

*Universidad de las Ciencias Informáticas*

*Facultad 6*



Esquema de una Ontología para el proceso de Ingeniería  
de Requisitos del Modelo Cubano de Calidad para el  
desarrollo de aplicaciones informáticas en la Industria  
Cubana de Software

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

**Autora:**

**Aleyda Isabel Labrada González**

**Tutora:**

**Ing. Lissette Rodríguez Verdecia**

**Co-tutora:**

**Ing. Glennis Tamayo Morales**

*La Habana, junio 2011*



*“Aprendí que no se puede dar marcha atrás, que la esencia de la vida es ir hacia adelante. La vida, en realidad, es una calle de sentido único”.*

*Agatha Christie*

## **Declaración de Autoría**

Declaro que soy la única autora del trabajo “Esquema de una Ontología para el proceso de Ingeniería de Requisitos del Modelo Cubano de Calidad para el desarrollo de aplicaciones informáticas en la Industria Cubana de Software” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Aleyda Isabel Labrada González

---

Autora

Ing. Lissette Rodríguez Verdecia

---

Tutora

Ing. Glennis Tamayo Morales

---

Co-Tutora

## **Agradecimientos**

*A mis padres Isabel Cristina y Juan Rafael por ser la razón de mi existir, por su dedicación, amor, cariño, comprensión, consejos, por enseñarme siempre el buen camino. No existen palabras para agradecerles, ni para decirles lo mucho que los quiero. Son los mejores padres del mundo. Los quiero con todas las fuerzas de mi corazón.*

*A Yuyito mi hermanito del alma, por ser el mejor hermano que voy a tener en esta vida, por ser mi amigo, mi compañero, mi guía, mi ejemplo, mi protector. Eres el regalo máspreciado que la Vida me ha dado.*

*A mi novio José Miguel (Tü), por su apoyo incondicional y sus consejos en los momentos precisos. Gracias por hacerme una persona cada día mejor, por aconsejarme, ayudarme, por darme tanto amor. Gracias por hacerme la persona más feliz. Yo también te Quiero y te Amo mucho, mucho...*

*A toda mi familia por confiar y esperar siempre lo mejor de mí.*

*A todas mis amistades por estar siempre a mi lado ayudándome y aconsejándome.*

*A todos los profesores, mis tutoras y amigos que de una forma u otra contribuyeron a la realización de este trabajo.*

*A todos muchas gracias...*

*Aleyda.*

**Dedicatoria**

*A mi Mamá, mi Papá, mi Hermano y mi Tü.*

*Por ser las personas que iluminan mi vida, pues por y para ellos vivo.*

*Aleyda*

## Resumen

En los últimos años, el proyecto de la web semántica ha servido para asentar lo que se puede denominar el uso actual del término Ontología. En este contexto, el desarrollo de sistemas que faciliten la gestión del conocimiento es un elemento estratégico para las organizaciones en general. El presente trabajo de diploma propone el diseño de una Ontología para el área de la Ingeniería de Requisitos, que forma parte de las áreas claves identificadas para crear el Modelo Cubano de Calidad para el desarrollo de aplicaciones informáticas. Dicha Ontología soporta los conceptos y relaciones del Proceso Base de Requisitos que es el punto de partida para el diseño de la misma. En el desarrollo del trabajo se exponen los conceptos fundamentales que marcan la evolución de las Ontologías, así como los elementos de su tipología, componentes y herramientas. Esta Ontología brinda respuestas a preguntas relacionadas con el Proceso Base de Requisitos y posibilita la representación de los principales conceptos relacionados con el dominio en cuestión, que permite, almacenar ese conocimiento de forma que pueda ser utilizado por determinadas aplicaciones informáticas y sistemas de información basados en Ontologías.

### **PALABRAS CLAVES:**

Ingeniería de Requisitos, Ontología, Proceso Base de Requisitos.

## Índice de Contenido

INTRODUCCIÓN .....	1
CAPÍTULO 1: FUNDAMENTO TEÓRICO .....	5
1.1. Gestión del Conocimiento .....	5
1.2. Herramientas para la Gestión del Conocimiento.....	6
1.3. Ontologías .....	7
1.3.1. Definiciones de Ontologías .....	7
1.3.2. Objetivos y Características de las Ontologías.....	9
1.3.3. Elementos que componen las Ontologías .....	10
1.3.4. Tipos de Ontologías .....	11
1.4. Herramientas para definir Ontologías .....	13
1.5. Aplicaciones y Beneficios de las Ontologías .....	15
1.6. Metodologías y Métodos para el desarrollo de Ontologías.....	17
1.6.1. Tipos de Métodos para el desarrollo de Ontologías.....	17
1.6.2. Tipos de Metodologías para el desarrollo de Ontologías.....	18
1.7. Proceso de desarrollo de Ontologías .....	19
1.7.1. Pasos para el desarrollo de Ontologías .....	20
1.8. Ontologías Fundacionales .....	22
1.8.1. Ontologías Fundacionales existentes .....	22
1.9. Formas de Representación de las Ontologías.....	24
1.9.1. Mediante un Fichero XML (Lenguaje OWL).....	24
1.9.2. Mediante un Gráfico .....	29
1.10. Ingeniería de Requisitos .....	29
1.10.1. Definición de Ingeniería de Requisitos .....	30
1.10.2. Ontologías en el ámbito de la Ingeniería de Requisitos .....	30
CAPÍTULO 2: DISEÑO DE LA ONTOLOGÍA .....	33
2.1. Reglas fundamentales en el diseño de Ontologías .....	33
2.2. Descripción de los pasos específicos para el diseño de la Ontología.....	33
2.2.1. Definir el dominio y alcance o ámbito de la Ontología .....	33
2.2.2. Considerar reutilizar Ontologías existentes .....	36
2.2.3. Enumerar los términos importantes en la Ontología.....	37
2.2.4. Definir las clases y la jerarquía de clases.....	37
2.2.5. Definir las propiedades de las clases (slots).....	38
2.2.6. Definir las características (facetas) de las propiedades (slots).....	39
CAPÍTULO 3: VALIDACIÓN DE LA ONTOLOGÍA .....	41
3.1. Validación mediante el razonador FACT++ .....	41
3.2. Validación a través del Proceso Base de Requisitos.....	45
CONCLUSIONES GENERALES .....	47

RECOMENDACIONES .....	48
REFERENCIAS BIBLIOGRÁFICAS.....	49
BIBLIOGRAFÍA .....	51
ANEXOS .....	56
GLOSARIO DE TÉRMINOS.....	64

## Índice de Figuras

Figura 1: Clasificación de Ontologías según su uso y reutilización. ....	13
Figura 2: Ontología representada mediante el lenguaje OWL. ....	28
Figura 3: Clases y subclases de una Ontología sobre periféricos de ordenador. ....	29
Figura 4: Diagrama conceptual en una pieza de una Ontología definida en el Proceso DSDM (Proceso de Desarrollo de Software Dirigido por Modelos). ....	31
Figura 5: Algunas clases de la Ontología. ....	38
Figura 6: Parte de la jerarquía de clases de la Ontología. ....	38
Figura 7: Propiedades de la clase Documentos. ....	39
Figura 8: Ejemplo de Facetas del slot “ <i>nombreDocumento</i> ” de la clase “ <i>Documentos</i> ”. ....	40
Figura 9: Ejemplo de Facetas, del slot “ <i>tieneManual</i> ” de la clase “ <i>Herramientas_Trazabilidad</i> ” con su propiedad inversa. ....	40
Figura 10: Uso del razonador FACT++ desde Protégé. ....	42
Figura 11: Verificación de compatibilidad con el razonador FACT++ desde Protégé. ....	43
Figura 12: Solución al error de clases incompatibles. ....	43
Figura 13: Resultados positivos al verificar la compatibilidad con el razonador FACT++ desde Protégé. ....	44
Figura 14: Verificación de la escritura en las clases con el razonador FACT++ desde Protégé (Uso de las comas (,) y de las diéresis (ü). ....	44
Figura 15: Solución de errores desde el código (uso de la coma”,”). ....	44
Figura 16: Solución de errores desde el código (uso de las diéresis en la palabra “Ambigüedades”)... ..	44

## Índice de Tablas

Tabla 1: Intención de las Ontologías en cada proceso de administración del conocimiento.....	16
Tabla 2: Procesos del ciclo de conversión del conocimiento.....	19
Tabla 3: Proceso Base de Requisitos. ....	35
Tabla 4: Resultados del cuestionario de validación.....	45

## Introducción

En la actualidad, las compañías en todo el mundo reconocen que la calidad de un producto se traduce en ahorro de costos y en una mejora general. La industria del desarrollo de software no es la excepción, por lo que en los últimos años se han realizado intensos trabajos para aplicar los conceptos de calidad en este ámbito. Además, la competencia existente entre empresas o compañías desarrolladoras de productos y servicios informáticos, es cada día más fuerte y el principal factor que marca las ventajas de unas sobre otras es la calidad.

El acelerado desarrollo tecnológico de los últimos 50 años, no deja de sorprender y Cuba trata de encontrar su propio camino en el mercado internacional, a partir del desarrollo nacional y del contexto externo en el cual se mueve, con la exportación de sistemas integrales y de diversos servicios. Por ello, se llevó a cabo el desarrollo de la Industria Cubana de Software (ICSW), que está llamada a convertirse en una significativa fuente de ingresos para el país, como resultado del correcto aprovechamiento de las ventajas del capital humano altamente calificado. Las empresas vinculadas a la ICSW, aspiran a desarrollar y entregar software confiable, a tiempo, sin olvidar las perspectivas de los clientes.

Actualmente, la Industria Cubana de Software rige su desarrollo por modelos de calidad, que constituyen estándares internacionales, por lo que el costo de certificación de las entidades pertenecientes a este ámbito es muy elevado.

El creciente y cada vez más acelerado desarrollo de la informática en Cuba, ha impuesto el desarrollo de un Modelo Cubano de Calidad que brinde a la Industria Cubana de Software una guía práctica que permita una mejora continua en sus procesos de desarrollo. El diseño del modelo está destinado a incidir favorablemente en la calidad de los proyectos informáticos, incorporando un conjunto de acciones dirigidas a potenciar la productividad y la gestión del conocimiento en las organizaciones.

La Universidad de las Ciencias Informáticas (UCI), se encuentra entre los principales pilares del auge que experimenta la informática actualmente en el país, conformando una infraestructura innovadora de formación académico-industrial y de servicios, para responder a los retos de las nuevas tecnologías de la información y las comunicaciones, que marcan pautas en el desarrollo integral del siglo XXI.

