.....	10
2.2. METODOLOGÍA ON-TO-KNOWLEDGE.....	13
2.3. PRINCIPIOS PARA EL DISEÑO DE ONTOLOGÍAS.....	16
<b>3. LENGUAJES PARA EL DESARROLLO DE ONTOLOGÍAS.....</b>	<b>17</b>
3.1. XML (eXTENSIBLE MARKUP LANGUAGE).....	17
3.2. RDF (RESOURCE DESCRIPTION FRAMEWORK).....	18
3.3. XOL (ONTOLOGY EXCHANGE LANGUAGE).....	18
3.4. OIL (ONTOLOGY INTERFACE LAYER) .....	19
3.5. DAML+OIL .....	19
3.6. OWL (WEB ONTOLOGY LANGUAGE).....	20
<b>4. HERRAMIENTAS PARA EL DESARROLLO DE ONTOLOGÍAS .....</b>	<b>21</b>
4.1. PROTÉGÉ .....	21
4.2. ONTOLINGUA.....	22
4.3. CHIMAERA.....	23
<b>5. ALGUNOS DESARROLLOS ONTOLÓGICOS .....</b>	<b>23</b>
5.1. UMLS (UNIFIED MEDICAL LANGUAGE SYSTEM).....	23
5.2. PHARMGKB (THE PHARMACOGENETICS AND PHARMACOGENOMICS KNOWLEGDE BASE) .....	25
5.3. ONTOLOGÍA PARA MEJORAR LOS SISTEMAS DE BASES DE DATOS DE TERRORISMO .....	27
5.4. ONTOLOGÍAS DESARROLLADAS POR STARLAB .....	27
<b>6. WEB SEMÁNTICA .....</b>	<b>28</b>
<b>7. UN CASO DE ESTUDIO EN LA CONSTRUCCIÓN DE ONTOLOGÍAS: ONTOLOGÍA DEL DOMINIO DE LOS INSECTOS ACUÁTICOS.....</b>	<b>31</b>
7.1. DESCRIPCIÓN DEL DOMINIO DE LOS INSECTOS ACUÁTICOS. ....	31
7.2. TAREAS DE CONCEPTUALIZACIÓN DE METHONTOLOGY: .....	32
7.3. CODIFICACIÓN DE LA ONTOLOGÍA CON PROTÉGÉ .....	38
<b>CONCLUSIONES.....</b>	<b>41</b>
<b>REFERENCIAS .....</b>	<b>42</b>

## Introducción

El uso de las ontologías ya es común en áreas como negocios, finanzas, internet, medicina, industrias, etc. En estos dominios las ontologías son usadas para formalizar capitales intelectuales, integrar fuentes de datos, representar y organizar conocimiento, entre otros, es decir, que son una alternativa de representación formal de conocimiento.

Las ontologías intentan formular un esquema conceptual exhaustivo y riguroso de un dominio dado con la finalidad de facilitar la comunicación, reusar y compartir información entre organizaciones, computadores y humanos. Una ontología define un vocabulario común que incluye además la interpretación de los conceptos básicos del dominio y sus relaciones.

¿Qué es una ontología?, ¿Para qué se desarrollan?, ¿Cómo se construyen?, ¿Qué lenguajes se utilizan para su formalización?, ¿Qué tipo de aplicaciones ontológicas se han desarrollado?, ¿Cuál es su utilidad en la Web semántica?, estas y otras preguntas son respondidas en éste documento, el cual presenta una revisión detallada de la literatura especializada en la ingeniería ontológica. El documento se ha estructurado de la siguiente manera:

En la primera sección se presentan diferentes definiciones dadas al término ontología, desde su origen, propuesto por Aristóteles, hasta su utilización en el área de inteligencia artificial. Además, se describen los componentes de las ontologías, se especifican los objetivos que se pretenden alcanzar con su desarrollo, las ventajas que se pueden obtener al utilizarlas y se presenta una clasificación considerando diferentes criterios como intención de uso, formalidad del lenguaje, generalidad, etc.

El proceso de construcción de ontologías y la descripción de dos metodologías ampliamente conocidas y utilizadas por la comunidad de desarrolladores de ontologías, se presentan en la Sección 2; además, se especifican algunas de las características que desde el punto de vista de diseño deberían exhibir las ontologías de calidad.

En las Secciones 3 y 4 se presentan algunos lenguajes y herramientas utilizados para desarrollar ontologías.

Algunas aplicaciones y sistemas de software, cuyo núcleo está constituido por ontologías se describen en la Sección 5.

En la Sección 6 se introduce el concepto de Web semántica y se especifica el rol de las ontologías en éste contexto.

Con la finalidad de mostrar la manera cómo pueden ser aplicados y utilizados algunos de los aspectos presentados en este documento, tales como componentes de las ontologías, clasificaciones, herramientas de codificación y metodologías de construcción; en la sección 7 se describe de manera detallada cómo se generó la ontología que estandariza y organiza el conocimiento del dominio de los insectos acuáticos.

# 1. ¿Qué es una ontología?

El término ontología tiene su origen en la filosofía, disciplina que trata de dar una explicación sistemática de la existencia; proviene de la conjunción de los términos griegos *ontos* y *logos* que significan existencia y estudio, respectivamente. Fue definido originalmente por Aristóteles en su empeño de clasificar todo lo existente en el universo, entre algunas de las definiciones de ontología se tienen:

- Ontología es la teoría o estudio del ser, más específicamente, de las características básicas de toda realidad. Durante el siglo XVII el término ontología se usaba como sinónimo de metafísica.
- Ontología es una rama de la metafísica que investiga y explica la naturaleza y las propiedades esenciales y relaciones de todo ser, o los principios y causas del ser.
- Ontología es el estudio metafísico de la naturaleza del ser y la existencia.
- Ontología es la rama de la metafísica que trata con la naturaleza del ser (Swartout, 1999).

## 1.1. Ontología en Inteligencia Artificial

Es común que cada comunidad que desarrolla ontologías adopte una definición propia dependiendo de sus necesidades. Los investigadores en inteligencia artificial adoptaron el término Ontología para describir todo lo que puede ser representado computacionalmente. A comienzo de los años noventa las ontologías pasaron a ser un tópico de investigación en inteligencia artificial, utilizándose de manera exitosa para representar conocimiento y en el procesamiento de lenguaje natural. Algunas definiciones de Ontología en Inteligencia Artificial son:

- Especificación explícita de una conceptualización (Ceccaroni, 2001).
- Librería de definiciones que pueden ser utilizadas para diferentes propósitos en diferentes dominios, que permiten compartir y reutilizar conocimiento y métodos de razonamiento entre agentes (Gómez, 1995).
- Sistema de conceptos definidos e interpretados de manera declarativa. El sistema define el vocabulario de un dominio y las restricciones que aplican para que éstos puedan ser combinados en el modelo del dominio (Devedzic, 2002).
- Resultado de seleccionar un dominio y aplicar sobre el mismo un método para obtener una representación formal de los conceptos que contiene y las relaciones que existen entre estos (Tramullas, 1999).
- Según Gruber (1993), una ontología es una especificación formal y explícita de una conceptualización compartida.

Esta última definición es la que logra mayor aceptación en la comunidad de desarrolladores de ontologías, por esta razón nos permitimos agregarle expresividad complementándola de la siguiente manera;

**Conceptualización:** Modelo abstracto de un fenómeno, que puede ser visto como un conjunto de reglas informales que restringen su estructura. Por lo general se expresa como un conjunto de conceptos (entidades, atributos, procesos), sus definiciones e interrelaciones (Uschold y Gruninger, 1996)

**Formal:** Organización teórica de términos y relaciones usados como herramienta para el análisis de los conceptos de un dominio.

**Compartida:** Captura conocimiento consensual que es aceptado por una comunidad.

**Explícita:** Se refiere a la especificación de los conceptos y a las restricciones sobre estos.

## 1.2. Objetivos de las ontologías

Según (Noy y McGuinness, 2001) algunos de los objetivos que se tienen para desarrollar ontologías son los siguientes:

- *Compartir entendimiento común de la estructura del conocimiento, entre personas o agentes de software.*

La ontología pone a disposición de los miembros de una comunidad los términos y conceptos del dominio de interés, lo cual permitirá a las personas o agentes de software extraer y agregar información según sus necesidades.

- *Permitir reutilizar el dominio de conocimiento*

Es posible que muchos dominios hagan uso de un conocimiento específico, si este conocimiento está constituido en una ontología podrá ser reutilizado por aquellos individuos que la necesiten sin necesidad de desarrollar una ontología propia.

- *Permitir separar conocimiento de dominio del conocimiento operacional*

Una ontología expresa el conocimiento del dominio de manera general de forma tal que pueda ser utilizado y manipulado por diversas técnicas o algoritmos.

- *Analizar el conocimiento del dominio*

Específicamente en lo que se refiere al estudio de los términos y relaciones que lo configuran, ya sea formalmente o no.

## 1.3. Usos de las ontologías

Es sabido que las personas, organizaciones y sistemas de software deben comunicarse entre sí para alcanzar los objetivos que se plantean en sus áreas de trabajo. Sin embargo, debido a diferentes necesidades, pueden variar ampliamente los puntos de vista y las asunciones con respecto a cuál es esencialmente el tema de la comunicación. Por lo general, cada uno de estos actores posee un lenguaje propio lo que da lugar a diferencias de conceptos, estructuras y métodos. La carencia de un entendimiento compartido conduce a:

- Pobre comunicación entre las personas de la organización, lo que trae como consecuencia dificultades en la identificación de requerimientos así como en la definición de las especificaciones de los sistemas.
- Métodos aplicados erróneamente y por lo tanto aplicaciones de software mal construidas, traen como consecuencia limitaciones para la interoperatividad de los sistemas informáticos y de un potencial reuso y compartición de la información.

Esto trae como consecuencia pérdida del esfuerzo y tiempo invertido en desarrollos de sistemas que no alcanzarán los objetivos planteados. El camino para solucionar estos

problemas es reducir o eliminar las barreras conceptuales y terminológicas y enfocarse en la búsqueda de un entendimiento compartido.

Las ontologías se usan para compartir entendimiento de algún dominio de interés el cual puede ser usado como marco de unificación para resolver problemas relacionados con ese dominio. Por tanto, una ontología especifica una conceptualización, una forma concreta de ver el mundo, y por ello cada ontología va a presentar un punto de vista diferente.

Una ontología puede funcionar como un marco para la unificación de los diferentes puntos de vista (ver Figura 1) y servir como base para:

- La comunicación entre personas con diferentes necesidades y puntos de vista.
- La interoperatividad entre sistemas alcanzada por una traducción entre diferentes modelos, métodos, paradigmas, lenguajes y herramientas de software.
- Apoyar a la ingeniería de sistemas al momento de reusar software, hacer el chequeo de consistencia (confiabilidad del software), y al momento de la especificación, sobre todo a la hora de levantar los requerimientos.

La principal utilidad de una ontología es ayudar a la compartición del conocimiento entre diferentes actores de un dominio determinado, como pueden ser personas, organizaciones y sistemas de software.

La comunicación es una de las principales actividades en cualquier dominio, sea esta una comunicación unidireccional o bidireccional, además la ambigüedad del lenguaje implica errores, malentendidos y esfuerzos improductivos, y esto es debido en gran parte a que las personas participantes en los proyectos tienen un conocimiento distinto del dominio del problema o que utilizan un lenguaje diferente para comunicarse. El uso de las ontologías ayuda a solventar estos problemas al incluir en un solo lugar todos los conceptos y relaciones del dominio de trabajo evitando errores en el manejo de la información unificando así el lenguaje de la comunicación.

En cuanto a los sistemas de software, el uso de ontologías en su desarrollo permite establecer correspondencia y relaciones entre los diferentes dominios de entidades de información. Pueden existir dos o más ontologías conectadas compartiendo información de dominios distintos, o se puede dar el caso que agentes de software requieran usar la información contenida en una ontología específica. Todo esto trae consigo la interoperatividad entre diferentes sistemas de computación ayudando a un entendimiento compartido más amplio y reutilizable.

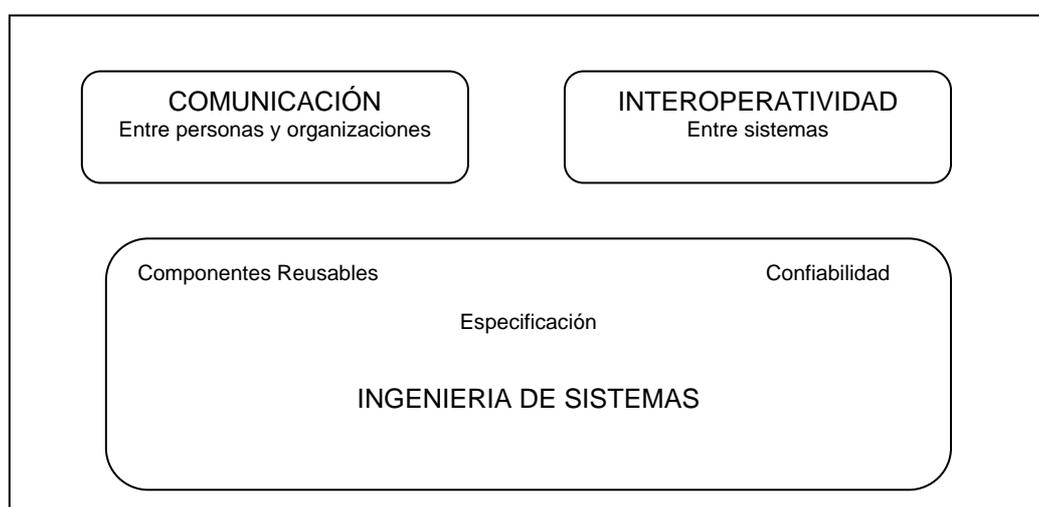


Figura 1. Usos de las Ontologías (Ushold, 1996)

## 1.4. Componentes de las Ontologías

Los componentes de una ontología varían de acuerdo al dominio de interés y a las necesidades de los desarrolladores. Por lo general entre los componentes se encuentran los siguientes (Lozano, 2002; Gómez y otros, 2004):

- **Clases:** Las clases son la base de la descripción del conocimiento en las ontologías ya que describen los conceptos (ideas básicas que se intentan formalizar) del dominio. Las clases usualmente se organizan en taxonomías a las que por lo general se les aplican mecanismos de herencia.
- **Relaciones:** Representan las interacciones entre los conceptos del dominio. Las ontologías por lo general contienen relaciones binarias; el primer argumento de la relación se conoce como el dominio y el segundo como el rango.
- **Funciones:** Son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de una ontología.
- **Instancias:** Representan objetos determinados de un concepto.
- **Taxonomía:** Conjunto de conceptos organizados jerárquicamente. Las taxonomías definen las relaciones entre los conceptos pero no los atributos de éstos.
- **Axiomas:** Se usan para modelar sentencias que son siempre ciertas. Los axiomas permiten, junto con la herencia de conceptos, inferir conocimiento que no esté indicado explícitamente en la taxonomía de conceptos. Los axiomas definidos en una ontología pueden ser estructurales o no estructurales: un axioma estructural establece condiciones relacionadas con la jerarquía de la ontología, conceptos y atributos definidos; un axioma no estructural establece relaciones entre atributos de un concepto y son específicos de un dominio. Los axiomas se utilizan también para verificar la consistencia de la ontología.
- **Propiedades (Slots):** Son las características o atributos que describen a los conceptos. Las especificaciones, rangos y restricciones sobre los valores de las propiedades se denominan *facets*. Para un concepto dado, las propiedades y las restricciones sobre éstos son heredadas por las subclases y las instancias de la clase.

## 1.5. Clasificación de las Ontologías

Las ontologías se pueden clasificar tomando en cuenta diferentes criterios, como por ejemplo, la intención de uso o tarea en particular, la formalidad del lenguaje utilizado para su construcción, la generalidad, entre otros. A continuación, se presentan algunas de estas clasificaciones (Gómez y otros, 2004; Uschold y Gruninger, 1996; FIPA, 2000):

### 1.5.1. Por su dependencia del contexto.

- **Ontologías de Nivel Superior:** Describen conceptos muy generales que son independientes de un problema particular o dominio. Por ejemplo, ontologías sobre el tiempo, el espacio, la materia, el objeto, el acontecimiento, la acción, entre otros
- **Ontologías de Dominio:** Proporcionan el vocabulario necesario para describir un dominio dado. Incluyen términos relacionados con los objetos del dominio y sus componentes, un conjunto de verbos o frases que dan nombre a actividades y procesos que tienen lugar en ese dominio, y conceptos primitivos que aparecen en teorías, relaciones y fórmulas que regulan o rigen el dominio.

- **Ontologías de Tareas:** Proveen un vocabulario sistemático de los términos usados para resolver problemas asociados con tareas particulares, ya sean dependientes o no del dominio.

- **Ontologías de Aplicación:** Contienen las definiciones necesarias para modelar el conocimiento requerido para una aplicación particular en un dominio dado. Describen conceptos que dependen tanto del dominio particular como de las tareas. Estas ontologías son especializaciones de las ontologías de dominio y de tareas.

La Figura 2 muestra las relaciones entre los tipos de ontologías por dependencia del contexto.

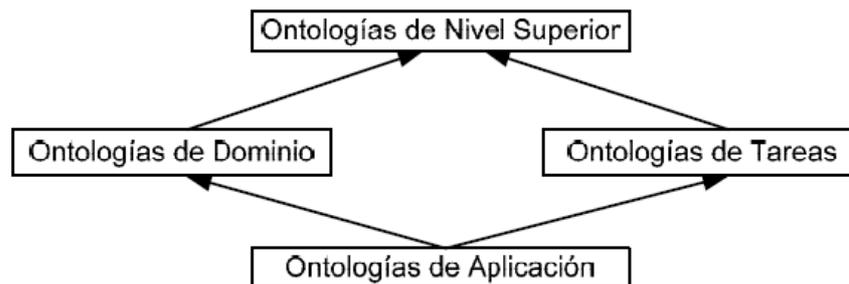


Figura 2. Tipos de Ontologías por su dependencia del contexto. (FIPA, 2000)

### 1.5.2. Por la granularidad de la conceptualización (cantidad y tipo de conceptualización).

- **Ontologías Terminológicas:** Especifican los términos que son usados para representar conocimiento en el universo de discurso. Suelen ser usadas para unificar vocabulario en un dominio determinado (contenido léxico y no semántico).

- **Ontologías de Información:** Especifican la estructura de almacenamiento de bases de datos. Ofrecen un marco para el almacenamiento estandarizado de información (estructura de los registros de una base de datos).

- **Ontologías de representación de conocimiento o Meta-ontologías:** Especifican la conceptualización que subyace en un paradigma o formalismo de representación de conocimiento, es decir, proporcionan el vocabulario necesario para modelar otras ontologías utilizando un determinado paradigma de representación de conocimiento.

### 1.5.3. Por su propósito de uso.

- **Ontologías para la comunicación entre personas:** Proporcionan los términos necesarios para describir y representar un área de conocimiento. Una ontología informal (no ambigua) puede ser suficiente.

- **Ontologías para la interoperabilidad entre sistemas:** Permiten realizar traducciones entre diferentes métodos, paradigmas, lenguajes y herramientas de software. La ontología se usa como un formato de intercambio de conocimiento.

- **Ontologías para beneficiar la ingeniería de sistemas:** Favorecen la reutilización de componentes, facilitan la adquisición de conocimiento e identificación de requerimientos, y aumentan la fiabilidad de los sistemas al proporcionar consistencia en el conocimiento utilizado.

#### 1.5.4. Por su nivel de formalidad.

Se refiere al grado de formalismo del lenguaje usado para expresar la conceptualización.

- **Ontología altamente informal:** Expresada en lenguaje natural (Glosario de términos).
- **Ontología informal estructurada:** Expresada en una forma restringida y estructurada de lenguaje natural, que permite incrementar la claridad y reducir la ambigüedad.
- **Ontología semi-formal:** Usa un lenguaje de definición formal, como los mostrados en la Sección 3.
- **Ontología rigurosamente formal:** La definición de términos se lleva a cabo de manera meticulosa usando semántica formal y teoremas.

#### 1.6. Tipos de aplicaciones ontológicas

Las organizaciones que desarrollan ontologías se esfuerzan cada vez más en la disminución de los costos y en el aumento de los beneficios que éstas proporcionan. Para alcanzar esta meta se requiere tener un entendimiento preciso del contexto de aplicación. A continuación se presenta un esquema de clasificación de aplicaciones ontológicas que proporciona a los desarrolladores lineamientos específicos sobre cómo deben usar las ontologías para alcanzar provecho en un contexto dado (Jasper y Uschold, 1999).

Este esquema de clasificación se basa en un conjunto de escenarios de aplicaciones ontológicas, entendiéndose por escenario un sistema o proceso que hace uso o se beneficia de la ontología.

Cada escenario considerado se caracteriza por:

- Propósito o beneficio (asistir la comunicación entre agentes humanos, alcanzar la interoperabilidad y mejorar los procesos y/o calidad de los sistemas de ingeniería de software)
- Rol (*Datos operacionales* -la información es consumida y producida por las aplicaciones-, *Ontologías* -información específica de términos y definiciones para conceptos importantes de un dominio-, *Lenguaje de representación ontológico* -información usada para desarrollar ontologías o aplicaciones-)
- Actores necesarios para implantar el escenario (autor de la ontología, autor de los datos operacionales, desarrollador de aplicaciones, usuario de aplicaciones y trabajador del conocimiento –persona que usa el conocimiento-)
- Soporte técnico
- Nivel de madurez

##### 1.6.1. Escenarios de aplicaciones ontológicas.

Los escenarios son abstracciones de aplicaciones específicas de ontologías tomadas de la industria o investigación, los escenarios son análogos a los casos de uso. Se consideran, entre otros, los siguientes escenarios:

**Autoría neutral:** Se crea un artefacto de información en un lenguaje y se traduce a diferentes formatos para ser usado en múltiples sistemas. Sus beneficios son: reutilización de

conocimiento, mejoras en el mantenimiento y retención de conocimiento de largo término. El artefacto creado puede ser una ontología o datos operacionales.

**Ontología como especificación:** Una ontología de un dominio dado es usada como la base para especificar y desarrollar alguna aplicación. Sus beneficios son: documentación, fiabilidad, mantenimiento y reutilización de conocimiento.

**Acceso común a la información:** La información es requerida por uno o varias personas o aplicaciones, pero la forma en que está expresada es mediante un vocabulario poco familiar o en formato inaccesible. La ontología devuelve la información entendible para proporcionar una comprensión compartida de los términos para mapear entre términos usados. Sus beneficios son: inter-operabilidad, uso más efectivo y reutilización del conocimiento.

**Ontología basada en búsquedas:** La ontología es usada para buscar en un repositorio de información una fuente en particular (páginas web, documentos, etc). Sus beneficios son: rápido acceso a la información y uso y reutilización más efectivo de las fuentes de conocimiento.

## 2. Proceso de Desarrollo de Ontologías

El proceso de construir una ontología no difiere mucho, en líneas generales, del usado para construir software. Según la definición dada por la IEEE, un software es un “programa de computación, procedimientos y documentación asociada, además de los datos para que se ejecute”. Las ontologías son productos de software y por lo tanto su desarrollo deberá seguir los estándares establecidos, por supuesto, adaptados a las características de las ontologías. En (FIPA, 2000), se identifican las actividades del proceso de desarrollo de software propuestas por la IEEE adaptadas al proceso de desarrollo de ontologías como sigue:

### • Actividades de administración de proyecto:

<i>Actividad</i>	<i>Objetivo</i>
Planificación	Identificar las tareas a realizar y los recursos disponibles, tales como, software, hardware, humanos, etc.
Control	Especificar los mecanismos para garantizar que las tareas se realicen según lo planificado.
Calidad	Especificar los estándares de calidad que deberán cumplir las tareas realizadas

### • Actividades de Desarrollo:

<i>Actividad</i>	<i>Objetivo</i>
Especificación	Realizar un documento que contenga información referente a: usuarios finales de la ontología, propósito, alcance, metas y grado de formalidad.
Conceptualización	Construir un modelo conceptual que describa el problema y su posible solución.

Formalización	Transformar el modelo conceptual en un modelo “semi-computable”, utilizando representaciones lógicas, grafos conceptuales, esquemas, etcetera.
Integración	Integrar ontologías existentes para garantizar la reutilización del conocimiento
Implantación	Codificar la ontología en un lenguaje formal
Mantenimiento	Actualizar la ontología cuando sea necesario

• **Actividades de Integración:**

<i>Actividad</i>	<i>Objetivo</i>
Adquisición de conocimiento	Adquirir conocimiento mediante la aplicación de técnicas apropiadas.
Evaluación	Evaluar la ontología.
Documentación	Documentar apropiadamente la ontología, para garantizar el éxito al ser compartida y reutilizada.

Al desarrollar ontologías es importante considerar las siguientes reglas básicas (Noy y McGuinness, 2001):

- No existe una manera única y correcta de modelar un dominio.
- El desarrollo ontológico es un proceso iterativo
- Los conceptos de la ontología deberán reflejar lo más fielmente posible a los objetos y relaciones del dominio.

En general, las metodologías proporcionan un conjunto de directrices que indican cómo deben realizarse las actividades identificadas en el proceso de desarrollo, qué técnicas son las más apropiadas en cada actividad y qué produce cada una de ellas. Son muchas las metodologías para desarrollar ontologías que se han propuesto, destacan entre otras, la de Uschold y King (Uschold, y King, 1995), la de Grüninger y Fox (Grüninger y Fox, 1995), Methontology (Gómez-Pérez y otros, 2004) y On-To-Knowledge (Staab y otros 2001), entre las mas conocidas.

A continuación, se describen dos metodologías de construcción utilizadas con frecuencia en desarrollos ontológicos como son Methontology y On-To-Knowledge.

### **2.1. Metodología Methontology**

Esta metodología fue desarrollada en el Laboratorio de Inteligencia Artificial de la Universidad Politécnica de Madrid (Gómez-Pérez y otros, 2004). Permite la construcción de ontologías a nivel de conocimiento e incluye la identificación del proceso de desarrollo de la ontología, un ciclo de vida basado en la evolución de prototipos y técnicas particulares para realizar cada actividad. Tiene sus raíces en las actividades identificadas por la IEEE para el proceso de desarrollo de software.

Methontology ha sido propuesta para la construcción de ontologías por la Fundación para Agentes Físicos Inteligentes (FIPA), que promueve la interoperabilidad entre las aplicaciones basadas en agentes (Corcho y otros, 2005).

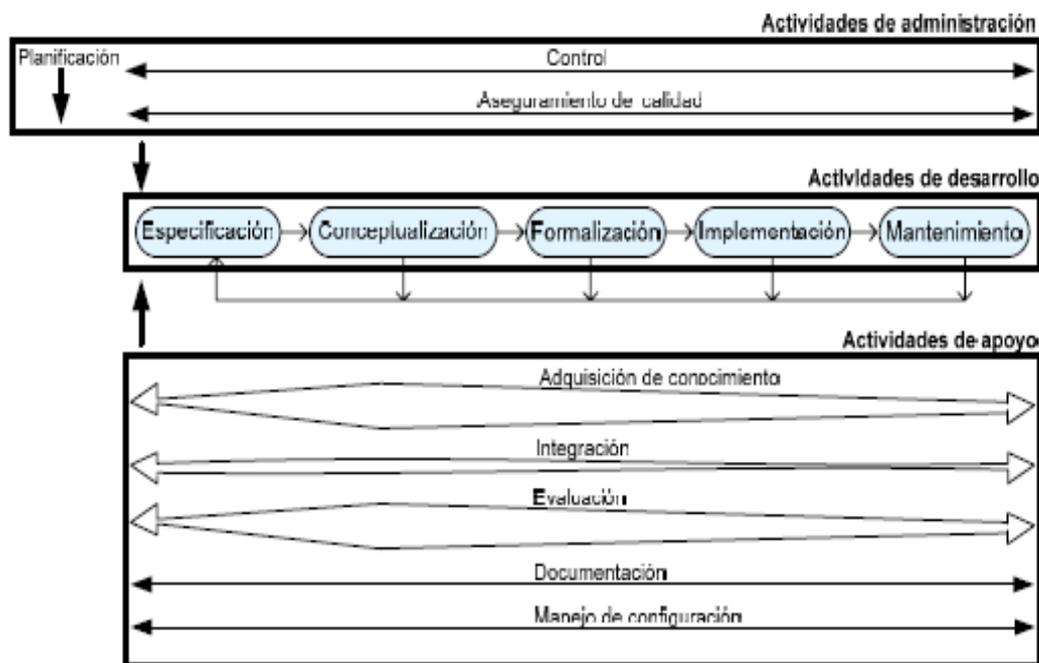


Figura 3. Tareas de la actividad de Conceptualización según Methontology. (Corcho y otros, 2005)

El ciclo de vida de Methontology se muestra en la Figura 3. Las actividades de control, aseguramiento de calidad, adquisición de conocimiento, integración, evaluación, documentación y manejo de configuración se realizan simultáneamente con las actividades de desarrollo. La conceptualización debe ser evaluada cuidadosamente para evitar la propagación de errores a las siguientes etapas del ciclo de vida de la ontología (Fernández, 1999).

La planificación se realiza antes del desarrollo de la ontología, por lo tanto no forma parte de su ciclo de vida. Las actividades de adquisición de conocimiento, integración y evaluación requieren un mayor esfuerzo en la etapa de conceptualización.

Methontology propone un ciclo de vida basado en la evolución de prototipos que permite añadir, cambiar y eliminar términos en cada nueva versión (prototipo) de la ontología.

Las actividades de desarrollo identificadas para Methontology son las siguientes:

**a) Especificación.**

Realizar un documento donde se señale el alcance, objetivos, propósito, nivel de formalidad y usuarios finales de la ontología.

**b) Conceptualización.**

Consiste en organizar y convertir una percepción informal de un dominio en una especificación semi-formal usando un conjunto de representaciones intermedias (tablas, diagramas) que puedan ser entendidas por los expertos del dominio y los desarrolladores de ontologías. En la Figura 4 se muestran los componentes de la ontología (conceptos, atributos, relaciones, constantes, axiomas formales, reglas e instancias) construidos en cada tarea, y se ilustra el orden propuesto para crear tales componentes durante la actividad de conceptualización. Las tareas de conceptualización en Methontology se describen a continuación:

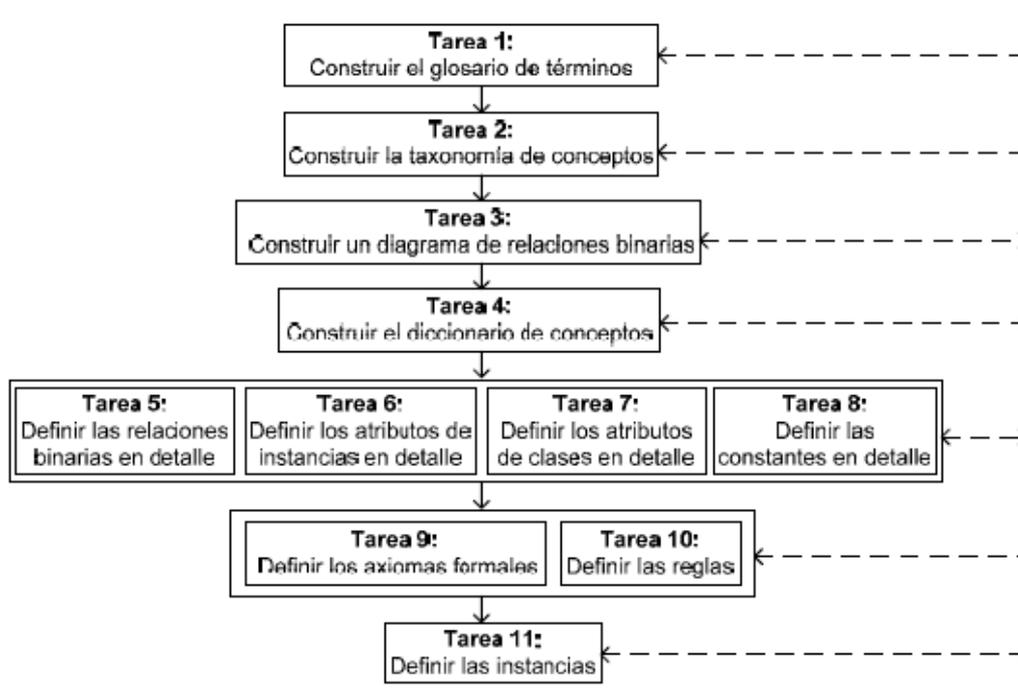


Figura 3. Tareas de la actividad de Conceptualización según Methontology.  
(Corcho y otros, 2005)

- **Tarea 1: Construir el glosario de términos.** El glosario de términos debe incluir todos los términos relevantes del dominio (conceptos, instancias, atributos, relaciones entre conceptos, etc.), sus descripciones en lenguaje natural, sus sinónimos y acrónimos.
- **Tarea 2: Construir la taxonomía de conceptos.** Cuando el glosario de términos tenga una cantidad importante de elementos, se debe construir una taxonomía que defina la jerarquía entre los conceptos. Se debe evaluar que la taxonomía creada no contenga errores.
- **Tarea 3: Construir un diagrama de relaciones binarias.** El objetivo de este diagrama es establecer las relaciones entre los conceptos de una o más taxonomías de conceptos. Se debe evaluar que el diagrama creado no contenga errores.
- **Tarea 4: Construir el diccionario de conceptos.** El diccionario de conceptos contiene los conceptos del dominio, sus relaciones, instancias, atributos de clases y atributos de instancias. Las relaciones, atributos de instancias, y atributos de clases son locales al concepto, lo que significa que sus nombres pueden repetirse en diferentes conceptos.
- **Tarea 5: Definir las relaciones binarias en detalle.** Se crea la tabla de relaciones binarias en la que se describe detalladamente todas las relaciones binarias incluidas en el diccionario de conceptos. Para cada relación binaria se debe especificar: nombre, conceptos fuente y destino, cardinalidad y relación inversa.
- **Tarea 6: Definir los atributos de instancia en detalle.** Se crea la tabla de atributos de instancias en la que se describe detalladamente todos los atributos de instancias incluidos en el diccionario de conceptos. Los atributos de instancias son aquellos atributos que describen

las instancias de un concepto, y sus valores pueden ser diferentes para cada instancia del concepto. Para cada atributo de instancia, se debe especificar: nombre, concepto al que pertenece, tipo de valor, rango de valores (en el caso de valores numéricos) y cardinalidad.