El Centro de Calidad para Soluciones Informáticas (Calisoft) perteneciente a la Universidad de las Ciencias Informáticas y al Ministerio de la Informática y las Comunicaciones, tiene como misión,

garantizar el crecimiento continuo de una producción de software con calidad en la organización a través de la definición de procesos, siguiendo las especificaciones de metodologías, estándares y modelos de desarrollo de software, por ende, es el principal responsable de la creación del Modelo Cubano de Calidad totalmente nacional e independiente, que cumpla con las perspectivas de la ICSW.

En este sentido han surgido diversos temas de investigación encaminados a tomar las mejores prácticas de los modelos existentes y adaptarlas a las necesidades propias de la Industria Cubana de Software, así como proveer al modelo de un valor agregado en cuanto a la gestión del conocimiento, aspecto novedoso que lo hará diferente de los modelos de calidad existentes.

La gestión del conocimiento en la ingeniería de requisitos aporta aspectos fundamentales como la reutilización de requisitos, incremento y medida de la calidad en requisitos; y mejora de la comunicación entre los miembros del equipo.

Dentro de las áreas claves identificadas y que formará parte del Modelo Cubano de Calidad, se encuentra la Ingeniería de Requisitos.

La gran cantidad de términos y conceptos que existen en la disciplina de Ingeniería de Requisitos y las múltiples interpretaciones que tienen los mismos en un dominio, no se encuentran organizados de manera tal que puedan ser reutilizados, por lo que se hace necesario la representación y organización de estos términos.

A partir de la situación anteriormente expresada, surge el siguiente **problema científico**: ¿Cómo contribuir a la gestión del conocimiento en el Proceso de Ingeniería de Requisitos del Modelo Cubano de Calidad para el desarrollo de aplicaciones informáticas en la Industria Cubana de Software?

Para brindarle una solución a este problema se define como **objeto de estudio**: La gestión del conocimiento en el Proceso de la Ingeniería de Requisitos, de donde se deriva que el **campo de acción** se encuentra enmarcado en la gestión del conocimiento guiada por Ontologías para el Proceso de la Ingeniería de Requisitos del Modelo Cubano de Calidad.

En correspondencia con lo antes expuesto se propone como **objetivo general de la investigación**: definir el esquema de una Ontología para el Proceso de Ingeniería de Requisitos del Modelo Cubano de Calidad para el desarrollo de aplicaciones informáticas en la Industria Cubana de Software.

Para cumplir el objetivo general se trazaron los siguientes **objetivos específicos**:

- Especificar los conceptos y relaciones del Proceso de Ingeniería de Requisitos, a tener en cuenta en la representación de la Ontología.
- Especificar la Ontología mediante la forma seleccionada.
- Validar teóricamente la Ontología.

Para dar cumplimiento de forma exitosa a los objetivos planteados, se definen las siguientes **tareas de investigación**:

1. Selección de las partes útiles y reutilizables de las Ontologías como técnicas de la gestión del conocimiento y su aplicación en el ámbito de la Ingeniería de Requisitos.
2. Caracterización de las Ontologías Fundacionales con vistas a definir cuál de ellas puede ser útil en la representación de la presente investigación.
3. Descripción de las metodologías y métodos para definir Ontologías y seleccionar alguno de ellos para la especificación de la Ontología.
4. Comparación de las formas de representación de las Ontologías y seleccionar una de ellas teniendo en cuenta su futura implementación mediante una herramienta automatizada.
5. Diseño de la Ontología mediante la forma escogida.
6. Selección del razonador semántico a utilizar para validar el diseño de la Ontología.
7. Validación del ajuste entre la Ontología diseñada y el Proceso Base de Requisitos, que sirvió como guía para el desarrollo de la misma.

Como **posible resultado** se propone obtener el diseño de una Ontología mediante un método que especifique los términos y relaciones del Proceso de Ingeniería de Requisitos del Modelo Cubano de Calidad para el desarrollo de aplicaciones informáticas en la Industria Cubana de Software, que garantice la gestión del conocimiento en dicha área.

La investigación está estructurada por: introducción, tres capítulos, conclusiones, recomendaciones, bibliografías, anexos y glosario de términos.

**Capítulo 1: Fundamento Teórico.** Constituye la base teórica de la investigación realizada. Se describen los principales conceptos relacionados con el término de Ontología. Además se realiza un estudio de las diferentes Ontologías desarrolladas en disímiles ámbitos, así como las variadas formas de representación. Se describen las metodologías y métodos existentes que se utilizan para la especificación de la Ontología.

**Capítulo 2: Diseño de la Ontología.** En este capítulo se realiza el diseño de la Ontología mediante la forma seleccionada y siguiendo una guía de pasos específicos que comienzan con la selección de los principales términos del dominio definido y continuando con la definición de clases, propiedades y características hasta obtener el esquema de la Ontología.

**Capítulo 3: Validación de la Ontología.** Se valida teóricamente la Ontología diseñada a través del uso del razonador seleccionado. También se realiza la validación mediante un cuestionario que posee preguntas relacionadas con la taxonomía, el vocabulario, el uso y aplicación de la Ontología.

## Capítulo 1: Fundamento Teórico

### INTRODUCCIÓN

En este capítulo se brinda una visión general de algunos conceptos relacionados con las Ontologías, necesarios para comprender el tema que se aborda en la presente investigación. Como parte fundamental de la misma se describe el estudio del estado del arte realizado a las Ontologías. Se muestran los distintos tipos de Ontologías que existen, métodos y metodologías utilizadas en la creación de las mismas, así como las aplicaciones y beneficios que reporta el aprendizaje de Ontologías. Igualmente se realiza una comparación de las diferentes herramientas, lenguajes y formas de representación existentes en la actualidad, y queda reflejado con la selección de uno de estos aspectos a utilizar para la futura especificación de la Ontología.

### 1.1. Gestión del Conocimiento

El conocimiento está en la mente de las personas, por tanto, no siempre está disponible cuando es necesario para la organización. Para tratar este problema surge la Gestión del Conocimiento (GC) cuya principal misión es crear un ambiente en el que el conocimiento y la información disponible en una organización sean accesibles a todos, permitiendo ser usados para estimular la innovación y mejorar las decisiones. Ese conocimiento que no solo está implícito en los documentos sino en el desarrollo de los procesos que se llevan a cabo en las organizaciones y en la mente de quienes ejecutan esos procesos, necesita ser representado para poder ser socializado entre los integrantes de una organización.

A continuación se expresan varias definiciones realizadas por diferentes autores para referirse a la Gestión del Conocimiento.

Arthur Andersen: *“Necesidad de acelerar el flujo de la información que tiene valor, desde los individuos a la organización y de vuelta a los individuos, de modo que ellos puedan usarla para crear valor para los clientes”*. (1)

Thomas Davenport: *“Proceso sistemático de buscar, organizar, filtrar y presentar la información con el objetivo de mejorar la comprensión de las personas en una especificada área de interés”*. (2)

Eduardo Bueno: *“Es la función que planifica, coordina y controla los flujos de conocimientos que se producen en la empresa en relación con sus actividades y con su entorno con el fin de crear unas competencias esenciales”*. (3)

Tejedor y Aguirre: *“Conjunto de procesos que permiten utilizar el conocimiento como factor clave para añadir y generar valor”*. (4)

A partir de las definiciones anteriores se puede asumir que la Gestión del Conocimiento es la disciplina que promueve la generación, colaboración y utilización del conocimiento para el aprendizaje de las personas en una organización, generándole nuevo valor y elevando el nivel de competitividad en una empresa con el fin de alcanzar sus objetivos con eficiencia y eficacia.

### 1.2. Herramientas para la Gestión del Conocimiento

Existen variadas herramientas para la administración y gestión del conocimiento que tienen como aspecto más significativo para su manipulación, el conocimiento. Entre ellas están:

- Software para la toma de decisiones grupales.
- Bases de datos.
- Las redes neuronales.
- La implementación de Intranets y Extranets para la gestión de información en las organizaciones.
- Sistemas expertos.
- Agentes inteligentes.
- Minería de datos.
- El desarrollo de bibliotecas digitales.
- La aplicación de las más modernas técnicas de extracción de conocimiento a partir de grandes volúmenes de datos y la visualización gráfica de estos.
- Ontologías.

La gestión del conocimiento en las organizaciones se auxilia de herramientas que no siempre permiten representar de una forma dinámica, los conocimientos que una organización genera en sus diferentes contextos. Las Ontologías, son una herramienta eficaz para modelar, compartir y reutilizar conocimiento en las organizaciones y contribuir a la toma de decisiones, haciendo inferencias a partir del conocimiento representado en estos sistemas.

## 1.3. Ontologías

### 1.3.1. Definiciones de Ontologías

El término Ontología proviene del griego ontos (estudio del ser) y logos (palabra). Filosóficamente, *“Ontología es la ciencia de qué es; es una explicación sistemática de la existencia, de los tipos de estructuras, categorías de objetos, propiedades, eventos, procesos y relaciones en cada área de la realidad”*. (5)

Por lo que el Diccionario de la Real Academia lo define *“como parte de la Metafísica que trata del ser en general y de sus propiedades trascendentales”*. (5)

Según Bosch, el término Ontología adquirió otro sentido en el pensamiento lógico de tradición racionalista y cuyo máximo exponente fue Wilhem Leibniz donde pasó a significar *“la representación del mundo de un sistema en general”*. (6)

En el entorno hipertextual<sup>1</sup>, la Ontología ha sido definida como:

*“Una representación **explícita** y **formal** de una **conceptualización compartida**”*. (7)

A continuación se explica el significado de cada uno de los términos que aparecen resaltados en la definición anterior.

Es **explícita** porque se define el conocimiento implícito que existe sobre determinada noción y esa definición es a través del lenguaje natural, lo que la dota de capacidades didácticas que permiten usarla. (5)

**Formal**, porque es legible por computadora, es decir, debe ser desarrollada y puesta en marcha a través de lenguajes computacionales, lo que permite también su reutilización en otros procesos como la gestión de información, específicamente, como sistemas para la organización y de recuperación de información en intranets, bibliotecas digitales y sitios web, entre otros. (5)

Según Tramullas, *“la **conceptualización** corresponde a una parte del mundo o universo que es objeto de tratamiento”* (8), mientras que Lozano, la define como *“una forma de entender y describir un dominio, por lo que constituye un modelo abstracto de algún fenómeno en el mundo. Modelo que se*

---

<sup>1</sup> Hipertextual es lo perteneciente o relativo al hipertexto, y este último es un texto que contiene elementos a partir de los cuales se puede acceder a otra información.

*construye a partir de identificar los conceptos que componen un dominio del conocimiento y las relaciones relevantes establecidas entre dichos conceptos". (9)*

Y **compartida** porque debe ser consensuada y aceptada por un grupo o comunidad científica, y he aquí tal vez la característica de mayor relación con la gestión del conocimiento, pues si un grupo o comunidad debe establecer consenso sobre cómo ven el campo de conocimiento sobre el que trabajan, es muy fácil así, que aflore el conocimiento tácito<sup>2</sup> que de manera individual posee cada miembro a un nivel micro y cada grupo a un nivel macro. (5)

Desde la perspectiva de las aplicaciones que las utilizan, existen tres definiciones diferentes de Ontología, aunque complementarias:

Para Swartout y colegas, *"una Ontología es un conjunto de términos estructurados jerárquicamente que describen un dominio. La Ontología será el "esqueleto" sobre el cual se construye luego, la base de conocimientos". (10)*

Para Bernaras y colegas, *"una Ontología proporciona los significados que describen explícitamente la conceptualización del conocimiento representado en una base de conocimientos". (11)*

Para Mizoguchi, una Ontología es *"un sistema de conceptos/vocabulario usados como primitivas para construir sistemas artificiales". (12)*

Las tres definiciones anteriores tienen en común la concepción de que las Ontologías proporcionan la conceptualización explícita de los términos de un dominio, que sirve como soporte para la implementación de bases de conocimientos preparadas para ser utilizadas por aplicaciones y que resuelven diferentes tareas. Además el término Ontología, se refiere al intento de formular un esquema conceptual en un dominio dado, permitiendo la reutilización y representación del conocimiento.

En la actualidad con el desarrollo de la informática, surgen también las Ontologías en informática, el cual *"hace referencia a la formulación de un exhaustivo y riguroso esquema conceptual dentro de uno o varios dominios dados; con la finalidad de facilitar la comunicación y el intercambio de información entre diferentes sistemas y entidades". (13)*

Estos conceptos están basados en la idea de conceptualización o representaciones del conocimiento que se tiene sobre un dominio como una visión simplificada del mundo. Además es el resultado del

---

<sup>2</sup>Conocimiento tácito: no puede ser formalmente comunicado. Es muy personal y difícil de verbalizar o comunicar ya que se trata de aptitudes físicas o de esquemas mentales, está muy enraizado en la experiencia individual.

proceso por el cual el ser humano forma su propia representación mental acerca de algún proceso, evento o suceso.

### 1.3.2. Objetivos y Características de las Ontologías

Las Ontologías de forma general tienen diferentes objetivos que se pueden resumir en: desarrollar mecanismos y estrategias para importar y reutilizar conceptualizaciones de un dominio a partir de estructuras o esquemas preexistentes; combinar técnicas automáticas y manuales. Representar procesos del tipo no algorítmicos<sup>3</sup>. Poder ser usadas por personas, bases de datos y aplicaciones que necesiten compartir información de un dominio y generar conocimiento a partir de él.

Existen numerosas características de las Ontologías, las cuales son contempladas como:

- Las Ontologías pueden ser consideradas como repositorios de información ligada a hechos particulares, a través de la cual se da una interpretación particular a los datos.
- El conocimiento de una Ontología puede ser verdadero, codificado en una Ontología, o deducido, que significa que se deriva por alguna forma de razonamiento.
- Los términos se referencian unos a otros.
- El diseño de una Ontología implica conocimiento social (el conocimiento debe ser aceptado por las personas que van a trabajar con la Ontología, como mínimo) sobre la temática de la Ontología, así como conocimiento técnico.
- Los conocimientos representados en una Ontología suelen ser estables.

Cada Ontología incorpora un punto de vista y contiene definiciones que proveen del vocabulario para referirse a un dominio y estas dependen del lenguaje que se usa para describirlas. En consecuencia, se pueden señalar algunas de sus características típicas:

- **Ontologías múltiples:** el propósito es hacer explícito algún punto de vista, por lo tanto, a veces será conveniente combinar dos o más Ontologías. Cada Ontología introduce conceptualizaciones específicas.
- **Distintos niveles de abstracción de las Ontologías:** estos niveles de generalización o abstracción dan una topología de Ontologías. La idea es caracterizar una red de Ontologías con el uso de multiplicidad y abstracción. Puesto que no se puede aspirar a tener una descripción

---

<sup>3</sup>Procesos del tipo no algorítmicos son los procesos que no usan algoritmos lógicos para su desarrollo. En la Inteligencia Artificial se utilizan estos procesos.

completa del mundo, se puede pensar en una estrategia de construcción gradual de abajo hacia arriba.

- **Multiplicidad de la representación:** un concepto puede ser representado de muchas formas, por lo que pueden coexistir múltiples representaciones de un mismo concepto.
- **Mapeo de Ontologías:** establecer relaciones entre los elementos de una o más Ontologías, para establecer conexiones, especializaciones y generalizaciones, entre otros.

### 1.3.3. Elementos que componen las Ontologías

Las Ontologías están compuestas por diferentes elementos, como son: conceptos, relaciones, funciones, instancias, propiedades, facetas y axiomas. A continuación se explican cada uno de ellos.

- **Conceptos:** Son identificados además, como clases o subclases. Son cualquier entidad sobre la que se pueda decir algo, a la que se le pueda poner un nombre. Reflejan las ideas, interpretaciones o abstracciones de las características de un grupo de objetos, actividades, fenómenos, descripción de una tarea, función, acción y estrategia. Se formalizarán en una taxonomía de forma jerárquica.
- **Relaciones:** Representan los enlaces entre los conceptos de un dominio. Dentro de las interrelaciones que pueden presentarse entre los conceptos se identifican dos como fundamentales:
  - ❖ **Relación binaria:** Enlaces entre dos términos. Estas relaciones pueden ser de:
    - *Sinonimia:* Hace referencia a las coincidencias de significados entre dos o más palabras.
    - *Antonimia:* Relaciona términos que expresan ideas opuestas o contrarias.
    - *Hiponimia:* Relación de dependencia en la que el significado de la palabra está incluida en otra más general. (*Es\_un, subclase\_de*)
    - *Meronimia:* Define la relación que se establece entre las partes y el todo (*Parte\_de*).
  - ❖ **Relación función-función:** Se refiere a la relación entre varios términos, donde uno de ellos es común para todos.
- **Funciones:** Constituyen un tipo especial de relación, a través de las cuales, se identifica un elemento mediante el cálculo de una función que considera varios elementos de la Ontología. Por ejemplo, pueden aparecer funciones como *categorizar-clase*, *asignar-fecha*, *madre-de*, *precio-de*.

- **Instancias:** Estas describen las propiedades específicas que no aparecen en los conceptos, aclarando su significado en un contexto específico, es decir son utilizadas para representar objetos determinados de un concepto.
- **Propiedad (Atributo, Slot):** Característica que permite describir más detalladamente la clase y sus instancias. Establece que la clase o concepto posee una propiedad que se concretará mediante un valor. Los valores de las propiedades o atributos pueden ser tipos básicos como cadenas de caracteres o números, pero también pueden ser otras clases o instancias.
- **Faceta (Restricción sobre las propiedades):** Es alguna propiedad de la propiedad. Por ejemplo, la cardinalidad<sup>4</sup>, si la propiedad es obligatoria o no, entre otras.
- **Axiomas:** Constituyen declaraciones de teoremas y reglas que deben cumplir las relaciones existentes en la Ontología, que permitirán, la inferencia de conocimiento. Representan aquel conocimiento que no fue definido en ninguno de los componentes anteriores y son sentencias siempre verdaderas para dicho dominio. Por ejemplo: "Las cosas que son iguales a una misma cosa son iguales entre sí." "El lunes va después del domingo".

No siempre todos los elementos deben aparecer en una Ontología. En el caso de que se tenga una Ontología que posee solo conceptos y relaciones se llama taxonomía u Ontología ligera.

### 1.3.4. Tipos de Ontologías

Las Ontologías pueden clasificarse de acuerdo a diferentes criterios, por ejemplo, Van Heistlas (14) clasifica según a la cantidad y tipo de estructura de la conceptualización en:

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

Según los agentes que vayan a emplear la Ontología se clasifican en:

---

<sup>4</sup>Cardinalidad: se define como la cantidad de elementos que contiene un conjunto.

- **Ontologías lingüísticas:** Se vinculan a aspectos lingüísticos tales como los gramaticales, semánticos y sintácticos, destinados a su utilización por los seres humanos.
- **Ontologías no lingüísticas:** Aquellas Ontologías destinadas a ser empleadas por robots o agentes inteligentes.
- **Ontologías mixtas:** Combinan las características de las anteriores.

Según el nivel de abstracción y razonamiento lógico permitido son clasificadas en:

- **Ontologías descriptivas:** Estas Ontologías incluyen descripciones, taxonomías de conceptos, relaciones entre los conceptos y propiedades, pero no permiten inferencias lógicas.
- **Ontologías lógicas:** Estas Ontologías permiten inferencias lógicas mediante la utilización de una serie de componente como la inclusión de axiomas, entre otros.

Guarino (15) clasifica las Ontologías de acuerdo con su dependencia y relación con una tarea específica en:

- **Ontologías Genéricas:** describen conceptos más generales y la categoría de más alto nivel. En ellas se representan conceptos generales y fundacionales del conocimiento como las estructuras parte/todo, la cuantificación, los procesos o los tipos de objetos, independientes de un dominio en particular.
- **Ontologías de Dominio:** estas describen un vocabulario relacionado con un dominio genérico. Son específicas para un tipo de artefacto, generalizaciones sobre tareas específicas en algún dominio. En las que se representa, el conocimiento especializado pertinente de un dominio o subdominio, como en la medicina, las aplicaciones militares y el tráfico.
- **Ontologías de Tareas o Técnicas básicas:** describen características generales de una tarea, actividad o artefacto. Por ejemplo, la evaluación de la contaminación sonora en ambientes urbanos o la descripción de características generales de componentes, procesos o funciones.
- **Ontologías de Aplicación:** estas describen conceptos que dependen tanto de un dominio específico como de una tarea específica y generalmente, son una especialización de ambas. Son usadas por la aplicación en las que se especifican las conceptualizaciones que subyacen a los formalismos de representación del conocimiento. Por ejemplo, las Ontologías de procesos de producción, de diagnóstico de fallas y de diseño intermedio de barcos.

Por último, se puede establecer una clasificación de las Ontologías en función de su uso y reutilización, esto se puede observar en la figura 1.



Figura 1: Clasificación de Ontologías según su uso y reutilización.

Entre las Ontologías de mayor utilidad en la gestión del conocimiento están las Ontologías de dominio, encargadas de modelar el conjunto de conceptos de un dominio y las relaciones que existen entre ellos. Otro tipo importante lo constituyen las Ontologías de aplicación, las cuales han sido usadas con éxito en el análisis de sistemas y para modelar procesos empresariales.