- **Tarea 7: Definir los atributos de clases en detalle.** Se crea la tabla de atributos de clases en la que se describe detalladamente todos los atributos de clases incluidos en el diccionario de conceptos. Para cada atributo de clase, se debe especificar: nombre, concepto donde es definido, tipo de valor, valor y cardinalidad.
- **Tarea 8: Definir las constantes en detalle.** Se crea la tabla de constantes en la que se describe detalladamente cada una de las constantes definidas en el glosario de términos. Para cada constante, se debe especificar: nombre, tipo de valor, valor y unidad de medida (para constantes numéricas).
- **Tarea 9: Definir los axiomas formales.** Se deben identificar los axiomas formales necesarios en la ontología y describirlos con precisión en una tabla. Para cada definición de axioma formal de debe especificar: nombre, descripción, expresión lógica que formalmente lo describe (preferiblemente utilizando lógica de primer grado), los conceptos, atributos y relaciones binarias a las cuales el axioma hace referencia y las variables utilizadas.
- **Tarea 10: Definir las reglas.** Se deben identificar cuáles reglas son necesarias en la ontología y describirlas en una tabla de reglas. Para cada regla, se debe especificar: nombre, descripción, expresión que formalmente la describe, los conceptos, los atributos y las relaciones a los que hace referencia y las variables usadas en la expresión. Para la especificación de las reglas se sugiere la forma: *Si <condiciones> entonces <consecuencias o acciones>*.
- **Tarea 11: Definir las instancias.** Una vez que el modelo conceptual de la ontología ha sido creado, se deben definir las instancias relevantes que aparecen en el diccionario de conceptos en una tabla de instancias. Para cada instancia se debe especificar: nombre, concepto al que pertenece y valores de los atributos.

### ***c) Formalización.***

Realizar la transformación del modelo conceptual en un modelo formal o semi-computable.

### ***d) Implementación.***

Realizar la codificación de la ontología utilizando un lenguaje formal (Ontolingua, XOL, OIL, DAML, OWL, entre otros).

### ***e) Mantenimiento.***

Esta actividad permite la actualización y corrección de la ontología.

## **2.2. Metodología On-To-Knowledge.**

La metodología On-To-Knowledge, es uno de los resultados mas importantes del proyecto europeo (EU-IST-1999-10132) del mismo nombre, que proporciona una guía específica para el desarrollo y mantenimiento de ontologías conjuntamente con el desarrollo de aplicaciones de gerencia de conocimiento. On-To-Knowledge se basa en el análisis de escenarios de uso y considera la identificación de los objetivos que deberían alcanzar las aplicaciones. Las fases de la metodología son: 1) Estudio de factibilidad 2) Comienzo 3) Refinamiento 4) Evaluación y 5) Mantenimiento (Staab y otros 2001; Sure and Studer, 2003; Sure y otros, 2003; Fensel y otros, 2002). A continuación se describen estas fases en detalle:

• **Fase 1. Estudio de factibilidad:** El correcto funcionamiento de un sistema de gerencia y manipulación de conocimiento dependerá de la manera como éste se integre en la organización en la cual se utilizará, es por esta razón que es necesario determinar los factores que podrían influir en su fracaso o éxito. Una manera de analizar la influencia de estos factores es mediante la realización de un estudio de factibilidad que permita identificar problemas y oportunidades, así como soluciones posibles. Un estudio de factibilidad ayuda a determinar la viabilidad económica y tecnológica del sistema a desarrollar.

Para esta fase la metodología On-To-Knowledge, adopta las directrices proporcionadas por CommonKads (Schreiber y otros, 1999; Alonso y otros, 2004). Sugiere el desarrollo del modelo de organización para identificar entre otros aspectos, problemas, oportunidades, posibles soluciones, fortalezas y personas involucradas. De manera particular recomiendan la identificación de las tareas, de los agentes (especialmente los humanos) y del conocimiento que utilizarán los agentes para llevar a cabo las tareas, utilizando las hojas de trabajos TM-1, AM-1 y TM-2 respectivamente (Sure y Studer, 2003).

Las actividades que se realizan en esta fase son:

1. Identificar problemas y oportunidades
2. Seleccionar áreas más prometedoras y soluciones potenciales
3. Identificar a las personas involucradas
4. Seleccionar herramientas de desarrollo

Productos que se obtienen en esta fase:

1. Hojas de trabajo de CommonKads para los modelos de organización, tareas, conocimiento y agentes.

• **Fase 2. Comienzo:** En esta fase se inicia el desarrollo real de la ontología, comenzando por la elaboración de un documento de especificación de requerimientos, en el cual se describe lo que la ontología deberá hacer, bosquejando el área de utilización. Se desarrolla una taxonomía informal, considerando qué conceptos deben ser incluidos y cuáles ser excluidos.

Los aspectos que debe mostrar el documento de requerimientos son, entre otros: la meta de la ontología, la especificación del dominio y el alcance, las aplicaciones que soportará la ontología, fuentes de conocimiento (dominio experto, organigramas, fuentes de conocimiento, esquemas de bases de datos, etc.), usuarios potenciales y escenarios de uso. Este documento se puede enriquecer adicionando preguntas de competencia (revisión de posibles preguntas que el sistema deberá responder que indiquen el alcance y contenido del dominio) y ontologías que puedan ser reusadas.

Las actividades que se realizan en esta fase son:

1. Especificar los requerimientos
2. Analizar las fuentes de conocimiento
3. Desarrollar una taxonomía informal

Productos que se obtienen en esta fase:

1. Documento de especificación de requerimientos.
2. Taxonomía informal

• **Fase 3. Refinamiento:** En esta fase se refina la taxonomía informal en función de los requerimientos y objetivos fijados en la fase anterior. El proceso de refinamiento conducirá al desarrollo de una ontología completa denominada ontología objetivo (*target*).

Las actividades que se realizan en esta fase son:

1. Adquirir conocimiento desde los expertos del dominio para refinar la taxonomía informal, hasta alcanzar la ontología *target*, la cual contendrá los conceptos relevantes del dominio y las relaciones entre estos. Esta actividad también contempla la identificación y modelado de los axiomas. La ontología *target* deberá expresarse en un nivel epistemológico utilizando para ello estructuras de representación como *frames*, lógica, grafos conceptuales, entre otros.
2. Formalizar la ontología mediante el uso de un lenguaje de implementación ontológico o algún editor de ontologías (Gómez y otros, 2004).

Productos que se obtienen en esta fase:

1. Ontología *objetivo* formalizada.

• **Fase 4. Evaluación:** En esta fase se prueba la utilidad de la aplicación desarrollada considerando los siguientes aspectos:

- Chequear si la ontología satisface los requerimientos especificados en el documento de requerimientos.
- Verificar la capacidad de respuesta a las preguntas de competencias.
- Chequear el funcionamiento de la aplicación desde el punto de vista de navegabilidad, búsqueda de conceptos y relaciones, entre otros aspectos.

Esta fase se ejecuta en varios ciclos hasta alcanzar el nivel especificado, siendo el principal insumo, la retroalimentación de los usuarios.

Las actividades que se realizan en esta fase son:

1. Sobre la base de la retroalimentación, revisar y expandir la ontología.
2. Analizar los patrones usados en el chequeo de funcionamiento.
3. Analizar las preguntas de competencia.

Productos que se obtienen en esta fase:

1. Aplicación basada en la ontología

• **Fase 5. Mantenimiento:** Esta fase tiene como objetivo reflejar en la ontología los cambios que se producen en la vida real. Para esto se realiza un mantenimiento frecuente siguiendo un conjunto de reglas estrictas para la actualización, eliminación e inserción. De manera similar a la fase de refinamiento, la retroalimentación de los usuarios es importante en la identificación de los cambios.

Las actividades que se realizan en esta fase son:

1. Reunir los cambios identificados
2. Actualizar la ontología según las reglas especificadas

En la Figura 5 se muestran las fases de On-To-Knowledge, indicando para cada una las actividades que se realizan y los productos que se obtienen.

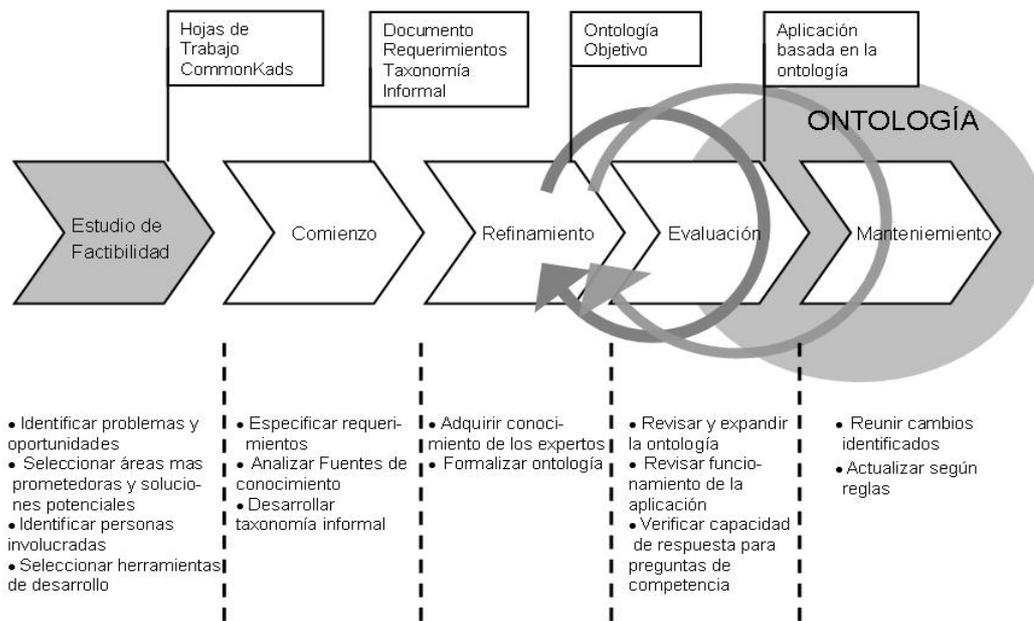


Figura 5. Fases de la metodología On-To-Knowledge

### 2.3. Principios para el diseño de ontologías

Al momento de diseñar una ontología, es necesario considerar algunas de las características deseables que éstas deberían exhibir. Los principios de diseño a considerar son los siguientes: (FIPA, 2000; Lozano, 2002; Nirenburg y Raskin, 2004; Silva, 2002; Alonso y otros, 2004).

- **Claridad y Objetividad:** Definir los conceptos en forma clara y objetiva utilizando lenguaje natural para evitar ambigüedades.
- **Coherencia:** Garantizar que todas las inferencias derivadas sean consistentes con los axiomas.
- **Completitud:** Los conceptos deben ser expresados en términos necesarios y suficientes.
- **Estandarización:** Siempre que sea posible, los nombres asignados a los términos deberán seguir un estándar, definiendo y respetando reglas para la formación de los mismos.
- **Máxima extensibilidad monótona:** Deberá ser posible incluir en la ontología especializaciones o generalizaciones, sin requerir una revisión de las definiciones existentes.
- **Principio de distinción ontológica:** Las clases de la ontología con diferente criterio de identidad, deberán ser disjuntas.
- **Diversificación de las jerarquías:** Para que la ontología se vea favorecida con los mecanismos de herencia múltiple, es conveniente usar tantos criterios de clasificación como sea posible, de manera de representar la mayor cantidad de conocimiento.
- **Minimización de la distancia semántica:** Conceptos similares deberán ser agrupados y representados utilizando las mismas primitivas.

- **Mínimo compromiso ontológico:** Una ontología debería imponer las menores exigencias posibles sobre el dominio que modela, es decir, se deben construir sólo los axiomas necesarios para representar el mundo a ser modelado.
- **Modularidad:** Al especificar una ontología se hacen definiciones de diferentes elementos como clases, relaciones y axiomas; tales definiciones se pueden agrupar en *teorías* que reúnen los objetos de una ontología más relacionados entre sí. Se puede lograr una organización altamente modular con máxima cohesión en cada módulo y mínima interacción, considerando que cada teoría es un módulo en la organización de la ontología. La modularidad permite flexibilidad y posibilidad de reuso de algunos módulos de la ontología.
- **Mínima dependencia con respecto a la codificación:** Una ontología debería permitir que los agentes que compartan los conocimientos, puedan ser implementados en diferentes sistemas y estilos de representación.

Un diseño ontológico ideal debería cumplir con todos estos criterios, pero no siempre es posible.

### 3. Lenguajes para el desarrollo de ontologías

A la hora de elegir un lenguaje para la definición de una ontología se deben considerar los siguientes aspectos:

- El lenguaje debe poseer una sintaxis bien definida para poder leer con facilidad la ontología.
- Debe tener una semántica específica para comprender perfectamente el funcionamiento de la ontología.
- Debe tener suficiente expresividad para poder capturar varias ontologías.
- Debe ser fácilmente traducible desde y hacia otros lenguajes ontológicos.
- Debe ser eficiente a la hora de realizar razonamiento.

A continuación se presentan algunos lenguajes utilizados en el desarrollo de ontologías:

#### 3.1. XML (eXtensible Markup Language)

XML es un meta-lenguaje derivado de SGML, que permite la definición de lenguajes de marcado adecuados para usos específicos. Es un lenguaje basado en marcas tipo etiquetas. XML es una manera flexible de crear formatos de información comunes y compartir tanto los formatos como los datos entre sistemas de computación. XML es un estándar W3C (World Wide Web Consortium).

Un documento XML permite estructurar de manera jerárquica la información que se pretende almacenar. El documento debe tener exactamente un elemento de nivel superior (elemento raíz).

La estructura de un documento XML se describe a través de DTDs (Definición de Tipo de Documento) o Schemas (Esquemas). Los DTDs poseen un lenguaje propio para su escritura, no admiten espacios de nombres (NameSpaces) y soportan tipos de datos muy limitados. Los Schemas utilizan sintaxis XML, admiten espacios de nombres y permiten definir tipos de datos simples, complejos y propios del usuario.

XML puede ser utilizado como un formato para el intercambio de datos y como lenguaje de serialización para otros lenguajes. Existen muchos analizadores (parsers) para XML que pueden ser reutilizados por los nuevos lenguajes que lo usan como su lenguaje de serialización (De Bruijn, 2003)

### **3.2. RDF (Resource Description Framework)**

RDF es una recomendación de la W3C para representar metadatos en la Web. Proporciona un medio para agregar semántica a un documento sin referirse a su estructura. RDF es una infraestructura para la codificación, intercambio y reutilización de metadatos estructurados. (Fensel y otros, 2000)

Debido a que RDF no define ningún vocabulario particular para la creación de los datos, era necesario un lenguaje que proporcionara las primitivas apropiadas. En consecuencia, se creó la especificación RDF Schema (RDFS). RDFS (o lenguaje de descripción de vocabulario RDF) amplía RDF con algunas primitivas básicas (basadas en marcos) para modelar ontologías, tales como clases, propiedades e instancias. También incorpora las relaciones *instancia-de* y *subclase-de*.

Un XML Schema indica el orden y la combinación de las etiquetas (estructura) en un documento XML; mientras que un RDF Schema sólo proporciona información sobre la interpretación (semántica) de las declaraciones en un modelo de datos RDF sin especificar nada sobre la sintaxis. (De Bruijn, 2003).

El modelo de datos de RDF está formado por recursos (objetos) y pares de atributos/valores (Fensel y otros, 2000). Un recurso representa cualquier entidad que pueda ser referenciada por un URI (Identificador Único de Recursos). Los atributos representan las propiedades de los recursos, y sus valores pueden ser entidades atómicas (por ejemplo: strings, enteros) u otros recursos. Un modelo RDF puede ser representado como un grafo dirigido, donde los recursos y los valores constituyen los nodos, y los atributos constituyen los arcos, un modelo RDF forma una red semántica.

Todo lo expresable en RDF, es expresable en sintaxis lineal de XML. La razón de la existencia de RDF es que provee un modo estándar para representar metadatos en XML. Usando directamente XML para representar metadatos, podrían obtenerse varias representaciones diferentes. Para procesar los modelos RDF, estos deben ser serializados en XML previamente (RDF representa el modelo abstracto y XML proporciona la representación textual concreta del modelo).

RDFS carece de capacidades para describir la semántica de los conceptos y las relaciones más allá de aquella provista por los mecanismos de herencia. Además, solo provee las primitivas más básicas para el modelado de ontologías y no proporciona soporte para definir axiomas directamente. RDFS no es lo suficientemente expresivo para representar ontologías de gran complejidad.

### **3.3. XOL (Ontology Exchange Language)**

Es un lenguaje para el intercambio de ontologías. Fue diseñado para ser utilizado como un lenguaje intermedio para permitir la transferencia de las ontologías entre diferentes sistemas de bases de datos, herramientas de desarrollo de ontologías o aplicaciones. Aunque XOL fue diseñado para el intercambio de ontologías bioinformáticas, puede ser utilizado para

ontologías de cualquier dominio. La sintaxis de XOL está basada en XML y la semántica está basada en OKBC-Lite, que es una forma simplificada del modelo de conocimiento de OKBC (Open Knowledge Base Connectivity). OKBC es un API (Interfaz de Programa de Aplicación) que permite el acceso a sistemas de representación de conocimiento basados en marcos (Karp y otros, 1999).

### 3.4. OIL (Ontology Interface Layer)

Es un lenguaje de representación de ontologías basado en Web y capas de inferencia, que combina las primitivas de representación de conocimiento de los lenguajes basados en marcos con la semántica formal y los servicios de razonamiento proporcionados por la lógica descriptiva (Fensel y otros, 2000). Incluye una semántica precisa para describir el significado de los términos.

OIL unifica tres aspectos importantes proporcionados por diferentes comunidades como son: lógica descriptiva, sistemas basados en marcos y lenguajes Web

- **Lógica descriptiva:** Un lenguaje formal para representar el conocimiento de un dominio de aplicación y razonar sobre el mismo.
- **Sistemas basados en marcos:** Sus primitivas de representación principales son las *clases* y sus *propiedades*. Las propiedades no tienen alcance global, sólo son aplicables en las clases en las que fueron definidas.
- **Estándares Web (XML y RDF):** OIL tiene una sintaxis bien definida en XML basada en DTDs y Schemas. Además, es una extensión de RDF y RDFS. OIL está formado por 4 capas:
  - **Core OIL:** coincide en gran parte con RDFS.
  - **Standard OIL:** es un lenguaje que abarca las primitivas de representación necesarias para proporcionar un adecuado poder expresivo a la ontología. Especifica la semántica y hace viable la inferencia.
  - **Instance OIL:** permite la definición de instancias en la ontología.
  - **Heavy OIL:** reservada originalmente para posibles extensiones futuras de OIL. Puede incluir capacidades adicionales de representación y razonamiento.

Cada capa adicional agrega funcionalidad y complejidad a la capa anterior. Esto permite que los agentes (humanos o máquinas) que se ubiquen en una capa particular puedan entender parcialmente las ontologías que se expresan en cualquiera de las capas más altas.

Principales limitaciones de OIL (Fensel y otros, 2000):

- Aunque OIL proporciona un mecanismo para heredar los valores de las superclases, estos valores no pueden ser sobrescritos en una especialización.
- Sólo un número fijo de propiedades algebraicas pueden ser expresadas sobre los atributos.
- No soporta dominios concretos (por ejemplo: números enteros).

### 3.5. DAML+OIL

Es un lenguaje de marcado semántico para recursos Web. Es un estándar propuesto por la W3C para la representación de ontologías y metadatos. DAML (DARPA Agent Markup Language) fue transformado a DAML+OIL a través de la inclusión de algunas características de OIL al lenguaje. DAML consiste en un formalismo que permite a los agentes de software interactuar entre ellos. (Su y otros, 2005)

DAML+OIL se construye sobre RDF y RDFS, pero proporciona primitivas de representación más ricas, comúnmente encontradas en la lógica descriptiva. Algunas de las ideas basadas en marcos proporcionadas por OIL fueron eliminadas.

DAML+OIL soporta tipos de datos complejos, a diferencia de OIL que sólo soporta el tipo de dato *string* (De Bruijn, 2003), los tipos de datos utilizados provienen de XML Schema. El lenguaje tiene una semántica bien definida.

### 3.6. OWL (Web Ontology Language)

Es un lenguaje de marcado semántico desarrollado por la W3C para publicar y compartir ontologías sobre el World Wide Web. Es una extensión del vocabulario de RDF y se deriva de DAML+OIL. OWL está diseñado para ser utilizado por aplicaciones que necesitan procesar el contenido de la información en lugar de sólo presentarla a las personas (McGuinness y Van Harmelen, 2004). OWL proporciona tres sub-lenguajes diseñados para ser utilizados por comunidades específicas de desarrolladores y usuarios. La característica que define a cada lenguaje es su expresividad.

- **OWL Lite:** Es el sub-lenguaje con sintaxis más simple, su intención es ser utilizado en situaciones donde se requiera una jerarquía de clases y restricciones simples.
- **OWL DL:** Es mucho más expresivo que OWL Lite y está basado en lógica descriptiva. Proporciona la máxima expresividad posible sin perder la completitud computacional (todas las conclusiones pueden ser deducidas) y la posibilidad (todos los cálculos se realizan en un tiempo finito).
- **OWL Full:** Es el sub-lenguaje más expresivo, su intención es ser utilizado en situaciones donde una alta expresividad es más importante que la capacidad de garantizar la completitud computacional y la posibilidad.

OWL DL es una extensión de OWL Lite y OWL Full es una extensión de OWL DL. Las afirmaciones siguientes se cumplen (sus inversas no): (McGuinness y Van Harmelen, 2004)

- Una ontología OWL Lite válida, es una ontología OWL DL válida.
- Cualquier conclusión válida de una ontología OWL Lite, será una conclusión válida en OWL DL.
- Una ontología OWL DL válida, es una ontología OWL Full válida.
- Cualquier conclusión válida de una ontología OWL DL será una conclusión válida en OWL Full.

Algunas ontologías desarrolladas con OWL son: *koala.owl* (ontología simple sobre humanos y marsupiales), [MGEDOntology.owl](#) (ontología biológica sobre Datos de Expresión de Microarreglos Genéticos), [shuttle-crew-ont.owl](#) (ontología sobre el equipo de un transbordador espacial), [fgdc-csdgm.owl](#) (ontología para contenido estándar de metadatos geoespaciales digitales), *camera.owl* (ontología sobre las partes individuales de una cámara fotográfica), [tambis-full.owl](#) (ontología de ciencia biológica desarrollada por el proyecto TAMBIS), *fsm.owl* (ontología simple para máquinas de estado finito), *hu.owl* (una división jerárquica de las unidades hidrológicas), entre otras (Protégé, 2005) .

## 4. Herramientas para el desarrollo de ontologías

Los editores de ontologías son herramientas especializadas que apoyan la construcción de estas. Las facilidades que proporcionan van desde la definición y modificación de conceptos, propiedades, relaciones, axiomas y restricciones, hasta la inspección y navegación.

Una de las herramientas más utilizadas por los desarrolladores de ontologías es Protégé, la cual se describe a continuación:

### 4.1. Protégé

Protégé es un software libre de código abierto implementado en Java, desarrollado en la Universidad de Stanford, que permite la construcción de ontologías de dominio. Es capaz de operar como una plataforma para acceder a otros sistemas basados en conocimiento o aplicaciones integradas, o como una librería que puede ser usada por otras aplicaciones para acceder y visualizar bases de conocimiento. La herramienta ofrece una interfaz gráfica que permite al desarrollador de ontologías enfocarse en la modelación conceptual sin que requiera de conocimientos de la sintaxis de los lenguajes de salida.

El modelo de conocimiento de Protégé está basado en marcos (frames). Las primitivas de representación internas en Protégé pueden ser redefinidas declarativamente, permitiendo tener representaciones apropiadas para una variedad de lenguajes de ontologías. Las primitivas de representación (elementos de su modelo de conocimiento) proporcionan clases, instancias de esas clases, propiedades que representan los atributos de las clases y sus instancias, y restricciones que expresan información adicional sobre las propiedades. Protégé comprueba la entrada de datos nuevos, y no permite dos clases o atributos con el mismo nombre. (ISCB, 2005)

Protégé puede correr como una aplicación local o a través de un cliente en una comunicación con un servidor remoto. El navegador Web de Protege permite a los usuarios compartir, navegar y editar sus ontologías utilizando un navegador Web estándar, lo que proporciona un ambiente de colaboración que puede ayudar a las comunidades en el desarrollo de ontologías. Protégé ha sido utilizado como el ambiente de desarrollo primario para muchas ontologías, y se ha convertido en la herramienta más utilizada en el mundo para trabajar con OWL. La comunidad de usuarios de Protégé regularmente contribuye a mejorar la calidad del software y participa en grupos de discusión en línea dedicados a formular preguntas, realizar peticiones de nuevas características y cuestiones de soporte técnico.

Protégé presenta una gran capacidad de extensión, debido al soporte de conectores (plugins), que son aditivos que se adquieren de manera individual y se acoplan al entorno de trabajo de Protégé para añadirle funcionalidad. Existen varios conectores disponibles para importar ontologías en diferentes formatos, incluyendo DAG-EDIT, XML, RDF y OWL. Las herramientas PROMPT, son conectores para Protégé que permiten a los desarrolladores integrar ontologías, trazar los cambios en las ontologías a través del tiempo y crear vistas de las mismas.

A continuación, se presentan algunos conectores de Protégé y las funcionalidades que ofrecen:

- *Protégé Web Browser*: es una aplicación Web basada en Java que permite a los usuarios compartir, navegar y editar sus ontologías utilizando un navegador Web estándar; lo que

proporciona un ambiente de colaboración que puede ayudar a las comunidades en el desarrollo de ontologías biomédicas

- *OWL*: permite cargar, guardar y editar ontologías OWL en Protégé.
- *DAML+OIL*: permite crear y editar ontologías DAML+OIL en Protégé. *RDF*: permite crear, importar y guardar archivos RDF(S) en Protégé.
- *Bean Generator*: genera los archivos Java, correspondientes a una ontología desarrollada en Protégé, para su posterior uso desde JADE.
- *Data Genie*: permite importar los datos de una base de datos arbitraria en Protégé.
- *Jambalaya*: proporciona un ambiente de visualización extensible, flexible, y escalable para la exploración, navegación, y entendimiento de las ontologías. Las clases e instancias son representadas como nodos en un grafico; los tipos diferentes se pueden distinguir utilizando distintos colores. Las flechas dirigidas (arcos) son utilizadas para mostrar las relaciones entre los conceptos y las instancias.
- *Media Slot Widget*: permite la inclusión y despliegue de archivos de audio y video en Protégé.
- *Prompt*: permite manejar múltiples ontologías, comparar versiones de la misma ontología, integrar ontologías y extraer una parte de una ontología.
- *PromptViz*: crea representaciones visuales de las diferencias entre dos versiones de una ontología.
- *Algernon Tab*: es un sistema de inferencia basado en reglas implementado en Java. Realiza encadenamiento hacia adelante y hacia atrás.

Los componentes de la interfaz de usuario estándar de Protégé para mostrar y adquirir datos pueden ser reemplazados con nuevos componentes que se adaptan mejor a tipos particulares de ontologías. La herramienta proporciona facilidades mediante las cuales el sistema puede generar automáticamente los formularios de entrada de datos para adquirir las instancias de los conceptos definidos por la ontología fuente.

Protégé tiene diferentes mecanismos de almacenamiento, incluyendo bases de datos relacionales, XML y archivos planos.

Protégé puede ser conectado directamente a programas externos con la finalidad de utilizar sus ontologías en aplicaciones inteligentes, tales como servicios de razonamiento y clasificación. Los desarrolladores de sistemas pueden utilizar la Interfaz de Programa de Aplicación (API) Java de Protégé para acceder y manipular las ontologías.

Protégé está disponible de manera gratuita, junto con los conectores y algunas ontologías, en su sitio Web (<http://protege.stanford.edu/>).

## 4.2. Ontolingua

Ontolingua es una herramienta de desarrollo para navegar, crear, editar, modificar, verificar, evaluar y usar ontologías. Contiene una librería de ontologías cuyas definiciones, axiomas y términos no-lógicos, pueden ser reutilizadas en la construcción de nuevas ontologías.

Ontolingua basa la construcción de ontologías en el principio de diseño modular. Esto permite que las ontologías de las librerías puedan ser reutilizadas de cuatro diferentes maneras:

- Inclusión: Una ontología A es explícitamente incluida en una ontología B.
- Polimorfismo: Una definición de una ontología es incluida en otra y refinada.
- Restricción: Una versión restringida de una ontología es incluida en otra
- Inclusión de Ciclos: Situaciones como la siguiente se pueden dar, más no son recomendables: la ontología A se incluye en la B, la ontología B se incluye en la C y la ontología C se incluye en la A.

### **4.3. Chimaera**

Chimaera es una herramienta que permite crear y mantener ontologías en la web, proporciona un ambiente distribuido para navegar, crear, editar, modificar y usar ontologías. Entre las facilidades que ofrece la herramienta se tienen: cargar bases de conocimiento en diferentes formatos, reorganizar taxonomías, resolver conflictos de nombres y editar términos. Destaca la capacidad para cargar datos de entrada en 15 diferentes formatos, tales como, KIF, Ontolingua, OKBC, Protegé, etcetera.

## **5. Algunos desarrollos ontológicos**

A continuación se describen algunos desarrollos ontológicos reportados en la bibliografía revisada:

### **5.1. UMLS (Unified Medical Language System)**

En 1986, la Biblioteca Nacional de Medicina (NLM, National Library of Medicine) comenzó un proyecto a largo plazo para construir el Sistema de Lenguaje Médico Unificado (UMLS), con el propósito de impulsar el desarrollo de sistemas computacionales para ayudar a los investigadores y profesionales del área de la salud a recuperar e integrar información biomédica desde diferentes fuentes automatizadas. Para este fin, la Biblioteca Nacional de Medicina, produce y distribuye las fuentes de conocimiento de UMLS (bases de datos) y las herramientas de software asociadas (programas), para su uso por parte de los desarrolladores en la construcción o mejoramiento de sistemas de información electrónica que crean, procesan, recuperan, integran y/o agregan datos e información biomédica y de salud. (UMLS® KNOWLEDGE SOURCES, 2005; Pastor y otros, 2001)

Las fuentes de conocimiento de UMLS son multipropósito y no están optimizadas para aplicaciones particulares, pero pueden ser utilizadas en sistemas que realizan una gama de funciones que involucran uno o más tipos de información (por ejemplo: registros de pacientes, literatura científica, lineamientos, datos de salud pública). Las herramientas de software asociadas ayudan a los desarrolladores en la personalización o utilización de las fuentes de conocimiento de UMLS para propósitos específicos. Las herramientas léxicas trabajan con más eficacia en combinación con las fuentes de conocimiento de UMLS, pero también se pueden utilizar independientemente. Todas las fuentes de conocimiento de UMLS y las

herramientas de software asociadas están disponibles de manera gratuita a través del sitio Web de la NLM (<http://www.nlm.nih.gov/>).

Existen tres fuentes de conocimiento en UMLS: el *Metathesaurus* (vocabulario), la *Red Semántica* (relaciones), y el *SPECIALIST Lexicon* (reglas del lenguaje) (UMLS® KNOWLEDGE SOURCES, 2005).

#### **- *Metathesaurus***

Es una base de datos de vocabulario multipropósito y multilingüe que contiene información sobre conceptos biomédicos y relacionados con la salud, sus distintos nombres y las relaciones entre ellos. El Metathesaurus 2004AC contiene información sobre más de 1 millón de conceptos biomédicos provenientes de más de 100 vocabularios controlados y clasificaciones (algunos en múltiples lenguajes) usados en registros de pacientes, datos de salud administrativos, bases de datos bibliográficas y sistemas expertos. En el Metathesaurus todos los vocabularios fuentes están disponibles en una única y detallada base de datos.

El Metathesaurus preserva los nombres, significados, contextos jerárquicos, atributos y relaciones entre los términos, presentes en sus vocabularios fuentes; agrega cierta información básica a cada concepto; y establece nuevas relaciones entre los términos de diferentes vocabularios fuentes.

El Metathesaurus está organizado por concepto o significado. En esencia, su propósito es enlazar nombres y vistas alternativas del mismo concepto (sinónimos, variantes léxicas y traducciones) e identificar relaciones útiles entre distintos conceptos. Todos los conceptos se asignan por lo menos a un tipo semántico de la red semántica. Muchos de los términos que aparecen en los nombres de conceptos en el Metathesaurus también aparecen en el SPECIALIST Lexicon. Las herramientas léxicas son usadas para generar la palabra, normalizarla, y normalizar la secuencia de índices en el Metathesaurus.

#### **- *Red Semántica***

El propósito de la red semántica es proporcionar una clasificación consistente de todos los conceptos representados en el Metathesaurus y un conjunto de relaciones útiles entre estos conceptos. Toda la información sobre conceptos específicos se encuentra en el Metathesaurus; la red proporciona información sobre el conjunto de tipos semánticos básicos o categorías, que se pueden asignar a estos conceptos, y define el conjunto de relaciones que pueden existir entre los tipos semánticos. El lanzamiento actual de la red semántica contiene 135 tipos semánticos y 54 relaciones. La red semántica funciona como una autoridad para los tipos semánticos que son asignados a los conceptos en el Metathesaurus.

Los tipos semánticos son los nodos en la red, y las relaciones semánticas entre ellos son los enlaces. Las agrupaciones principales de tipos semánticos incluyen organismos, estructuras anatómicas, función biológica, sustancias químicas, acontecimientos, objetos físicos, y conceptos o ideas. El alcance actual de los tipos semánticos de UMLS es muy amplio, permitiendo la clasificación semántica de una amplia gama de terminología en dominios múltiples.

#### **- *SPECIALIST Lexicon***

Contiene información sintáctica básica sobre los términos biomédicos incluidos en el Metathesaurus. Para cada palabra o término se registra la información sintáctica, morfológica, y ortográfica necesaria para el Sistema de Procesamiento de Lenguaje Natural.

Los programas léxicos se diseñan para tratar el alto grado de variabilidad en las palabras y términos del lenguaje natural. A menudo, las palabras tienen varias formas de conjugación

que podrían ser consideradas instancias de la misma palabra. Los términos de las palabras múltiples en el Metathesaurus y otros vocabularios controlados pueden tener variantes en el orden de las palabras, además de sus variantes alfabéticas y de conjugación. Las herramientas léxicas le permiten al usuario abstraerse en gran parte de los diferentes tipos de variaciones.