En el presente trabajo se propone, de acuerdo a las clasificaciones anteriores, el uso de Ontologías de dominio, ya que describen los conceptos esenciales, relaciones y teorías de los diferentes dominios de interés. En el caso de la presente investigación estará enfocada al dominio de la Ingeniería de Requisitos, lo que resulta de gran utilidad para representar las relaciones entre los conceptos de esta área del conocimiento, así como modelar el dominio o parte de él.

#### 1.4. Herramientas para definir Ontologías

Entre las herramientas más destacadas utilizadas en la actualidad para la construcción de las Ontologías se encuentran:

- **Ontology Server**<sup>5</sup>: herramienta que permite al usuario la construcción de Ontologías que comparten grupos geográficamente distribuidos. Fue desarrollado en el laboratorio de Sistemas de Conocimiento en la Universidad de Stanford. Este servidor es una extensión de Ontolingua. Al comienzo el término Ontolingua se usaba para referirse tanto al lenguaje para representar

<sup>5</sup>Su servidor se encuentra disponible en : <http://www-ksl-svc.stanford.edu:5915>

Ontologías como a la herramienta utilizada para construirlas. Hoy el término se utiliza para referirse al lenguaje proporcionado por el Ontology Server. Su arquitectura permite el acceso a una librería de Ontologías, traductores de lenguajes y a un editor para crear y navegar por una Ontología.

- **Protégé<sup>6</sup>**: Es un editor de Ontologías escrito en Java, gratuito y de código abierto, fue creado en la universidad de Stanford, donde tras él existe una gran comunidad de desarrolladores y de usuarios universitarios, empresariales y gubernamentales. Posee una interfaz muy intuitiva y en la actualidad está orientado hacia la edición y desarrollo de Ontologías y bases de conocimientos. Posee un editor central sobre el cual se monta una librería de plugins<sup>7</sup>. Tiene las capacidades de importación/exportación a OWL, OIL, RDF, Prolog y FLogic. Además ofrece un repositorio de Ontologías en RDF, DAML+OIL y OWL. Proporciona dos modos diferentes de modelar Ontologías: el editor Protégé-frames y el editor Protégé-OWL. Otras funcionalidades que posee Protégé son los diferentes medios para exportar la Ontología, tales como: XML, RDFS, OWL y bases de datos relacionales, entre otros. Igualmente, dada la arquitectura de implementación de Protégé, existen diversas formas para acceder a esta por medio de lenguajes de programación como Java.
- **OntoEdit<sup>8</sup>**: es una herramienta para la edición de Ontologías, que posee una interfaz abierta permitiendo a los usuarios ajustarla a sus necesidades. Apoya el desarrollo y mantenimiento de las Ontologías utilizando medios gráficos en un entorno web y también puede aplicarse a una Intranet.
- **OilEd<sup>9</sup>**: es un editor de Ontologías gratuito que permite al usuario construir Ontologías utilizando el lenguaje DAM+OIL. Aunque no posee todas las funcionalidades de otros editores de Ontologías, proporciona suficiente funcionalidad para permitir a los usuarios construir Ontologías. Una de las desventajas que presenta este editor es la carencia de recursos para soportar Ontologías grandes, migración e integración de otras Ontologías y diferenciación de versiones.

Para el diseño de la Ontología se utilizará como herramienta el Protégé, ya que se caracteriza por ser un editor libre, de código abierto, de fácil manipulación, arquitectura extensible y buena documentación. Tiene soporte multiplataforma y multiusuario, actualmente tiene usuarios activos en

---

<sup>6</sup> Se encuentra disponible en: <http://protege.stanford.edu>

<sup>7</sup> Un plugin es un módulo de hardware o software que añade una característica o un servicio específico a un sistema más grande.

<sup>8</sup> Se encuentra disponible es la siguiente URL: <http://www.ontoknowledge.org/tools/ontoedit.shtml>

<sup>9</sup> Se encuentra disponible en la siguiente dirección URL: <http://oiled.man.ac.uk>

más de 100 países, empresas y universidades. Además se escogió Protégé porque tiene la capacidad de generar interfaces gráficas, a partir de la Ontología de dominio. En su núcleo, Protégé implementa un conjunto de estructuras de modelado de conocimiento y actividades que ayudan a la creación, visualización y manipulación de Ontologías en varios lenguajes de representación del conocimiento.

### 1.5. Aplicaciones y Beneficios de las Ontologías

En la actualidad existen diferentes campos en los que se realizan proyectos donde se aplican Ontologías, entre ellos están:

- **Comercio electrónico:** Las Ontologías se utilizan en este campo para realizar una descripción del dominio, en este caso un sector comercial, así como una descripción de los usuarios, sus tendencias y relaciones entre ellos. El uso de Ontologías facilita la agrupación en conjuntos de usuarios con comportamientos e intereses similares. La obtención de estos grupos permitirá realizar estudios de mercado de forma más sencilla y personalizar la información a mostrar al usuario según sus intereses.
- **Búsqueda de información en Internet:** Las Ontologías se utilizan también para la anotación de recursos en Internet (documentos, páginas web, imágenes) y para guiar la búsqueda en un dominio concreto. Esto proporcionará una mayor flexibilidad de búsqueda, incrementando la precisión y la recuperación de los documentos buscados.
- **Biomedicina:** Este campo está siendo uno de los de mayor interés por parte de los ingenieros de Ontologías e ingenieros de conocimiento<sup>10</sup>, ya que en él existen numerosos recursos, desde artículos de investigación<sup>11</sup> hasta bases de datos de hospitales, pasando por bases de datos públicas describiendo interacciones entre proteínas y genes<sup>12</sup>. La información necesita ser procesada e interpretada, y las Ontologías pueden ser de gran ayuda para integrar toda esta información y acceder a ella de forma eficiente.
- **Bibliotecas digitales:** Las bibliotecas digitales proporcionan acceso a grandes cantidades de información en forma de documentos digitales, que pueden tener formatos muy variados y estar distribuidos en sistemas informáticos. Las técnicas basadas en las Ontologías permiten tratar esta heterogeneidad mediante la descripción de objetos y repositorios, posibilitando un acceso sencillo a los recursos digitales, consistente y coherente.

---

<sup>10</sup>NCBO: <http://www.bioontology.org/>

<sup>11</sup>PubMed: <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi>

<sup>12</sup>UniProt: <http://www.ebi.uniprot.org/index.shtml>

En la ciencia y en la tecnología, los problemas derivados de la falta de comprensión común entre personas, organizaciones y aplicaciones adquieren una gran importancia. Las Ontologías proporcionan una comprensión común de un determinado dominio y eliminan estas confusiones de conceptos y términos. Se usan también para lograr la interoperabilidad<sup>13</sup> entre sistemas informáticos, razonar automáticamente y para la ingeniería de software.

Las Ontologías son imprescindibles en el desarrollo de la Web Semántica y en los futuros sistemas de gestión empresarial, ya que permitirán que las aplicaciones coincidan en los términos que usan cuando se comunican. Mediante ellas, será mucho más fácil recuperar información relacionada temáticamente, incluso cuando no existan enlaces directos entre las páginas web.

Otras de las aplicaciones de las Ontologías es que son empleadas en el razonamiento automático, es decir, sin intervención humana, usando para ello reglas de inferencia. Un motor de razonamiento puede usar los datos de las Ontologías para inferir conclusiones de ellos.

En el ámbito de la ingeniería del software, las Ontologías ayudan a la especificación de los sistemas facilitando el acuerdo entre desarrolladores y usuarios.

La aplicación de Ontologías en cada proceso de administración de conocimiento puede ser muy amplia. La tabla 1 muestra la intención de las Ontologías en este caso.

**Tabla 1: Intención de las Ontologías en cada proceso de administración del conocimiento.**

<b>Proceso de administración del conocimiento</b>	<b>Participación Ontológica</b>	<b>Procesos ontológicos relacionados</b>	<b>Tipo de Ontología</b>
Generación y adquisición.	Elementos del conocimiento. Categorización y enriquecimiento semántico. Extracción y adquisición del conocimiento.	Creación Ontológica automática y semiautomática. Búsqueda sobre la Ontología.	Ontología de Dominio. Ontología de aplicación.
Representación.	Creación del mapa de	Integración de la	Ontología de Dominio.

---

<sup>13</sup>Dos sistemas son interoperables si pueden trabajar conjuntamente de una forma automática, sin esfuerzo por parte del usuario.

	conocimiento. Recuperación de elementos del conocimiento.	Ontología.	
Almacenamiento.	Persistencia de Metadata <sup>14</sup>	Persistencia de la Ontología.	Ontología de Dominio. Ontología de aplicación.
Transferir.	Expansión de la consulta del usuario.	Búsqueda sobre la Ontología.	Ontología de aplicación.
Integrar.	Integrar el conocimiento. Compartir el conocimiento.	Integración de la Ontología. Creación del mapa de Ontología.	Ontología de Dominio. Ontología de aplicación.
Evaluar.	Monitoreo del conocimiento.	Logs <sup>15</sup> de uso de la Ontología.	Ontología de aplicación.
Mantener.	Mantenimiento del conocimiento.	Evolución de la Ontología.	Ontología de Dominio. Ontología de aplicación.

Los beneficios de utilizar Ontologías se pueden resumir de la siguiente forma:

- Proporcionan una forma de representar y compartir el conocimiento utilizando un vocabulario común.
- Permiten usar un formato de intercambio de conocimiento.
- Proporcionan un protocolo específico de comunicación.
- Permiten una reutilización del conocimiento.

## **1.6. Metodologías y Métodos para el desarrollo de Ontologías**

### **1.6.1. Tipos de Métodos para el desarrollo de Ontologías**

Hay dos métodos<sup>16</sup> principales que permiten diferenciar tipos de Ontologías según su construcción:

<sup>14</sup>Metadata es definida como "Datos acerca de los datos" o "información acerca de la información". También como información sobre objetos web, comprensible por máquinas.

<sup>15</sup>Un log es un registro de actividad de un sistema, que generalmente se guarda en un fichero de texto, al que se le van añadiendo líneas a medida que se realizan acciones sobre el sistema.

<sup>16</sup>En la definición dada por IEEE, los métodos y técnicas son parte de las metodologías ya que son usados para llevar a cabo las tareas dentro de los diferentes procesos en los que consiste una metodología.

- **Kactus:** es un método de construcción de Ontologías que se basa en tomar una base de conocimiento y a partir de esta, determinar y conceptualizar cuáles son los términos y relaciones más importantes que representarán a la Ontología.
- **Sensus:** es un método que representa a las Ontologías construidas a partir de una rama de una Ontología más general y que es especializada para obtener una Ontología nueva. Es decir, consiste en crear Ontologías específicas de dominio a partir de una Ontología más general.

Además existen otros métodos como:

- **Cyc KB:** consiste en la codificación de información extraída a mano y la adición posterior de más información, usando herramientas de soporte y formalización.
- **Uschold y King:** se basa en identificar el propósito para posteriormente construir, evaluar y documentar las Ontologías.
- **CO4 y KA<sup>2</sup>:** permiten la construcción de la Ontología de forma distribuida y colaborativa.