Las herramientas principales asociadas a UMLS son (UMLS® KNOWLEDGE SOURCES, 2005):

#### ***- Servidor de las fuentes de conocimiento de UMLS***

El servidor de las fuentes de conocimiento de UMLS (UMLSKS, UMLS Knowledge Source Server) consiste de un conjunto de herramientas interactivas basadas en Web y una interfaz de programador, para permitir a los usuarios y desarrolladores acceder a las fuentes de conocimiento de UMLS, incluyendo los vocabularios dentro del Metathesaurus.

También contiene el sitio de descargas para los archivos de datos de UMLS. El UMLSKS es el punto de partida para acceder a las tres fuentes de conocimiento desarrolladas y distribuidas por la NLM como parte del proyecto UMLS. Debido a que contiene los archivos completos del Metathesaurus, solo tienen acceso a UMLSKS los usuarios registrados que han firmado el acuerdo de licencia para el uso del UMLS Metathesaurus.

#### ***- MetamorphoSys – Programa de instalación y personalización de UMLS***

MetamorphoSys es una aplicación Java, que se debe utilizar para instalar localmente todas las fuentes de conocimiento de UMLS. Es una herramienta que apoya la creación y el refinamiento de subconjuntos personalizados del Metathesaurus. En general, el Metathesaurus se debe personalizar para ser utilizado con eficacia en aplicaciones específicas. Entre las opciones disponibles están: inclusión o exclusión de vocabularios fuentes, lenguajes y tipos de términos específicos, especificación del conjunto de caracteres de salida (7-bit ASCII o Unicode UTF-8) y del formato de salida (Formato de Lanzamiento Original o Formato de Lanzamiento Rico) para los archivos de Metathesaurus.

## **5.2. PharmGKB (The Pharmacogenetics and Pharmacogenomics Knowledge Base)**

PharmGKB (<http://www.pharmgkb.org/>) es una base de conocimiento farmacogenética y farmacogenómica construida para apoyar la representación, almacenamiento, análisis y difusión de las variaciones relevantes de los datos del genotipo y fenotipo en respuesta a los medicamentos. Los conceptos claves sobre los cuales se construye PharmGKB incluyen: genes, medicamentos, enfermedades e información fenotípica (PharmGKB, 2005).

PharmGKB es una herramienta de investigación disponible públicamente en Internet, desarrollada por la Universidad de Stanford con el financiamiento del Instituto Nacional de Salud (NIH). El proyecto se inició en Abril de 2000 y la primera versión de la base de conocimiento apareció en línea en Febrero de 2001 (Hewett y otros, 2002).

Existen muchos tipos de datos relevantes para la farmacogenética. Para facilitar su indexación y recuperación, los conjuntos de datos de PharmGKB son: 1) registrados con sus genes y/o medicamentos asociados, y 2) clasificados en cinco categorías generales de conocimiento farmacogenético (PharmGKB, 2005). Estas categorías se muestran en la Figura 6.

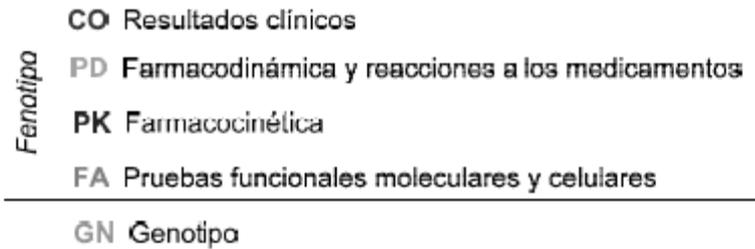


Figura 6. Categorías de conocimiento farmacogenético. (PharmGKB, 2005)

**Fenotipo:** Son las características observables de un organismo producidas por la interacción del genotipo con el ambiente. Para la farmacogenética, el “ambiente” se define a menudo como la exposición a un medicamento, aunque puede incluir otras variables también.

**Resultados clínicos:** Los datos en esta categoría demuestran que la variabilidad genética en respuesta al efecto de un medicamento puede causar diferencias considerables en los resultados clínicos, tales como, índices de curación, efectos secundarios y muerte.

**Farmacodinámica y reacciones a los medicamentos:** La farmacodinámica es el estudio de las relaciones entre la concentración de un medicamento en su sitio de acción y la magnitud del efecto biológico o fisiológico que es alcanzado. La variación genética en el área de acción de un medicamento puede causar grandes diferencias en la respuesta de un organismo al medicamento. Los datos en esta categoría documentan que las respuestas biológicas o fisiológicas a un medicamento varían, y que esta variación puede ser asociada con la variación de uno o más genes.

**Farmacocinética:** Es la ciencia que estudia el paso de los medicamentos a través del cuerpo. La variación genética en los procesos involucrados en la absorción, distribución, metabolismo, o eliminación de un medicamento puede dar lugar a cambios en la eficacia del mismo.

**Pruebas funcionales moleculares y celulares:** Una prueba molecular mide las características de una molécula (o conjunto de moléculas). Una prueba celular mide la respuesta o características de una célula (o conjunto de células). La variación genética puede alterar los resultados de estas pruebas, y esto puede estar correlacionado con las variaciones de las reacciones del organismo a los medicamentos.

**Genotipo:** Es la constitución genética de un organismo, representada por todos los genes que posee como miembro de una especie. La variación en el genotipo representa diferencias en la secuencia de genes dentro de una especie.

PharmGKB consiste de datos y una plataforma de software llamada PharmGen. El código de PharmGen está escrito en Java y basado en una arquitectura estándar de tres capas, (ver Figura 7), cuyas fortalezas principales son la flexibilidad y portabilidad. Cada capa es completamente independiente de las otras y puede ser desarrollada y modificada por separado.

La capa superior es una colección de herramientas para la manipulación y recuperación de conocimiento, que operan sobre el conocimiento de las ontologías. La capa intermedia es una base de conocimiento basada en marcos que apoya una organización jerárquica orientada a objetos de los datos. La capa inferior proporciona una base de datos relacional para el almacenamiento físico del conocimiento, que apoya las transacciones, el acceso seguro y los mecanismos para mantener la integridad de los datos.

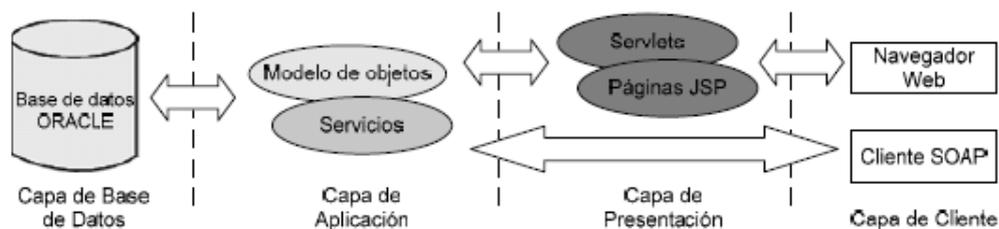


Figura 7. Arquitectura de tres capas. (PharmGKB, 2005)

PharmGKB incluye dos ontologías: una de vocabulario controlado (colección de términos organizados en una jerarquía, que intenta servir como una nomenclatura estándar que proporcione un conjunto de términos comunes para ser compartidos por múltiples usuarios y sistemas) y otra de dominio (contiene el esquema de la base de conocimiento, es decir, el conjunto de clases y las propiedades que las caracterizan). Los investigadores utilizaron Protégé para el desarrollo de las ontologías y de la base de conocimiento (Oliver y otros, 2002).

Los investigadores registrados pueden enviar diferentes tipos de datos que contienen información sobre genes, medicamentos, enfermedades, entre otros. Los envíos se pueden realizar a través de formularios Web o de archivos XML. Una vez que un envío ha sido validado y aceptado, se integra en la base de conocimiento existente de PharmGKB donde se hace accesible

### 5.3. Ontología para mejorar los sistemas de bases de datos de terrorismo

Esta ontología asiste a personas que se encargan de prevenir y mitigar actos de terrorismo como investigadores y agencias policiales, en tareas como identificación, agrupamiento y clasificación de documentos en el dominio del terrorismo. La ontología proporciona la información necesaria para la creación y mejoramiento de sistemas de bases de datos de terrorismo (Gruenwald y otros, 2003).

Alguna de las funcionalidades de esta ontología son: búsqueda por palabras claves exactas, búsqueda por palabras que tienen el mismo sentido, palabras cercanas a otras en el texto, búsqueda por asuntos específicos en documentos, entre otras. Además, contiene información de ejecutores, objetivos físicos y armas utilizadas en actos terroristas. La ontología fue desarrollada a partir de una base de conocimiento de textos realizada en Oracle.

### 5.4. Ontologías desarrolladas por STARLab

Este laboratorio de investigación (Systems Technology and Applications Research Laboratory, Faculty of Sciences, Vrije Universiteit Brussel) (<http://www.starlab.vub.ac.be/website/projects>) lleva adelante el desarrollo de un conjunto de ontologías, entre los que destacan las siguientes:

- *BonaTema*: Sistema de ontología y base de conocimiento, que contiene información biológica sobre la levadura.

- *OntoWeb*: Ontología cuyo objetivo es proporcionar a los investigadores y a las industrias que prestan servicios de ontologías, el intercambio de información en áreas como recuperación de información, administración de conocimiento, bioinformática y comercio electrónico.
- *FFPOIROT*: Esta ontología, que estará disponible en varios idiomas (francés, inglés, alemán e italiano), proporcionará recursos de información orientados a la prevención de fraudes para dominios de finanzas.
- *Hipermuseum*: Ontología basada en Internet para apoyar el uso activo de datos digitales de museos en el aprendizaje y en presentaciones.
- *Ontobasis*: Fundamentos, construcción, servicios y aplicaciones de ontologías

## 6. WEB Semántica

En los últimos años, la Web se ha convertido en una parte importante de la vida cotidiana de una gran cantidad de personas, cambiando la forma en que se comunican, hacen negocios y realizan su trabajo. La Web es un medio extraordinariamente flexible y económico para la comunicación, el comercio y los negocios, ocio y entretenimiento, acceso a información y servicios, educación y difusión de cultura, entre otras cosas; que puede ser utilizado en cualquier momento e independientemente de la ubicación geográfica. Su propiedad esencial es la universalidad (Castells, 2003; Hossain y El Saddik, 2004; Berners-Lee y otros, 2001).

La World Wide Web contiene grandes cantidades de información, que ha sido creada por distintas organizaciones, comunidades e individuos para diferentes propósitos. Los usuarios pueden acceder fácilmente a esta información, especificando la dirección de un recurso, o utilizando un motor de búsqueda y siguiendo los enlaces para encontrar otros recursos relacionados. (Koivunen y Millar, 2001)

El éxito de la Web se basa en su simplicidad y disponibilidad. Sin embargo, debido al crecimiento inmenso del número de usuarios, se hace cada vez más difícil organizar, localizar e integrar el conocimiento disponible; originando que algunas tareas requieran un tiempo excesivo para las personas o simplemente se encuentren fuera del alcance de las mismas.

En la actualidad, la mayor parte del contenido que se encuentra en la Web está diseñado para ser entendido por las personas, no para que pueda ser manipulado eficientemente por programas computacionales (Berners-Lee y otros, 2001). La información presente en la Web se encuentra estructurada mediante lenguajes de etiquetado, que únicamente describen la forma en que dicha información debe ser mostrada al usuario por el navegador, pero no expresan nada sobre su significado (semántica). Muchos de los buscadores actuales, basados en palabras claves, suelen devolver información irrelevante y no reconocen palabras diferentes que tengan el mismo significado de la palabra buscada.

La Web Semántica propone superar las limitaciones de la Web actual mediante la introducción de descripciones explícitas del significado, la estructura interna y la estructura global de los contenidos disponibles en la World Wide Web. La Web Semántica es una extensión de la Web actual, en la cual la información tiene un significado bien definido para permitir que las computadoras y las personas trabajen cooperativamente (Berners-Lee y otros, 2001).

Se trata de una corriente promovida por el propio inventor de la Web y presidente del consorcio W3C, Tim Berners-Lee, cuya finalidad es lograr que las máquinas puedan entender y utilizar el contenido de la Web sin necesidad de la intervención humana. Esta nueva Web estaría formada por agentes inteligentes de software capaces de navegar y realizar operaciones por las personas para ahorrarles trabajo y optimizar los resultados. Las tecnologías de la Web Semántica buscan desarrollar una Web más cohesionada, donde sea más fácil localizar, compartir e integrar la información y los servicios, para aprovechar al máximo los recursos disponibles (Castells, 2003).

Para que la Web Semántica funcione, las computadoras deben tener acceso a colecciones estructuradas de información y conjuntos de reglas de inferencia, que puedan utilizar para llevar a cabo el razonamiento automático. Estas necesidades se pueden satisfacer utilizando ontologías.

La infraestructura tecnológica necesaria para llevar a cabo el proyecto de la Web Semántica se muestra en la Figura 8.

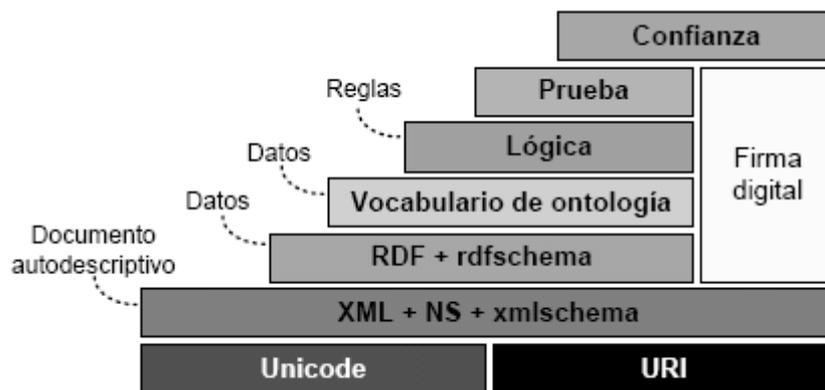


Figura 8: Arquitectura de la Web Semántica (Berners-Lee y otros, 2001).

- **Unicode:** es una norma de codificación de caracteres cuyo objetivo es asignar a cada posible carácter de cada posible lenguaje un número y nombre único. Unicode garantiza la utilización de conjuntos de caracteres internacionales.
- **URI (Uniform Resource Identifier):** texto corto que identifica unívocamente recursos tales como servicio, página, documento, enciclopedia, etcetera, accesible en una red. Normalmente consta de dos partes: un identificador del método de acceso o protocolo al recurso, por ejemplo http:, mailto:, ftp:, y el nombre del recurso.
- **XML + NS + xmlschema:** La capa XML, junto con las definiciones de espacios de nombres (NS) y esquemas, asegura que se puedan integrar las definiciones de la Web Semántica con los otros estándares basados en XML:

**XML:** aporta una sintaxis para los documentos estructurados, pero no les aporta ningún tipo de restricción sobre el significado.

**NS (NameSpaces):** se encargan de asociar elementos y atributos de nombres usados en XML con espacios de nombre identificados por referencias URI. Con esto se logra que los elementos cuya denominación es la misma puedan ser diferenciados por el NS al cual están asociados.

**XMLSchema:** es lenguaje el cual define la estructura del XML. Al tener un documento XML asociado a un XMLSchema se puede saber si este es un XML válido y bien formado.

- **RDF + rdfschema:** es una capa que permite dotar de mayor significado al modelo, debido a que esta basado en un lenguaje que permite expresar y definir diferentes ideas en forma de Tripleta (sujeto, predicado, objeto). Con RDF y RDFS es posible hacer declaraciones sobre los objetos y definir vocabularios, que pueden ser referenciados a través de un URI.

**RDF:** es un modelo de datos para objetos ("recursos") y para las relaciones entre ellos, provee una semántica simple para este modelo de datos, a la vez que este modelo de datos puede ser representado en sintaxis XML. Es un modelo común (Framework) que permite hacer afirmaciones sobre los recursos (Description) y que hace posible que estos recursos pueden ser nombrados por URIs (Resource).

**RDF Schema:** es un vocabulario para describir propiedades y clases de recursos RDF, con una semántica para generalización de jerarquías de aquellas propiedades y clases.

- **Lenguaje de Ontologías:** son los lenguajes que permiten representar las ontologías y posibilitan la utilización de las mismas por parte de entes computarizados. Esta capa apoya el desarrollo de vocabularios y la definición de relaciones entre los diferentes conceptos.

- **Lógica:** además de las ontologías, se necesitan reglas de inferencia, que permitan obtener conclusiones a partir de ciertos valores dados.

- **Pruebas:** Es necesario realizar pruebas para conocer si las reglas que se han definido son del todo correctas, para así ver el funcionamiento de la Web.

- **Confianza:** los entes encargados de recolectar información de la Web, deben ser muy escépticos en cuanto a lo encontrado, hasta que no confirmen las fuentes de la información recolectada.

- **Firma Digital:** es un conjunto de datos asociados a un mensaje digital que permite garantizar la identidad del firmante y la integridad del mensaje. Con esta capa se pueden detectar alteraciones en los documentos

En la Web Semántica, se utilizan metadatos para describir páginas Web, bases de datos, modelos, conceptos y otros recursos Web. Algunos de los beneficios que proporcionará la Web Semántica son: mecanismos de búsqueda mejorados, interfaces de usuarios adaptativas, servicios Web eficaces, filtración colaborativa de la información, integración flexible de datos de dominios múltiples, administración eficiente del conocimiento y deducción de hechos adicionales que no están explícitamente definidos. La Web Semántica supone el paso de la información al conocimiento (Hossain y El Saddik, 2004).