### 1.6.2. Tipos de Metodologías para el desarrollo de Ontologías

- **Metodología de Uschold y King:** propone los siguientes pasos para desarrollar Ontologías: 1) identificar el propósito; 2) capturar los conceptos y relaciones entre estos conceptos y los términos utilizados para referirse a estos conceptos y relaciones; 3) codificar la Ontología. La Ontología debe ser documentada y evaluada, y se pueden usar otras Ontologías para crear la nueva.
- **Metodología Methontology:** permite construir Ontologías totalmente nuevas o reutilizar otras Ontologías. El entorno incluye la identificación del proceso de desarrollo de la Ontología, donde se incluyen las principales actividades (evaluación, conceptualización, configuración, integración e implementación), un ciclo de vida basado en prototipos evolucionados y la metodología propiamente dicha, que especifica los pasos a ejecutar en cada actividad, las técnicas usadas, los productos a obtener y su forma de evaluación. Esta metodología está parcialmente soportada por el entorno de desarrollo ontológico WebODE y propone las siguientes etapas: 1) especificación, 2) conceptualización, 3) formalización, 4) implementación y 5) mantenimiento.
- **Metodología Kactus:** se construye la Ontología sobre una base de conocimiento por medio de un proceso de abstracción. Kactus define los siguientes pasos: 1) especificación de la aplicación, 2) diseño preliminar basado en categorías ontológicas top-level relevantes y 3) refinamiento y estructuración de la Ontología.

- **Metodología Sensus:** es un enfoque top-down para derivar Ontologías específicas del dominio a partir de grandes Ontologías. En esta metodología se identifica un conjunto de términos que son relevantes en un dominio particular. Tales términos se enlazan manualmente a una Ontología de amplia cobertura. Los usuarios seleccionan automáticamente los términos relevantes para describir el dominio y acotar la Ontología Sensus. Consecuentemente, el algoritmo devuelve el conjunto de términos estructurados jerárquicamente para describir un dominio, que puede ser usado como esqueleto para la base de conocimiento.

Finalmente, después de realizar un estudio de las diversas metodologías se puede decir que todavía no existe una metodología estandarizada o aceptada como principal en el diseño y construcción de Ontologías, por lo que todas se pueden emplear; pero se decidió utilizar una guía de pasos específicos que existen para el desarrollo de Ontologías que aparece en el epígrafe 1.7.1, la cual es referenciada en varias bibliografías relacionadas con el diseño y construcción de Ontologías. Estos pasos son más específicos, concretos y entendibles que los pasos que las metodologías proponen. Permiten la reducción de tiempo y rapidez en todo el proceso de diseño y especificación. Resumen las buenas prácticas de las distintas metodologías existentes.

## 1.7. Proceso de desarrollo de Ontologías

El proceso de construcción de una Ontología, permite convertir el conocimiento tácito que poseen los integrantes de la organización o dominio, en conocimiento explícito y representable, permitiendo un intercambio de conocimientos entre los miembros del dominio. A continuación, en la tabla 2 se muestra que el proceso de construcción de Ontologías en las organizaciones, puede contribuir de manera favorable a los procesos del ciclo de conversión del conocimiento tácito a explícito, descrito por Nonaka y Takeuchi. (16)

**Tabla 2: Procesos del ciclo de conversión del conocimiento.**

Ciclo de conversión del conocimiento	Desarrollo y uso de la Ontología
<b>De tácito a tácito:</b> se produce a través de procesos de socialización, es decir, a través de la adquisición de conocimientos e información mediante la interacción directa con el mundo exterior: con otras personas, con otras culturas.	Obtención de inferencias a partir de Ontologías que apoye el proceso de toma de decisiones, mediante la búsqueda de patrones de comportamiento.

<p><b>De tácito a explícito:</b> se produce a través del diálogo.</p>	<p>Para poder representar conocimiento se necesita de la colaboración de los integrantes de un dominio. En la construcción de la Ontología es el paso referido a la obtención del conocimiento que será representado a través del consenso entre los integrantes del dominio.</p>
<p><b>De explícito a explícito:</b> se combinan diferentes formas de conocimiento explícito mediante documentos o bases de datos.</p>	<p>Uso de la Ontología en la representación de conocimiento contenido en bases de datos, intranets, bibliotecas digitales.</p>
<p><b>De explícito a tácito:</b> es la interiorización del conocimiento. Incorporación del conocimiento tácito por parte de los individuos de cualquier organización.</p>	<p>La Ontología puede usarse como modelo para el aprendizaje en un individuo, grupo u organización.</p>

### 1.7.1. Pasos para el desarrollo de Ontologías

A continuación se listan los pasos especificados para la creación de una Ontología:

1. **Definir el dominio y alcance o ámbito de la Ontología:** Es definir el dominio que cubrirá la Ontología, puntualizar el uso, la aplicación final de dicha Ontología; especificar a qué tipo de preguntas responderá la Ontología, recolectar la información necesaria. Una de las formas de determinar el ámbito de la Ontología es preparar una lista con preguntas de competencia. Estas preguntas deberán servir como base de pruebas más adelante.
2. **Considerar reutilizar Ontologías existentes:** Comprobar si es posible usar y extender fuentes de conocimientos ya existentes y en caso de poder aplicarse se recolectan términos y características de Ontologías comunes y que puedan ser de utilidad para el dominio del problema. Ejemplos de Ontologías reutilizables se pueden encontrar en *Ontolingua*<sup>17</sup> o en *DAML ontology library*<sup>18</sup>.

---

<sup>17</sup><http://www.ksl.stanford.edu/software/ontolingua/>

<sup>18</sup><http://www.daml.org/ontologies/>

3. **Enumerar los términos importantes en la Ontología:** Es útil definir una lista con los términos que se quieren almacenar en la Ontología, así como las propiedades que posee cada uno de ellos. El contenido de la lista debe ser preciso y carente de ambigüedades.
4. **Definir las clases y la jerarquía de clases:** De la lista creada en el paso 3, se seleccionan aquellos términos independientes para constituir las clases. A partir de estas se organiza la jerarquía. Para desarrollar una jerarquía de clases existen varios enfoques como son:
  - Un proceso de desarrollo **top-down** comienza con la definición de los conceptos más generales en el dominio y la subsiguiente especialización de los conceptos.
  - Un proceso de desarrollo **bottom-up** comienza con la definición de las clases más específicas, las hojas de la jerarquía, con el subsiguiente agrupamiento de esas clases en conceptos más generales.
  - Un proceso de desarrollo **combinado** es el resultado de una combinación de los enfoques top-down y bottom-up: primero se definen los conceptos más sobresalientes y luego se generalizan y especializan apropiadamente.

Ninguno de estos métodos es mejor que cualquiera de los otros. La decisión a tomar depende de la visión personal de quien modela el dominio.

5. **Definir las propiedades de las clases (slots):** Las clases por sí solas no proveen suficiente información para responder a las peticiones y las preguntas de competencia. Por tanto, se deben describir los conceptos propios de la estructura interna de las clases. Por lo general los términos que no fueron seleccionados en el paso 4, pasan a considerarse propiedades de las clases.
6. **Definir las características (facetas) de las propiedades(slots):** Las propiedades tienen diferentes facetas que describen:
  - **Cardinalidad:** define cuántos valores puede tener una propiedad y cuántos valores puede tener un slot. Algunos sistemas solamente distinguen entre cardinalidad simple (admitiendo a lo sumo un valor) y cardinalidad múltiple (admitiendo cualquier cantidad de valores).
  - **Tipo de valor:** describe el tipo de valor que posee una propiedad. Los tipos de valores más comunes son:
    - ❖ **String:** el valor es una simple cadena de caracteres.
    - ❖ **Number:** (algunas veces los tipos de valores Float e Integer son usados por ser más específicos) describe slots con valores numéricos.
    - ❖ **Boolean:** son simples banderas *sí* o *no*.
    - ❖ **Enumerated:** especifican una lista específica de valores admitidos para el slot.

- ❖ **Instance:** admiten la definición de relaciones entre individuos. Los slots con tipo de valor Instance deben también definir una lista de clases admitidas de las cuales las instancias pueden provenir.
- **Dominio:** conjunto de clases que describe o caracteriza la propiedad.
- **Rango:** clases permitidas para una propiedad de tipo instancia.
- **Valores permitidos:** lista de valores permitidos para una propiedad.
- **Tipo de propiedad:** características de la propiedad. Las más comunes son: *Funcional* (propiedades con un valor único), *Simétrica* (si una propiedad es simétrica, y el par  $(x, y)$  es una instancia de esa propiedad simétrica P, entonces el par  $(y, x)$  es también una instancia de la propiedad simétrica P), *Transitiva* (una propiedad es transitiva, si el par  $(x, y)$  es una instancia de la propiedad transitiva P, y el par  $(y, z)$  es otra instancia de la propiedad transitiva P, entonces el par  $(x, z)$  también es una instancia de P) e *Inversa* (propiedad inversa de otra propiedad, si se estableciera la propiedad P1 como inversa de la propiedad P2, y se relaciona X con Y mediante la propiedad P2, entonces Y estaría relacionado con X mediante la propiedad P1).

### 1.8. Ontologías Fundacionales

Las Ontologías fundacionales (foundational ontology) son conceptualizaciones que contienen especificaciones de conceptos independientes del dominio y relaciones basados en principios formales derivados de la lingüística, la filosofía y la matemática. El rol de estas Ontologías, es servir como punto de partida para la construcción de nuevas Ontologías, suministrar un espacio de referencia para realizar comparaciones fáciles y rigurosas entre diferentes criterios ontológicos y para crear estructuras fundacionales para analizar, armonizar e integrar Ontologías existentes y metadatos estándares.

#### 1.8.1. Ontologías Fundacionales existentes

- **DOLCE** (Ontología Descriptiva para la Ingeniería Lingüística y Cognoscitiva): es el primer módulo de la Biblioteca de Ontologías Fundacionales. DOLCE tiene una clara tendencia cognitiva. Lo interesante de DOLCE es que no adopta una postura prescriptiva y no obliga a una visión unilateral sobre la naturaleza intrínseca del mundo. Las categorías de DOLCE, son artefactos cognitivos, que dependen en última instancia de la percepción humana, las improntas culturales y las convenciones sociales. En otras palabras, DOLCE es una teoría formal de primer orden sobre las cuáles son categorías primitivas de la percepción de la mente humana.

- **SUMO** (Ontología Combinada Superior Sugerida): La Ontología combinada superior o el SUMO es una Ontología superior prevista como Ontología de la fundación para una variedad de sistemas de tratamiento de la información de la computadora. El SUMO se refirió a los conceptos del metanivel<sup>19</sup>, y de tal modo llevaría naturalmente a un esquema de la clasificación para las enciclopedias. Se ha prolongado considerablemente para incluir una Ontología de nivel medio y docenas de Ontologías del dominio. Define una jerarquía de las clases, las reglas y las relaciones relacionadas. El SUMO se organiza para la interoperabilidad de los motores automatizados del razonamiento para maximizar la compatibilidad.
- **BFO** (Ontología Formal Básica): Es una Ontología formal de alto nivel basada en principios probados para la construcción de Ontologías. Se utiliza como base para definir un subconjunto de Ontologías de dominio construido para fines específicos. **OBO** es una especialización de la Ontología formal básica (BFO). OBO define Ontologías para anatomía, genética, bioquímica, fenotipos, secuencias y técnicas de investigación. Además de OWL-DL, OBO utiliza un lenguaje propietario denominado OBO-EDIT. Las Ontologías OBO son de dominio público por lo tanto pueden ser usadas libremente por cualquier persona o institución.
- **LRI-Core** (Ontología de alto nivel): Puede ser considerada un importante instrumento en el desarrollo de Ontologías jurídicas; es el resultado de un decenio de trabajo en diversas ramas del Derecho. Esta Ontología suministra un modelo global y extensible dentro del cual puede establecerse correspondencia entre datos pertenecientes a diversos dominios y permitir la integración de estos.

Se mostraron anteriormente algunas de las Ontologías fundacionales más importantes desarrolladas hasta el momento con vistas a definir cuál de ellas puede ser útil en la representación de la Ontología. Se decidió no utilizarlas y hacer la Ontología desde cero, es decir, no se va a tomar un punto de partida para la construcción de la nueva Ontología que se hará en este trabajo. Esta decisión se basa en los siguientes aspectos:

- Las Ontologías fundacionales consumen un tiempo considerable en entenderlas y familiarizarse con suficiente detalle con sus conceptos y relaciones, tiempo que no se compensa con la utilización de alguna de las Ontologías vistas anteriormente en el dominio que ocupa a este trabajo.

---

<sup>19</sup>Entidades generales que no pertenecen a un dominio específico del problema.

- Son demasiadas abstractas y muy expresivas para el dominio tan específico (la Ingeniería de Requisitos específicamente en el Proceso Base de Requisitos) que se trata en esta investigación.
- La mayoría de las Ontologías fundacionales caracterizadas con anterioridad abarcan grupos de dominios que no se ajustan al de este trabajo.

## 1.9. Formas de Representación de las Ontologías

El conocimiento inferido se puede representar de diferentes maneras, como en ficheros XML+ RDF + OWL, figuras y diagramas, entre otros. Aquí se muestran algunos ejemplos de representaciones ontológicas.

### 1.9.1. Mediante un Fichero XML (Lenguaje OWL)

En el ejemplo tomado de (17) se emplea el concepto “*humanidad*” para expresar la Ontología, también se explica la teoría preliminar del uso de estos lenguajes.

#### a) Espacio de Nombres:

Para poder emplear términos en una Ontología, es necesario indicar de qué vocabulario provienen. De este modo, al igual que sucede en muchos documentos XML, una Ontología empieza con la declaración del espacio de nombres que se encierra en una etiqueta *rdf*.

```
<rdf: RDF
    xmlns = http://domain.tld/path/humanidad#
    xmlns: owl = http://www.w3.org/2002/07/owl#
    xmlns:rdf = http:// www.w3.org/1999/02/22-rdf-syntax-ns#
    xmlns:rdfs = http:// www.w3.org/2000/01/rdf-schema#
    xmlns :xsd = http:// www.w3.org/2001/XMLSchema#
```

En el primer espacio de nombre se indica el actual. En los cuatro siguientes se indican los espacios de nombre relativos a owl, rdf, rdfs y xsd. Esto es necesario para poder emplear etiquetas propias definidas para esos lenguajes.

#### b) Cabecera de la Ontología:

A continuación de la declaración del espacio de nombres se puede indicar la cabecera que describe el contenido de la Ontología actual. Es la etiqueta *owl: Ontology* que permite indicar estas informaciones.

```
<owl:Ontology rdf:about=""
  <rdfs:comment>Ontología que describe la humanidad</rdfs:comment>
  ...
```

**c) Las Clases:**

Las entidades del “mundo real” que se pueden categorizar en grupos o conjuntos de objetos con características similares, forman las clases de la Ontología. Las entidades pueden ser cosas físicas como por ejemplo, los automóviles, o conceptuales como teorías científicas. Estas constituyen el núcleo de una Ontología y describen los conceptos de un dominio concreto.

Un ejemplo de una clase podría ser “Automóvil”, que idealmente representaría a todos los automóviles del mundo. De este modo, cada coche sería una instancia o ejemplar de la clase “Automóvil”. Una clase puede tener además subclases, que representan conceptos más específicos que el de su superclase. La clase “Automóvil” se podría dividir en las subclases “Turismo”, “Todoterreno” y “Deportivo”.

Una clase puede ser definida de cuatro maneras diferentes:

1. **Indicador de clase:** La declaración se hace nombrando directamente la clase. Este es la única de las cuatro maneras que permite definir un nombre para la clase. En los otros casos las clases definidas se conocen como “anónimas”.

```
<owl: Class rdf: ID="Humano"/>
```

2. **Enumeración de los individuos:** Mediante una enumeración de los individuos que pertenecen a una clase se puede definir la clase que los posee. Esto se hace mediante la propiedad *owl: oneOf*.

```
<owl:Class>
  <owl:one of rdf:parse Type="Collection">
    <owl:Thing rdf:about="María"/>
    <owl:Thing rdf:about="Luis"/>
  </owl:oneOf>
</owl:Class>
```

A través del tipo *Collection* definido en la variable *parse Type* de *rdf* se indica que lo que se presenta es un conjunto de elementos, en este caso de “personas”.

3. **Restricción de las propiedades:** En este caso se define una clase anónima cuyas instancias deben satisfacer una determinada propiedad. Este tipo de restricciones pueden ser de valor o de cardinalidad. En el primer caso la restricción se limita a un valor de una propiedad del individuo. En el segundo caso la restricción se enfoca al número de valores que puede tomar una determinada propiedad.

```

<owl:Class rdf:ID="FamiliaNumerosa">
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#tieneNumHijos"/>
      <owl:min cardinality rdf:datatype="&xsd;nonnegative Integer">
        3</owl:min cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>

```

En este caso la clase anónima viene definida entre las etiquetas *subClassOf*. El ejemplo muestra como la clase *FamiliaNumerosa* debe de ser una subclase de una clase anónima que indica que para la propiedad *tieneNumHijos* el valor ha de ser por lo menos 3.

4. **Herencia:** A través de una propiedad de herencia se puede crear una nueva clase. Esto se hace mediante la etiqueta *subClassOf*.

```

<owl:Class rdf:ID="Hombre">
  <rdfs:subClassOf rdf:resource="#Humano"/>
</owl:Class>

```

**d) Las Instancias:**

Los individuos, instancias o ejemplares, consisten en representaciones de objetos o elementos particulares de una clase. Se denominan indistintamente individuos o instancias de la clase. Hay que señalar que es difícil distinguir entre individuos y clases. Ejemplos de instancias son "España", "Documento 141203448-5" y "Ford Mustang".

Un individuo puede hacerse miembro de una clase de diversas maneras: mediante una declaración se indica que un determinado individuo pertenece a una clase y se indica además cuáles son los valores de propiedad para ese individuo.

```
<Humano rdf:ID="María">
  <tienePorMadre rdf:resource="#Rosa"/>
  <tienePorPadre rdf:resource="#Miguel"/>
</Humano>
```

Mediante una declaración anónima. Para ello se procede como en el caso anterior pero obviando el nombre del individuo.

Estableciendo una relación entre el individuo y otro ya existente. Esto se consigue utilizando las propiedades, *owl:sameAs*, *owl:differentFrom* y *owl:allDifferent*.

```
rdf:about="Jose">
  <owl:sameAs rdf:resource="Pepe"/>
</rdf:Description>
```

### e) Las Propiedades:

Las entidades que pertenecen a una clase poseen atributos determinados, por ejemplo, tienen un nombre, un color o un peso. Por tanto, las propiedades consisten en pares de atributo/valor y sirven para describir de forma conveniente las características relevantes de las entidades que forman las clases. Algunos ejemplos son "Población", "ISBN" y "Precio".

Las propiedades permiten expresar hechos sobre las clases y sus instancias.

OWL distingue entre dos tipos de propiedades:

- **Propiedades de objeto** que permiten unir instancias de clase. Este tipo de propiedad es una instancia de la clase *owl: Object Property*.
- **Propiedades de tipo de dato** que permiten enlazar individuos con valores. Este tipo de propiedad es una instancia de la clase *owl: Data Type Property*.

```
<owl:Object Property rdf:ID="tienePorPadre"/>
<owl:Data Type Property rdf:ID=" tieneNumHijos"/>
```

Es posible asignar otro tipo de características a las propiedades, entre las que se encuentran la transitividad, la simetría y otros.

Especificar una propiedad no es más que restringir la relación que simboliza. Así, deben verse las propiedades como una función que hace corresponder a un individuo otro individuo o bien un valor. Por lo tanto, se pueden definir el dominio y la imagen de esta propiedad. Además se puede definir una propiedad como una especialización de una existente. Como se muestra en el siguiente ejemplo:

```
<owl:Object Property rdf:ID="vive">
  <rdfs:domain rdf:resource="#Humano"/>
  <rdfs:range rdf:resource="#País"/>
</owl:Object Property>
```