Los Servicios Web Semánticos son una línea importante de la Web Semántica, que propone describir no sólo información sino definir ontologías de funcionalidad y procedimientos para especificar servicios Web: sus entradas y salidas, las condiciones necesarias para que se puedan ejecutar, los efectos que producen, o los pasos a seguir cuando se trata de un servicio compuesto. Estas descripciones procesables por máquinas permitirán automatizar el descubrimiento, la composición y la ejecución de servicios, así como la comunicación entre unos y otros (Castells, 2003).

## **7. Un caso de estudio en la construcción de ontologías: Ontología del dominio de los insectos acuáticos.**

Con la finalidad de poner en práctica y aplicar los conceptos expuestos anteriormente, en esta sección se presenta la construcción de una ontología para el dominio de los insectos acuáticos (Chaudary y Scarioni, 2005), desarrollada con la metodología Methontology, e implementada con el editor Protégé. Este desarrollo está enmarcado en el proyecto de investigación financiado por el Consejo de Desarrollo Científico y Humanístico de la Universidad Central de Venezuela titulado: **Sistema basado en conocimiento para la identificación de Especies de Insectos Acuáticos: Orden Plecóptera, Género Anacroneuria**, e identificado con el número **PG. 3.13.5405.04**

### **7.1. Descripción del dominio de los insectos acuáticos.**

El crecimiento de las ciudades trae como consecuencia cambios en el medio ambiente. Como resultado, las características naturales se ven afectadas para dar paso al desarrollo industrial con la consecuente alteración de zonas verdes y cursos de agua. Una de las consecuencias de un desarrollo acelerado, no planificado y sin criterios de preservación del ambiente, es la contaminación atmosférica y de los cuerpos de aguas continentales (ríos y lagos), y no continentales (océanos). Resulta indispensable medir los niveles de contaminación en las ciudades, a fin de preservar la calidad de vida de sus habitantes, por lo tanto es necesario medir frecuentemente los niveles de sustancias químicas en el aire y en el agua. En esta última es importante determinar el componente biológico para impedir la propagación de enfermedades por el consumo del agua (Cressa, 1994).

Por lo anteriormente expuesto se tiene que el monitoreo de la calidad del agua es indispensable para la preservación de la salud pública. En consecuencia, se han implementado una serie de métodos biológicos que permiten cuantificar los cambios que se producen en los sistemas acuáticos. En este sentido, en los últimos años se han venido utilizando a los insectos acuáticos como indicadores de contaminación en ríos y lagunas; la presencia, diversidad y abundancia de estos, refleja el nivel de contaminación.

Dada la importancia que los insectos acuáticos tienen en estos ecosistemas, se hace necesario conocer su dinámica poblacional (ciclo de vida, reproducción, alimentación, emergencia, entre otros) y es por esto que el Laboratorio de Ecología de Sistemas Acuáticos Continentales, del Instituto de Biología Experimental (IBE), de la Facultad de Ciencias, de la Universidad Central de Venezuela, se ha planteado como objetivo principal el estudio de estos insectos. Un aspecto a considerar en este estudio consiste en la realización del inventario de los insectos que integran las comunidades de macroinvertebrados de los diferentes ríos del país, con especial énfasis en los Parques Nacionales. Este inventario permitirá mantener un registro de los componentes de dicha comunidad así como la cuantificación de los cambios que en ella se presenten (ciclo de vida, reproducción, alimentación, emergencia, entre otros). No obstante, la identificación de los macroinvertebrados en los laboratorios se ve limitada por la poca bibliografía existente sobre insectos tropicales y la falta de material referencial acerca de los insectos presentes en Venezuela. Igualmente, la información disponible se encuentra dispersa y a veces en medios de difícil acceso, haciendo que a muchos de los investigadores se les dificulte el proceso de estudio e investigación.

El desarrollo de una ontología para este dominio permitiría solventar en parte el problema de tener la información en diferentes lugares, proporcionando un repositorio en común de los conceptos y relaciones del dominio, donde se logre una estandarización del vocabulario,

compartir información ya sea entre los expertos en el área u otros sistemas basados en conocimiento, además de ser una guía rápida de referencia y consulta para la investigación.

Además, una ontología resultará de gran importancia ya que propone un sistema de almacenamiento para usar y compartir información entre los expertos del Laboratorio de Ecología de Sistemas Acuáticos Continentales, y la comunidad de ecólogos en general, facilitando su trabajo y reduciendo los tiempos de identificación ya que la información va a estar centralizada, disponible y accesible en un solo lugar. De esta manera, cualquier persona interesada en el dominio de los insectos acuáticos podrá encontrar en dicha ontología el conocimiento relacionado con estos insectos, sin necesidad de pasar largas horas buscando y consultando material localizado en diferentes lugares.

La estandarización del vocabulario también será una ventaja de esta ontología, ya que con estas se evitan confusiones de nomenclatura, lo cual es de gran importancia en este dominio debido a que las características de un insecto deben referenciarse de manera única para garantizar identificaciones precisas y confiables.

## 7.2. Tareas de conceptualización de Methontology:

A continuación se describen las tareas de conceptualización de la metodología Methontology, la cual fue seleccionada para desarrollar la ontología.

**Tarea 1: Construir el Glosario de Términos:** Este glosario está constituido por los términos de interés del dominio (conceptos, instancias, atributos, relaciones entre conceptos, etcétera.) y sus descripciones en lenguaje natural. En la Tabla 1 se muestra un extracto de este glosario.

Tabla 1. Extracto del glosario de términos

Nombre	Descripción
Insecto	Artrópodo de respiración traqueal, con el cuerpo dividido distintamente en cabeza, tórax y abdomen, con un par de antenas y tres pares de patas. Algunos tienen uno o dos pares de alas y sufren metamorfosis durante su desarrollo.
Artrópodos	Son animales invertebrados, de cuerpo con simetría bilateral, cubierto por cutícula, formado por una serie lineal de segmentos más o menos ostensibles y provistos de apéndices compuestos de piezas articuladas o artejos
Insectos Acuáticos	Insectos que viven en ambientes acuáticos
Morfología	Parte de la biología que trata de la forma de los seres orgánicos y de las modificaciones o transformaciones que experimenta
Morfología Externa	Forma del individuo en el exterior. En el caso de los insectos, en el estado adulto el cuerpo de los insectos esta dividido en tres partes; cabeza, tórax y abdomen.
Morfología Interna	La morfología interna comprende: el aparato digestivo, el sistema circulatorio, el sistema respiratorio, el sistema nervioso, y el aparato reproductor.
Metamorfosis Completa	La metamorfosis completa o desarrollo holometábolo, se caracteriza porque la forma inmadura del insecto es diferente al adulto tanto en la morfología como en la ecología. El insecto inmaduro es

	totalmente diferente morfológicamente al adulto. En general, los insectos con metamorfosis completa tienen cuatro estados bien definidos: huevo, larva, pupa y adulto. La pupa es un estado intermedio entre la larva y el adulto, y se caracteriza por ser inactiva.
Metamorfosis Incompleta	La metamorfosis incompleta o desarrollo hemimetábolo, se caracteriza porque los individuos jóvenes y adultos, son muy similares en forma, excepto por el tamaño. Las alas se desarrollan externamente durante los estados inmaduros. La metamorfosis incompleta consta de tres estados: huevo, larva o ninfa, y adulto.
Cabeza	En ella se encuentran importantes estructuras que son utilizadas para funciones sensitivas: ojos compuestos, ocelos, antenas y piezas bucales. La morfología de estas estructuras puede ser característica de las diferentes especies.
Tórax	En la región media del cuerpo denominado tórax, se encuentran las patas articuladas y las alas. El tórax se divide en tres segmentos: el protórax soporta las patas delanteras, el mesotórax las patas del medio y las alas delanteras y el metatórax las patas traseras y las alas traseras en caso de que existan.
Abdomen	En el abdomen se encuentran los órganos reproductivos y un importante número de aberturas respiratorias. Es la última región del cuerpo de los insectos.

**Tarea 2: Construir una Taxonomía de Conceptos:** En la Figura 9 se observa la taxonomía de conceptos para la ontología desarrollada, en la cual se define una jerarquía de conceptos, elaborada a partir del glosario de términos y conjuntamente con los expertos del dominio.

**Tarea 3: Construir un diagrama de relaciones binarias:** En este diagrama se establecen los tipos de relaciones entre los conceptos especificados en la taxonomía. En la Figura 10 se muestra un fragmento de éste en el cual se presentan las relaciones entre el concepto *Abdomen* y los conceptos *Alas*, *Aparato Sexual*, *Placas Abdominales* y *Branquias*.

**Tarea 4: Construir el diccionario de conceptos:** El diccionario de conceptos contiene los conceptos más importantes del dominio, según la opinión de los expertos, sus relaciones, instancias, atributos de clases y atributos de instancias. En la Tabla 2 se presenta una porción de éste.

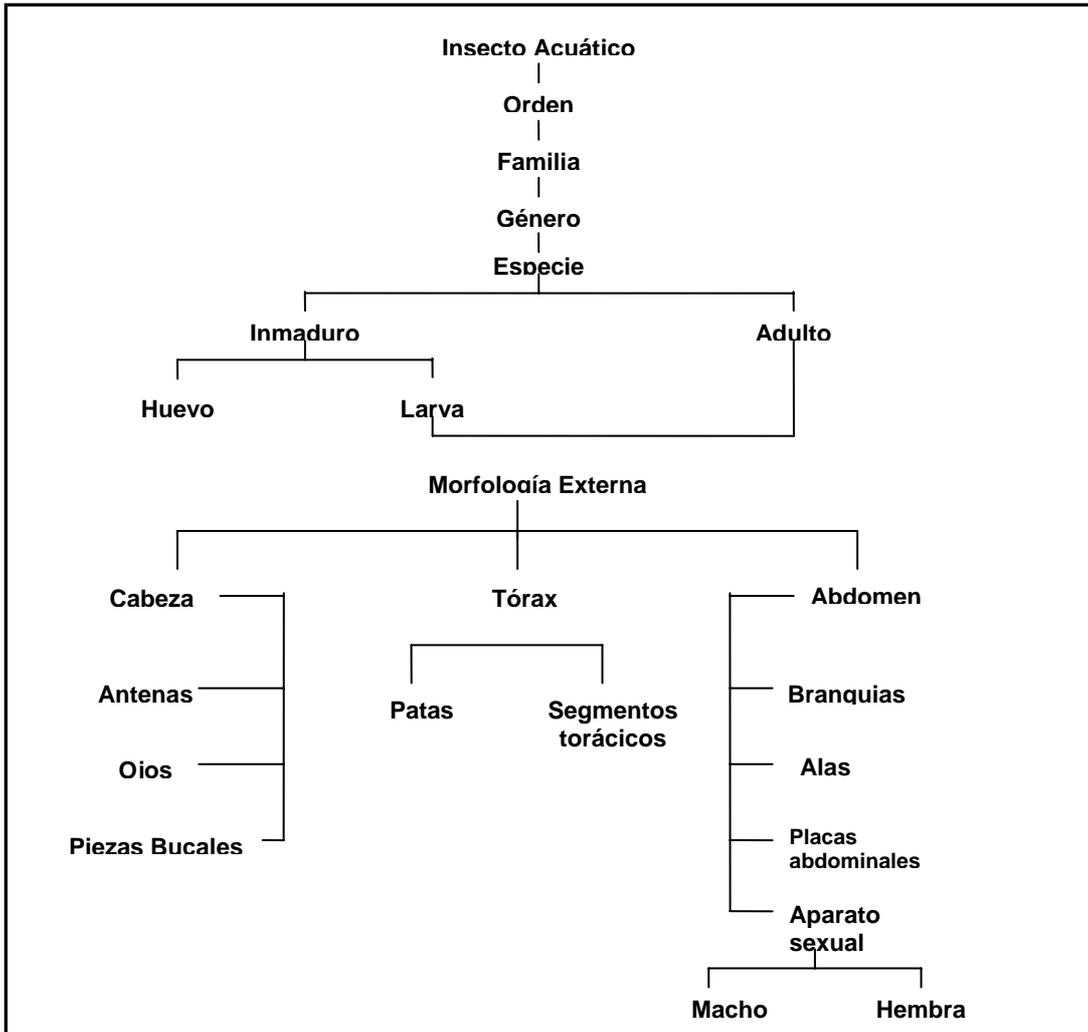


Figura 9. Taxonomía de conceptos

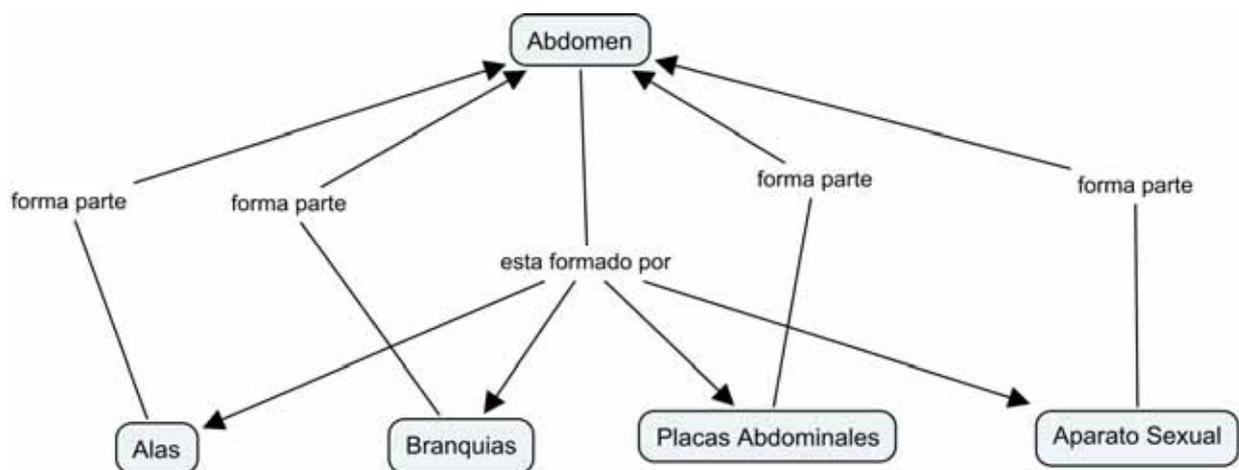


Figura 10. Extracto del diagrama de Relaciones Binarias

Tabla 2. Porción del diccionario de conceptos

Nombre	Instancias	Atributos de Clases	Relaciones
Orden	Coleoptera, Diptera, Efemeroptera, Lepidoptera, Megaloptera, Odonata, Plecoptera, Tricoptera.	Familias	pertenece
Familia	Perlidae	Géneros	pertenece
Género	Anacroneuria, Enderleina, Macrogynoplax.	Características Generales Color del Cuerpo Especies	pertenece

**Tarea 5. Definir las relaciones binarias en detalle:** Las relaciones a detallar, mediante una *tabla de relaciones binarias*, son aquellas especificadas en el diccionario de conceptos. Para cada relación se especifica el nombre, conceptos fuente y destino, cardinalidad y relación inversa. Un fragmento de esta tabla que muestra las relaciones *forma parte de*, *pertenece* y *esta formado por*, se presenta en la Tabla 3.

Tabla 3. Extracto de la Tabla de relaciones binarias

Nombre	Concepto Origen	Concepto Destino	Cardinalidad	Relación Inversa
pertenece	Familia	Orden	N:1	tiene
	Género	Familia	N:1	
	Especie	Género	N:1	
	Inmaduro	Especie	N:1	
	Adulto	Especie	N:1	
forma parte de	Cabeza	Morfología Externa	1:1	está formada por
	Tórax	Morfología Externa	1:1	
	Abdomen	Morfología Externa	1:1	
está formado por	Abdomen	Alas	1:1	forma parte de
		Aparato Sexual	1:1	
		Placas Abdominales	1:1	
		Branquias	1:1	

**Tarea 6. Definir los atributos de instancias en detalle:** Para la Ontología de Insectos Acuáticos, existe una clase del tipo concreto llamada clase Insecto Acuático, que es la clase más general de la jerarquía que no acepta especializaciones. Todas las demás clases son del tipo abstracto, es decir, todas tienen por lo menos una instancia.

Para todas las clases abstractas, los atributos de las clases son los mismos que los de las instancias creadas para una clase determinada. Esto quiere decir que no existe una diferenciación entre los atributos, ya que todas las clases son de la misma forma.

Por todo lo anterior, no se tienen atributos de instancias para esta Ontología, los atributos de las clases son al mismo tiempo los atributos de las instancias.

**Tarea 7. Definir los Atributos de Clases en detalle:** Los atributos de las clases son las cualidades o características que van a tener cada una de las clases de la ontología. Para cada

atributo de clase se describe el nombre, el concepto o clase al cual corresponde, el tipo de dato, el rango de posibles valores a tomar o valor predeterminado (si lo tiene), y la cardinalidad para cada atributo, ver Tabla 4.