A continuación, en la Figura 2, se muestra parte de la Ontología anterior representada mediante el lenguaje OWL.

```
<?xml version="1.0" ?>
<rdf: RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
...
<owl:Class rdf:ID="Perifericos">
  <rdf:comment>
Los periféricos de ordenador están conectados a la CPU pero no forman parte de ella.
  </rdf:comment>
</owl:Class>

<owl:Class rdf:ID="Entrada">
  <rdf:comment>
Los periféricos de entrada son una subclase de periféricos de ordenador.
  </rdf:comment>

  <rdfs:subClassOf rdf:resource="#Perifericos" />
</owl:Class>

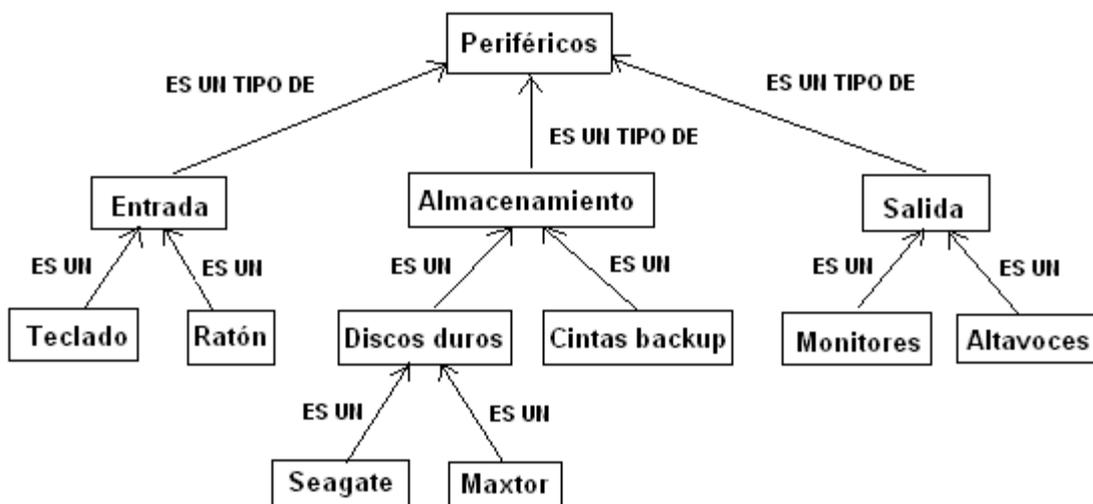
<owl:Class rdf:ID="Teclados">
  <rdf:comment>
Los teclados son una subclase de los periféricos de entrada.
  </rdf:comment>
  <rdfs:subClassOf rdf:resource="#Entrada" />

  <rdfs:subClassOf rdf:resource="#Perifericos" />
</owl:Class>
...
</rdf:RDF>
```

**Figura 2: Ontología representada mediante el lenguaje OWL.**

### 1.9.2. Mediante un Gráfico

Otra forma de representar las Ontologías es mediante un gráfico, como se muestra en la Figura 3, donde se representa un ejemplo de clases y subclases de una Ontología de periféricos de ordenador.



**Figura 3: Clases y subclases de una Ontología sobre periféricos de ordenador.**

En este trabajo se escoge la representación de la Ontología, a través de un gráfico porque realizan la representación mediante modelos y diagramas de manera precisa, entendible y económica. También permiten tomar la información suministrada por los interesados y representarla efectivamente. Sin dejar de mencionar que disminuye las probabilidades de errores y se trata de la forma más compacta y sencilla de presentar los resultados y realizar análisis para la toma de decisiones. Otro de los motivos por lo cual se escogió esta forma de representación es, porque es mucho más fácil comprender una imagen clara, correspondiente a grandes cantidades de datos obtenidos, que todo un párrafo al respecto.

### 1.10. Ingeniería de Requisitos

La ingeniería de requisitos es la primera etapa en la construcción de software y es una de las más cruciales; comprende la definición de requisitos y la elaboración del modelo conceptual del sistema. En esta etapa se captura la información relevante del dominio (conocido como el dominio del problema) y se levantan los requisitos que debe cumplir el software.

### 1.10.1. Definición de Ingeniería de Requisitos

A continuación se expresan varios conceptos sobre la ingeniería de requisitos que proporcionan varios autores según su nivel de experiencia, sentido común o simplemente por su forma de ver los requisitos respecto al desarrollo de un determinado proyecto.

Según la IEEE<sup>20</sup>: *“La Ingeniería de Requisitos es una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo. Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal. Una representación documentada de una condición o capacidad de un sistema”*. (18)

Según Pressman: *“Ingeniería de Requisitos ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software”*. (19)

Según Ortas: *“La Ingeniería de Requisitos se define, como un conjunto de actividades en las cuales, utilizando técnicas y herramientas, se analiza un problema y se concluye con la especificación de una solución (a veces más de una)”*. (20)

Se puede definir la Ingeniería de Requisitos como la ciencia y la disciplina de identificar, documentar, verificar y administrar los requisitos de un sistema. La Ingeniería de Requisitos en sí cumple un papel primordial en el proceso de construcción y producción de un software, es decir, está basado en las necesidades planteadas por los clientes en un nivel muy general, donde se descubren, documentan, analizan y se definen los servicios o componentes de lo que se desea producir, además de las restricciones que tendrá el producto o software. Su principal tarea consiste en la definición del proceso a seguir en la construcción de un software, y facilitar la comprensión de lo que el cliente requiera.

### 1.10.2. Ontologías en el ámbito de la Ingeniería de Requisitos

#### **Ontología para el Proceso de DSDM (Proceso de Desarrollo de Software Dirigido por Modelos) considerando Requisitos No-Funcionales.**

Esta Ontología consta de cuatro grandes grupos: requisitos, estilos, artefactos y modelos. Para todos ellos, se definen los términos básicos y se muestran sus relaciones mediante diagramas de clase UML.

---

<sup>20</sup>Instituto de Ingenieros Eléctricos y Electrónicos (El Institute of Electrical and Electronics Engineers (IEEE), por sus siglas en inglés)

Se establece la diferencia entre requisitos funcionales y requisitos no funcionales (RNF) y se clasifican los RNF en RNF arquitectónicos y RNF tecnológicos. (Ver figura 4)

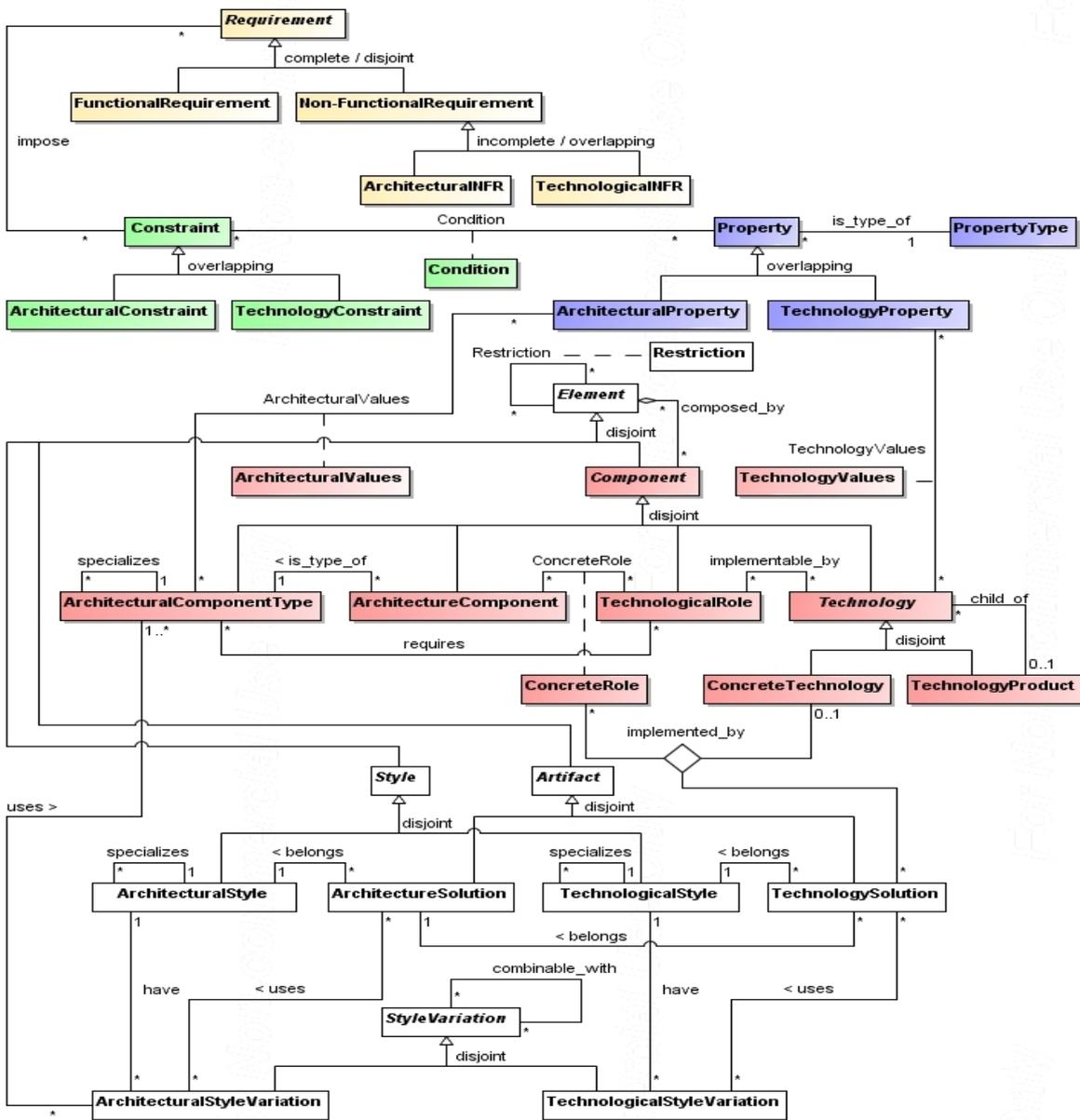


Figura 4: Diagrama conceptual en una pieza de una Ontología definida en el Proceso DSDM (Proceso de Desarrollo de Software Dirigido por Modelos).

Existen otras Ontologías definidas en este dominio; pero para áreas como Calidad, la Ingeniería de Software y otras, que para esta investigación no son de gran importancia. Se muestra como ejemplo a la Ontología anterior porque es la que más se asemeja al dominio de la Ingeniería de Requisitos,

además es totalmente nuevo el dominio que se desea abarcar en la Ontología de este trabajo, por ende es difícil de encontrar Ontología relacionadas con este.

### **CONCLUSIONES DEL CAPÍTULO**

En este capítulo se expusieron los principales conceptos relacionados con esta investigación, así como la posible solución a los problemas planteados. Fueron estudiados los aspectos relacionados con las Ontologías y la gestión del conocimiento llegando a la conclusión de que las Ontologías son una herramienta eficaz para modelar, compartir y reutilizar conocimiento en las organizaciones y contribuir a la toma de decisiones.

De todos los tipos de Ontologías que se estudiaron, se optó por la Ontología de Dominio por sus características que son perfectamente aplicables al dominio donde se desea crear la Ontología que en este caso es el dominio de Ingeniería de Requisitos.

Se mostraron las principales herramientas y formas de representación utilizados en el diseño de las Ontologías, y se ha decidido utilizar la herramienta Protégé y la forma gráfica para representar todos los términos, conceptos, propiedades y atributos del esquema de la Ontología.

Existen diversas metodologías para guiar el proceso de desarrollo de Ontologías, pero no existe una metodología estándar creada por especialistas en el desarrollo de Ontologías, por lo que se decide seguir una serie de pasos que son generales y han de tomarse en consideración a la hora de desarrollar una Ontología.