Tabla 4. Extracto de la Tabla de atributos de clases

Nombre	Concepto	Tipo de Dato	Rango Valor	Cardinalidad
Ciclo de Vida	Insecto Acuático	String	Parte del Ciclo de Vida se realiza en el Ecosistema Acuático.	1:N
Orden	Orden	String		1:1
Familias que componen el Orden	Orden	Instance	Familia	1:N

**Tarea 8. Definir en detalle las constantes:** Las constantes especificadas en el glosario de términos, se definen utilizando una *tabla de constantes*. Para cada constante se especifica el nombre, tipo de valor, valor y unidad de medida (para constantes numéricas). Esta tarea no aplicó para este dominio.

**Tarea 9. Definir los axiomas formales:** Los axiomas necesarios en la ontología se describen con precisión en una tabla. Para cada definición de un axioma formal se especifica el nombre, descripción, expresión lógica que lo describe (usando lógica de primer orden), los conceptos, atributos y relaciones binarias a los cuales el axioma hace referencia y las variables utilizadas. La Tabla 5 muestra algunos de estos axiomas.

Tabla 5. Algunos axiomas de la ontología

Nombre	Descripción	Concepto	Relación
Un solo Orden	Un insecto solo puede pertenecer a un solo Orden.	Insecto Acuático	
Pertenencia	Una Familia solo puede pertenecer a un Orden, un Género a una Familia y una Especie a un Género.	Orden Familia Género Especie	Pertenece
Morfología Externa de un Insecto Acuático	Un Insecto Acuático tiene tantas morfologías externas como estados en su vida tenga.	Insecto Acuático Inmaduro Adulto Morfología Externa	Tiene

**Tarea 10. Definir las reglas:** Las reglas necesarias en la ontología, se definen en una *tabla de reglas*. Para cada regla se especifica el nombre, descripción, expresión que formalmente la describe y conceptos a los que hace referencia. Para su especificación se sugiere la forma: *Si* <condiciones> *entonces* <consecuencias o acciones>. Algunas de las reglas que fueron definidas para la ontología, se muestran en la Tabla 6.

Tabla 6. Algunos axiomas de la ontología

Nombre	Expresión	Concepto	Relación
¿Cuales Órdenes de insectos no tienen alas en las larvas?	Match All:true Inmaduro presenciaAlasRudimentarias is Other:No	Inmaduro	es un
¿Cuales Órdenes tienen un solo par de alas y el otro para equilibrio (halterios)?	Match All:true Adulto presenciaAlas contains Other:1	Adulto	
¿Cuales Órdenes tienen Alas Rudimentarias como Inmaduros (Larva)?	Match All:true Inmaduro presenciaAlasRudimentarias is Other:Si	Inmaduro	es un

**Tarea 11: Definir las Instancias:** En esta tarea se definen las instancias que han sido identificadas para cada clase en la ontología. Se muestra por medio de una tabla, su nombre, la clase o concepto al cual pertenece, los atributos y los valores de esos atributos. En la Tabla 7 se encuentra el Orden Coleoptera

Tabla 7. Extracto de la Tabla de instancias para el orden Coleoptera

Nombre	Concepto	Atributo	Valor
Coleoptera	Orden	Orden	Coleoptera
		Familias que componen el Orden	
Coleoptera	Adulto	Características del Aparato Reproductor	Desarrollado
		Características del Cuerpo	Cuerpo altamente esclerotizado y corto.
		Pares de Patas	3 pares de patas no fusionadas al cuerpo.
		Presencia de Alas	2 pares de alas.
		Cuerpo Quitinoso	
Coleoptera (Adulto)	Morfología Externa	Morfología Externa de	Coleoptera (Adulto)
Coleoptera (Adulto)	Cabeza	Orientación de la Cabeza	Hypognato
		Color de la Cabeza	
Coleoptera (Adulto)	Antenas	Estructura Basal	Si
		Tamaño	Bien desarrolladas
		Flagelo	11 o menos segmentos
		Pedicelo	Si
		Forma	Variables: claviforme, aserrada, etc.
Coleoptera (Adulto)	Ojos	Ocelli	2 o 3
		Ojos Compuestos	Desarrollados
		Ojos Simples	Desarrollados
Coleoptera	Piezas	Desarrollo de Piezas Bucales	Bien desarrolladas

(Adulto)	Bucales		
		Maxila	Si, con palpos con 3 segmentos
		Mandíbulas	Si, bien desarrolladas y lóbulo molar ausente en predadores. Son usadas para masticación.
		Labium	Si
		Labrum	Si

### 7.3. Codificación de la ontología con Protégé

Una vez finalizada la aplicación de las tareas de Methontology, se obtuvo como resultado el modelo conceptual del dominio de los insectos acuáticos. En este modelo se representan los ocho órdenes de insectos acuáticos existentes en Venezuela, además se describe amplia y completamente al Orden Plecoptera y sus 30 especies. El modelo contabiliza en total, veinte y cuatro (24) clases, setecientas diez y siete (717 ) instancias y ciento veinte (120) atributos.

El siguiente paso en el desarrollo de la ontología consistió en codificar el modelo generado en un lenguaje ontológico; para esto se seleccionó la herramienta de edición Protégé.

A continuación se presentan algunas pantallas del editor de ontologías Protégé, que describen parte del proceso de construcción de la Ontología de Insectos Acuáticos. La Figura 11 muestra la pantalla principal del editor, en la cual se observa el área ocupada por la taxonomía de conceptos en la parte izquierda de la pantalla, y el área de los atributos para esas clases, en la parte derecha de la pantalla. Además de un área destinada para mostrar las características generales de las clases como su nombre, su rol, la documentación, entre otras.

La Figura 12 muestra la pantalla de edición de instancias. En el área de la derecha se encuentra la jerarquía de clases de la ontología. En el área central se muestran las instancias para cada una de las clases pertenecientes a la taxonomía de conceptos (desarrollada en la Tarea 11 de Methontology). El área de la derecha muestra el editor de instancias, en donde se introduce la información particular para una instancia determinada. La edición de las instancias consiste en asignar valores a los atributos creados para cada clase (Tarea 7 de Methontology).

Con la finalidad de que la ontología pudiese ser accedida y recorrida (navegada) desde diferentes puntos geográficos por la comunidad interesada en este dominio, se desarrolló una aplicación con orientación a la Web, utilizando el *plug-in* de Protégé, denominado Web Protégé Browser, la aplicación proporciona además funcionalidades como búsquedas, visualización de imágenes y gráficos y solución a interrogantes por parte de los usuarios.

El Web Protégé Browser posee las siguientes características técnicas:

- a) Clases desarrolladas con el lenguaje de programación Java y
- b) Páginas Web con tecnología JSP y HTML con el uso de Frames.

Dado que el Protégé Web Browser esta desarrollado sobre Java y JSP, la construcción de la aplicación Web se limitó a la construcción de páginas JSP, unas cuantas clases de propósito general (en su mayoría existentes) y unos pocos Servlets. Para la interfaz Web, se restringió al uso de Swing y HTML. Se utilizó la API de Protégé ya que ésta contiene las fuentes necesarias para la construcción de aplicaciones que requieran manipular archivos Protégé de repositorios de conocimiento.

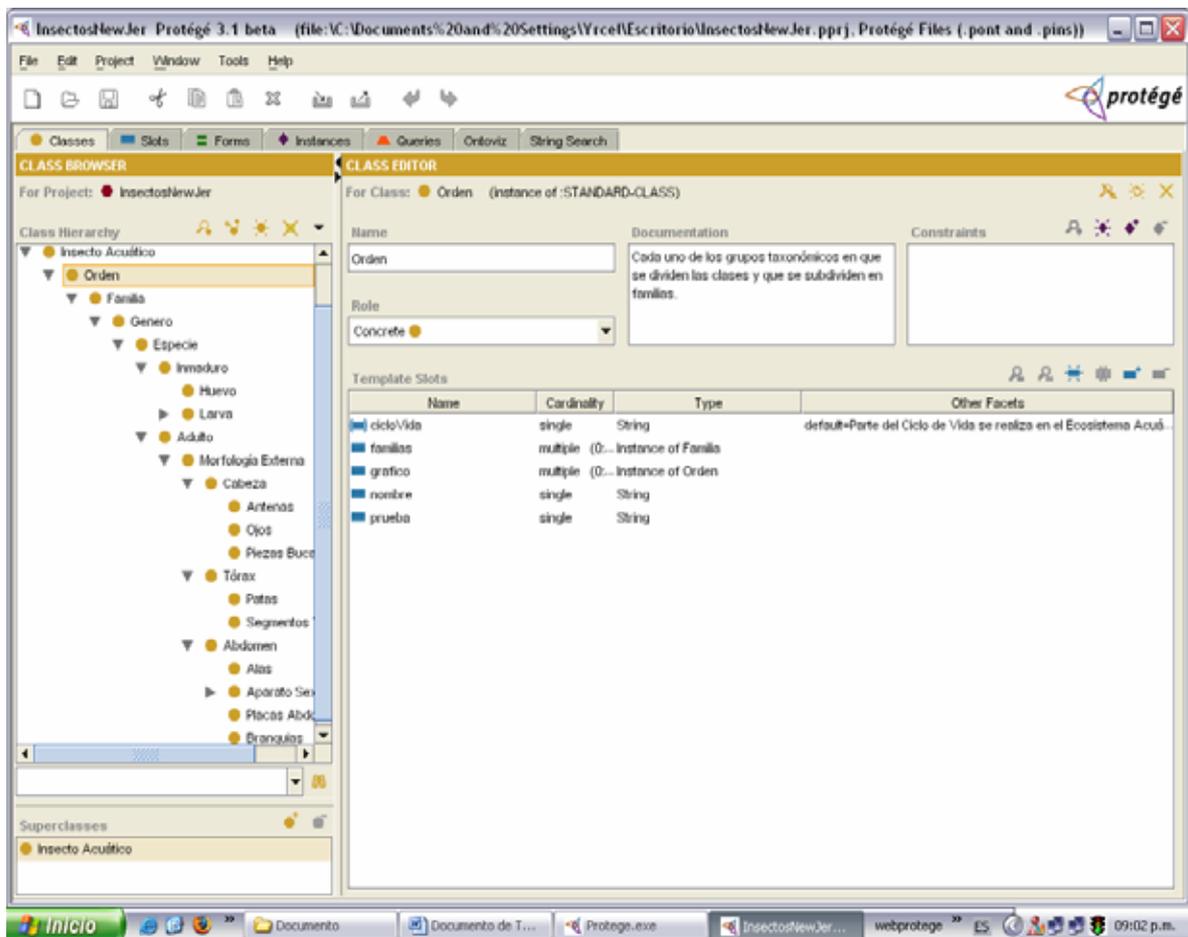


Figura 11. Página del editor Protégé: taxonomía de conceptos.

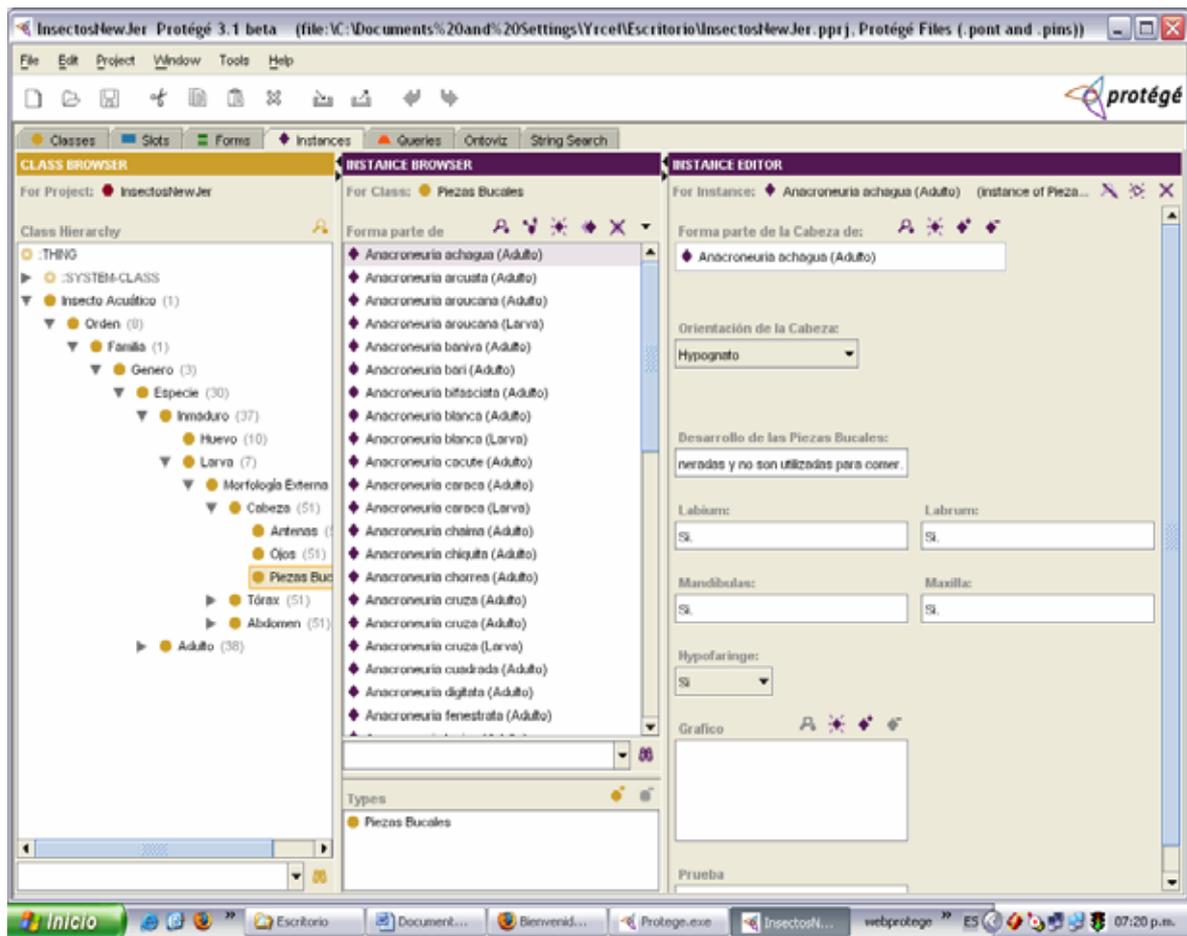


Figura 12. Creación de las instancias de la ontología

A continuación se presenta las principales clases java utilizadas en la aplicación:

- KBManager: Esta clase controla la apertura y los accesos a la base de conocimiento, sus funciones y la mayoría de las opciones alrededor de ésta.
- KBTreeModel: Clase encargada de establecer las distintas configuraciones por las que va pasando la aplicación. Esta clase manipula el flujo de datos ya que es la encargada de determinar las posiciones actuales dentro de la jerarquía de clases e instancias, determinando las ramas del árbol y las opciones que surgen al seleccionar las distintas opciones.
- MyQueryClass: Esta clase contiene la lógica de consultas de la aplicación. La sección de preguntas de la aplicación se lleva a cabo a través de ésta, tanto para la concepción de la pregunta como para las respuestas.
- GraphImageServlet: Conjuntamente con la clase ImageUtilities se encargan de darle apariencia a la aplicación. Está creada con código Swing.

En la Figura 13 se observa la página principal de la aplicación Web que identifica el sitio y permite acceder a las diferentes funcionalidades que éste ofrece y a la navegación y recorrido por la ontología.

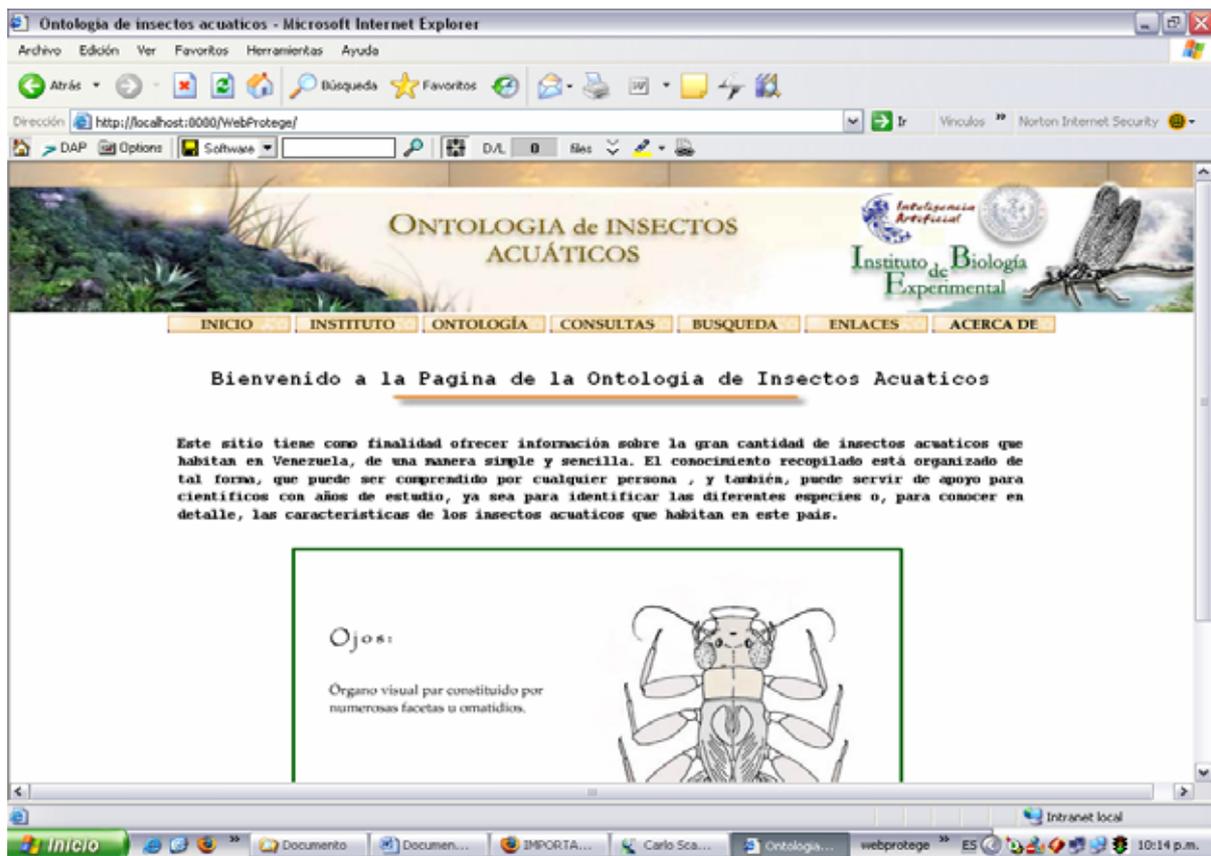


Figura 13. Página Principal de la aplicación Web.

En la Figura 14 se observa un aspecto de la navegación de la ontología que muestra la descripción de un concepto particular (Orden).

## Conclusiones.

En este documento se logró recopilar información acerca de los aspectos básicos y generales de las ontologías. Se presentan un conjunto de definiciones dadas al término desde sus orígenes filosóficos, hasta su uso en inteligencia artificial; los componentes y tipos de ontologías existentes, se revisaron también algunas metodologías usadas en desarrollos ontológicos y lenguajes y editores para su formalización.

Un aspecto de interés que permitió aplicar y entender los conceptos antes expuestos, fue la descripción detallada del desarrollo de una ontología en un dominio biológico. Se especificó ampliamente como se aplicó la metodología de construcción Methontology, la utilización del editor de ontologías Protégé, y la implementación de una aplicación Web que permite, a la comunidad interesada, acceder a través de Internet al conocimiento almacenado en la ontología desarrollada.

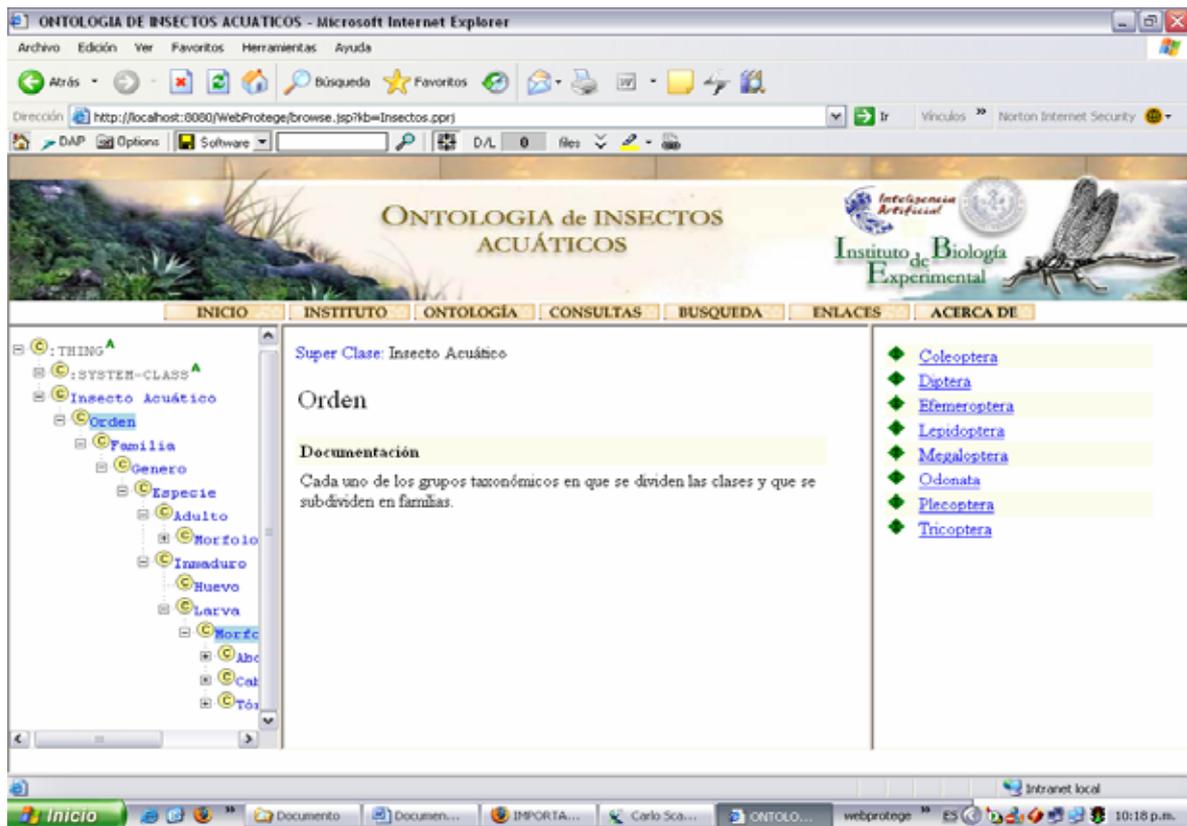


Figura 14. Navegación de la ontología: Consulta del concepto Orden.

## Referencias

- Alonso Amparo, Güijarro Bertha, Lozano Tello Adolfo, Palma José y Tabeada Maria. (2004). Ingeniería del Conocimiento Aspectos Metodológicos. Pearson Prentice Hall. ISBN84-205-4192-3. España.
- Berners-Lee Tim, Hendler James y Lassila Ora. (2001). The Semantic Web. Scientific American, Mayo 2001. pp. 28-37. Disponible en fecha Enero de 2007 en: <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-4A9809EC588EF21>
- Castells, Pablo. (2003). La Web Semántica. Escuela Politécnica Superior, Universidad Autónoma de Madrid. Conferencia impartida en el Curso de Verano sobre Interacción en la Web, Universidad de Castilla - La Mancha, Puertollano, Septiembre 2003. Disponible en fecha Febrero de 2007 en: <http://www.ii.uam.es/~castells/publications/castells-uclm03.pdf>.
- Ceccaroni Luigi. (2001). Onto WEDSS- An ontology-Based environmental decision-support system for the management of wastewater treatment plants. Disponible en fecha Febrero 2007 en: <http://www.angelfire.com/scfi2/technopapa/Thesis.htm>

- Chaudary Y, Scarioni C. (2005). Ontología para el dominio de insectos Acuáticos. Trabajo Especial de Grado. Escuela de Computación. Facultad de Ciencias Universidad Central de Venezuela. Caracas, Venezuela.
- Corcho Oscar, Fernández López Mariano, Gómez Pérez Asunción, López Cima Ángel. (2005). Building legal ontologies with METHONTOLOGY and WebODE. Law and the Semantic Web. Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications. Springer-Verlag, LNAI 3369. Disponible en fecha Febrero de 2007 en: [http://www.cs.man.ac.uk/~ocorcho/documents/LawSemWeb2004\\_CorchoEtAl.pdf](http://www.cs.man.ac.uk/~ocorcho/documents/LawSemWeb2004_CorchoEtAl.pdf)
- Cressa, Claudia. (1994). Structural changes of the macroinvertebrate community in a tropical river. Verhandlungen Der Internationalen Vereinigung Für Theoretische Und Angewandte Limnologie . Venezuela.
- De Bruijn, Jos. (2003). Using Ontologies. Enabling Knowledge Sharing and Reuse on the Semantic Web. Digital Enterprise Research Institute (DERI) Technical Report DERI-2003-10-29. Austria. Disponible en fecha Enero 2007 en: <http://www.debruijn.net/publications/DERI-TR-2003-10-29.pdf>
- Devedzic Vladan. (2002). Understanding ontological Engineering. University of Belgrade. Yugoslavia. School of business Administration Department of Information Systems. Yugoslavia Communications of the ACM. Volume 45. Issue 4 Disponible en fecha Febrero 2007 en: <http://fon.fon.bg.ac.yu/~devedzic/CACM2002.pdf>
- Fensel D, Horrocks I, Van Harmelen F, Decker S, Erdmann M y Klein M. (2000). OIL in a Nutshell. In Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management (EKAW-2000), R. Dieng et al. (eds.), Lecture Notes in Artificial Intelligence, LNAI, Springer-Verlag. Disponible en fecha Enero de 2007 en: <http://www.cs.vu.nl/~frankh/postscript/EKAW00.pdf>
- Fensel D, van Harmelen F, Klein M, Hans A, Broekstra J, Fluij C, van der Meer J, Hughes J, Krohn U, Davies J, Engels R, Bremdal B, Lau T, Novotny B, Reimer U and Horrocks I. (2002). On-To-Knowledge: Ontology-based Tools for Knowledge Management. Conference on Electronic Business and Electronics Work. October. Madrid, España.
- Fernández López M. (1999). Overview of methodology for building ontologies. Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5) Stockholm, Sweden, August. Disponible en fecha diciembre de 2006 en: <http://www.lsi.upc.es/~bejar/aia/aia-web/4-fernandez.pdf>
- Foundation for Intelligent Physical Agents. (2000). FIPA Ontology Service Specification. Número de documento: XC00086C. Disponible en fecha febrero de 2007 en: <http://www.fipa.org/specs/fipa00086/XC00086C.pdf>
- Gómez-Pérez A, Fernández-López M and Corcho M. (2004). Ontological Engineering. Springer Verlag London.
- Gómez-Pérez Asuncion. (1995). Some ideas and examples to evaluate ontologies. 1043-0989/95 IEEE.
- Gruber T. (1993). Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Available as Technical Report KSL 93-04, Knowledge Systems Laboratory, Stanford University. Disponible en fecha febrero de 2007 en <http://citeseer.ist.psu.edu/gruber93toward.html>
- Gruenwald L, McNutt G and Mercier A. (2003). Using an ontology to improve search in a terrorism Database Systems. Proceeding of he 14<sup>th</sup> International Workshop on database and expert system applications (DEXA'03). 1529-4188/03. IEEE Computer Society

- Grüninger M, Fox M. (1995). Methodology for the design and evaluation of ontologies. IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing. Disponible en fecha Febrero de 2007 en: <http://citeseer.ist.psu.edu/grninger95methodology.html>
- Hewett Michael, Oliver Diane, Rubin Daniel, Easton Katrina, Stuart Joshua, Altman Russ y Klein Teri. (2002). PharmGKB: the Pharmacogenetics Knowledge Base. Nucleic Acids Research, Vol. 30, No 1. Stanford Medical Informatics. SMI Report Number: SMI-2002-0910. USA. Disponible en fecha: Mayo 2006 en: [http://smi-web.stanford.edu/pubs/SMI\\_Reports/SMI-2002-0910.pdf](http://smi-web.stanford.edu/pubs/SMI_Reports/SMI-2002-0910.pdf)
- Hossain Anwar y El Saddik Abdulmotaleb. (2004). Creating Knowledge and Semantic Web Applications on the Web. In Proceedings of the 2nd International Symposium on Innovation in Information & Communication Technology, Amman, Jordan, April 21-22, 2004. Disponible en fecha Febrero de 2007 en: <http://www.site.uottawa.ca/~elsaddik/abedweb/publications/semanticWeb.pdf>
- ISCB (International Society for Computational Biology). (2005). Protégé. Conferencia Internacional Anual N° 13 sobre Sistemas Inteligentes para Biología Molecular. Disponible en fecha diciembre 2006 en: <http://www.iscb.org/ismb2005/demos/31.pdf>
- Jasper Robert and Uschold Mike. (1999). A framework for understanding and classifying ontology applications. Procc of the IJCAI-99 Workshop on ontologies and problem solve methods. Seattle. USA.
- Karp Peter, Chaudhri Vinay y Thomere Jerome. (1999). XOL: An XML-Based Ontology Exchange Language. Versión 0.4. Technical report 559. AI Center, SRI International. Disponible en fecha Enero 2007 en: <http://www.ai.sri.com/pkarp/xol/xol.html>
- Koivunen Marja-Riitta y Millar Eric. (2001). W3C Semantic Web Activity. In Proceedings of the Semantic Web Kick-off Seminar in Finland Nov 2, 2001. Disponible en fecha enero de 2007 en: <http://www.w3.org/2001/12/semweb-fin/w3csw>
- Lozano Tello, Adolfo. (2002). Métrica de idoneidad de ontologías. Tesis Doctoral. Escuela Politécnica de Cáceres. Departamento de Informática. Universidad de Extremadura. España. ISBN: 84-7723-537-6. Disponible en fecha enero 2007 en: <http://quercusseg.unex.es/adolfo/tesis.htm>
- McGuinness Deborah y Van Harmelen Frank. (2004). OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004. Disponible en fecha enero 2007 en: <http://www.w3.org/TR/2004/REC-owl-features-20040210/#s1.3>
- Nirenburg Sergei, Raskin Victor. (2004). Ontological Semantics. MIT Press ISBN 0262140861.
- Noy N. and McGuinness D. (2001). Ontology development 101: A Guide to creating your first ontology. Stanford University. Stanford knowledge Systems Laboratory. Technical Report KSL-01-05.
- Oliver Diane, Rubin Daniel, Stuart Joshua, Hewett Michael, Klein Teri y Altman Russ. (2002). Ontology Development For A Pharmacogenetics Knowledge Base. Stanford Medical Informatics. Stanford University. PMID: 11928517. Disponible en fecha diciembre 2006 en: <http://helix-web.stanford.edu/psb02/oliver.pdf>
- Pastor Xavier, Lozano Raimundo, Jaureguizar Jon, Alonso Albert. (2001). Uso de UMLS para desarrollar un modelo clínico en un sistema de historia clínica electrónica. IV Congreso de Informática de la Salud. Madrid, del 28 al 30 de Marzo de 2001.
- PharmGKB Web Site. (2005). PharmGKB Definitions and Technical Architecture. Disponible en fecha: Mayo 2006 en: <http://www.pharmgkb.org/resources/references.jsp>
- Protégé Web Site: <http://protege.stanford.edu/>

- Schreiber G, Akkermans H, Anjewierden A, deHoog R, Shadbolt N, VandeVelde W, Wielinga B. (1999). Knowledge Engineering and Management: The CommonKADS Methodology. The MIT Press.
- Silva Muñoz, Lydia. (2002). Estudo de ontologías para representação de conteúdos de ensino baseado na www. Instituto de Informática. Universidade Federal do Rio Grande do Sul (UFRGS). Brasil. Disponible en fecha Abril 2006 en: [http://www.inf.ufrgs.br/adapt/publicacoes/TI\\_Lydia.pdf](http://www.inf.ufrgs.br/adapt/publicacoes/TI_Lydia.pdf)
- Staab S, Studer R, Schnurr H and Sure Y. (2001). Knowledge Process and Ontologies. IEEE Intelligent Systems. 16(1):26-34. January-February.
- Su Xiaomeng e Ilebrekke Lars. (2005). Using a Semiotic Framework for a Comparative Study of Ontology Languages and Tools. Information Modeling Methods and Methodologies p 278-299. Norwegian University of Science and Technology, Norway. Disponible en fecha: Mayo 2006 en: <http://www.idi.ntnu.no/~xiaomeng/paper/IDEAchapter.pdf>
- Sure Y, and Studer R. (2003). A Methodology for Ontology-based Knowledge Management. Towards the Semantic Web Ontology-driven Knowledge Management. Edited by Davies John, Fensel Dieter y van Harmelen Frank. Chapter 3. John Wiley & Sons, LTD.
- Sure Y, Staab S, and Studer R. (2003). On-To-Knowledge Methodology. Handbook on Ontologies, chapter 6, pp. 117-132. Springer.
- Swartout William. (1999) Ontologies. IEEE Intelligent Systems. 1094/716/99. January-February, 1999.
- Tramullas, Jesús. (1999). Agentes y ontologías para el tratamiento de información: clasificación y recuperación en Internet. IV Congreso ISKO España. Granada 22-24 Abril. p.247-252 Disponible en fecha: Diciembre de 2006 en: <http://www.tramullas.com/papers/isko99.pdf>
- UMLS® KNOWLEDGE SOURCES. January Release 2005AA Documentation. (2005). Disponible en fecha: Mayo 2005 en: [http://www.nlm.nih.gov/research/umls/umlsdoc\\_2005aa.pdf](http://www.nlm.nih.gov/research/umls/umlsdoc_2005aa.pdf)
- Uschold Mike. (1996). Building Ontologies: Towards a Unified Methodology. Artificial Intelligence Application Institute. University of Edinburgh. Reino Unido. Disponible en: <http://citeseer.nj.nec.com/uschold96building.html>
- Uschold M & Gruninger M. (1996). Ontologies: Principles, Methods and Applications. AIAI-TR-191. Knowledge Engineering Review. Vol 11 Number 2. Disponible en fecha febrero de 2007 en <http://citeseer.ist.psu.edu/uschold96ontologie.html>
- Uschold M, King M. (1995). Towards a Methodology for Building Ontologies. IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing. Disponible en fecha Febrero de 2007 en: <http://citeseer.ist.psu.edu/uschold95toward.html>