## Capítulo 2: Diseño de la Ontología

### INTRODUCCIÓN

En este capítulo se detalla todo el proceso de diseño y desarrollo del esquema de la Ontología propuesta para el dominio de Ingeniería de Requisitos, específicamente para el Proceso Base de Requisitos propuesto para el Modelo Cubano de Calidad para el desarrollo de aplicaciones informáticas. Se hace un estudio profundo de todos los términos relacionados con el dominio donde se especifican los conceptos, relaciones y atributos de los mismos. Además se describen los pasos y reglas a seguir para el diseño de la Ontología a través de la herramienta Protégé.

### 2.1. Reglas fundamentales en el diseño de Ontologías

Resulta conveniente destacar las reglas fundamentales en el diseño de Ontologías. Estas reglas pueden ayudar, en el momento de tomar decisiones relativas al diseño de la Ontología:

1. No hay una forma correcta de modelar un dominio, al contrario, siempre hay alternativas viables. La mejor solución casi siempre depende de la aplicación que se tiene en mente y las extensiones que se anticipa.
2. El desarrollo de Ontologías es un proceso necesariamente iterativo.
3. Los conceptos en la Ontología deben ser cercanos a los objetos (físicos o lógicos) y relaciones en el dominio de interés. Esos son muy probablemente sustantivos (objetos) o verbos (relaciones) en oraciones que describe el dominio.

Es necesario definir para qué va a ser utilizada la Ontología y cuán detallada o general será, tener en cuenta las alternativas viables y determinar cuál es la más acertada, sin dejar de pensar en que la Ontología no es más que un modelo de la realidad del mundo y los conceptos definidos deben reflejar esta realidad tanto como sea posible. Después de que se defina una versión inicial de la Ontología, esta puede ser evaluada para corroborar su consistencia. Este proceso de diseño iterativo probablemente continuará a través del ciclo de vida entero de la Ontología.

### 2.2. Descripción de los pasos específicos para el diseño de la Ontología

Los pasos que se siguieron para el desarrollo y construcción de la Ontología (ver apartado 1.7.1) fueron los siguientes:

#### 2.2.1. Definir el dominio y alcance o ámbito de la Ontología

Para comenzar se delimitó un dominio y alcance que no es más que dar respuesta a las siguientes preguntas:

¿Cuál es el dominio que la Ontología cubrirá?

¿Para qué usar la Ontología?

¿Para qué tipos de preguntas la información en la Ontología debería proveer respuestas?

¿Quién usará y mantendrá la Ontología?

Se ha definido como dominio para la Ontología propuesta, el dominio de la Ingeniería de Requisitos específicamente en el Proceso Base de Requisitos (ver tabla 3). Será usada como herramienta para la gestión del conocimiento en sistemas de información basados en Ontologías. Brindará respuestas a preguntas relacionadas con el Proceso Base de Requisitos que estén vinculados a determinadas aplicaciones. También proveerá cuáles serán los roles de esos procesos, las actividades y los diferentes documentos que han sido utilizados para su realización. Tal es el caso del Proceso Base de Requisitos, que gestiona las necesidades de los clientes y estos están relacionados con determinados roles y actividades. La usarán los analistas de software en la organización donde se utilice la Ontología y será mantenida en principio, por las mismas personas que la usen.

Tabla 3: Proceso Base de Requisitos.

Proceso Base de Requisitos				
Criterios de Entrada		Necesidades del cliente		
Criterios de Salida		Producto desarrollado		
Roles	Entrada	Guía/Apoyo	Actividades	Salida
			Inicio	
Cliente Analista	- Catálogo de personal de la organización cliente	- Criterios para caracterizar los clientes del proyecto	1. Caracterización y selección de los clientes.	- Lista de clientes adecuados para proponer los requisitos
Cliente Analistas		- Técnicas de obtención de requisitos (Cuestionarios, preguntas de las entrevistas, fichas de captura de requisitos)	2. Obtención de los requisitos del cliente.	- Lista de requisitos del cliente (resultado de: entrevistas, cuestionarios, tormentas de ideas, talleres, etc). - restricciones
J'Proyecto Analistas	- Lista de requisitos del cliente	- Criterios para chequear requisitos del cliente	3. Entender, analizar y refinar los requisitos	- Priorización de los requisitos. - Riesgos de los requisitos - Lista de requisitos entendidos (chequeados) - Lista de requisitos en conflicto
Analistas	- Lista de requisitos del cliente - restricciones	- Plantilla de Especificación de Requisitos de Software	4. Especificar los requisitos del cliente	- Especificación de Requisitos de Software (ERS)
Analistas	- Especificación de Requisitos de Software (ERS)	- Herramienta de Gestión de Proyecto	5. Comprometerse a los requisitos del cliente	- tickets (aceptados)
Cliente J'Proyecto Analistas	- Lista de requisitos en conflicto - ERS		6. Revisión, validación y aceptación de requisitos de cliente	- ERS (firmada por las partes)
Analistas	- Lista de clientes - ERS	- Herramienta de trazabilidad - Manual de la herramienta	7. Crear sistema de trazabilidad con requisitos y clientes	- Requisitos y Clientes insertados en la Herramienta de trazabilidad (relaciones establecidas)
J'Proyecto Arquitecto de Software Analistas Administrador de la calidad	- Solicitud de Cambios	- Herramienta de trazabilidad		- Cambio realizado ó cambio rechazado o postpuesto
Analistas J'Proyecto Arquitecto de Software	- ERS		8. Especificación de los requisitos técnicos (o funcionales)	- Requisitos técnicos distribuidos en subsistemas o módulos - Arquitectura Inicial - Diccionario de datos - Diagramas de estado - Modelaje de requisitos
J'Proyecto Analistas	- Requisitos técnicos	- Criterios para chequear requisitos técnicos	9. Analizar requisitos técnicos	- Minuta de reunión - Requisitos técnicos chequeados contra criterios.
J'Proyecto Miembros del proyecto	- Requisitos técnicos		10. Realizar y registrar compromiso de miembros del proyecto a requisitos técnicos	- tickets (aceptados)
Analistas	- Requisitos técnicos	- Herramienta de trazabilidad - Manual de la herramienta	11. Actualizar sistema de trazabilidad con requisitos técnicos	- Requisitos técnicos relacionados con requisitos del cliente y otros productos (herramienta de trazabilidad actualizada)
Analista/ Administrador de la calidad		- Guía de trazabilidad - Sistema de trazabilidad - Lista de chequeo	12. Revisar inconsistencias, omisiones y ambigüedades	- Inconsistencias - Acciones Correctivas
			Fin	

Para determinar el alcance de una Ontología sería a través de la elaboración de una serie de preguntas de competencias que la base de conocimientos creada debe ser capaz de responder. Estas preguntas servirán después, de control e inicialmente no tienen por qué ser tan exhaustivas. En el dominio que se intenta modelar, las posibles preguntas de competencia, serían las siguientes:

- ¿Quiénes son los responsables de la obtención de los requisitos del cliente?
- ¿Con qué actividad está relacionada la actividad de validar los requisitos?
- ¿A qué tipo de rol pertenece la solicitud de cambios?
- ¿Qué herramienta de trazabilidad se puede utilizar para crear sistema de trazabilidad con requisitos y clientes?
- ¿Cuáles son los diagramas que pueden ser modelados a partir de la especificación de los requisitos técnicos?
- ¿Cuáles son los riesgos de los requisitos que se obtienen al entrarle los requisitos del cliente y técnicos?
- ¿Quiénes son los encargados de asegurar la consistencia entre los requisitos del cliente con otros productos o artefactos del proyecto?
- ¿Quiénes llevan el seguimiento de los requisitos?
- ¿Qué tipo de actividad se ejecuta con la evaluación del impacto de los cambios?
- ¿Quiénes son los responsables del registro de peticiones de cambios?

Semejante a estas preguntas, se pueden elaborar otras y así la Ontología incluirá la información de varias actividades realizadas por los diferentes roles y tipos de requisitos, documentos de entradas, listas que se generan en las salidas, herramientas de trazabilidad y otros.

### 2.2.2. Considerar reutilizar Ontologías existentes

Es recomendable considerar la reutilización de una o algunas Ontologías existentes, refinando o tomando de ellas la definición de conceptos y la estructura de relaciones jerárquicas que han establecido. En el caso específico de esta investigación, se hicieron búsquedas de Ontologías no solo sobre el dominio de Ingeniería de Software, sino además, sobre la Ingeniería de Requisitos. Para ello se revisaron las bibliotecas de Ontologías reusables en la web, tales como: Ontolingua<sup>21</sup>, la biblioteca de Ontologías DAML<sup>22</sup>, Ontologías comerciales como UNSPSC<sup>23</sup>, RosettaNet<sup>24</sup> y DMOZ<sup>25</sup>. A partir de dicha revisión se obtuvo como resultado que existen Ontologías relacionadas con la ingeniería de software, pero no existe una Ontología enfocada a representar los principales conceptos y relaciones

---

<sup>21</sup><http://www.ksl.stanford.edu/software/ontolingua>

<sup>22</sup><http://www.daml.org/ontologies>

<sup>23</sup><http://www.unspsc.org>

<sup>24</sup><http://www.rosettanet.org>

<sup>25</sup><http://www.dmoz.org>

en el dominio de la Ingeniería de Requisitos, donde se defina un proceso completo de este tipo, elemento fundamental de la presente investigación.

### 2.2.3. Enumerar los términos importantes en la Ontología

Al analizar el dominio, se identificaron una serie de términos relevantes que en pasos posteriores ayudarán a definir clases, propiedades e instancias de la Ontología. Para ello, se definió una lista con los términos más importantes que tratará la Ontología, la mayoría de esos términos fueron extraídos de la tabla 3. Después se organizaron en dos grupos según el rol que podrían jugar en la Ontología. En el primer grupo se ubicaron aquellos términos que describían objetos con existencia independiente (posibles clases), y en el segundo grupo aquellos términos que describían cómo eran esos objetos (posibles propiedades de clase). En el primer grupo se identificaron 114 términos y en el segundo 222 términos. Ver anexos 2 y 3.

Es importante obtener una lista integral de términos sin preocuparse del recubrimiento entre los conceptos que representan, relaciones entre los términos, o cualquier propiedad que los conceptos puedan tener, o si los conceptos son clases o slots.

### 2.2.4. Definir las clases y la jerarquía de clases

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

A partir de la lista de términos definida en el paso anterior y de su organización, se seleccionaron los términos candidatos a ser clases; de esos términos seleccionados se analizaron las relaciones existentes entre ellos y finalmente se determinaron las clases que formarán parte de la Ontología. Luego se organizaron las clases jerárquicamente. En el Anexo 2 se listan las clases definidas para la Ontología propuesta.

Las figuras 5 y 6, muestran algunas clases y jerarquía de clases respectivamente, de la Ontología diseñada en el Protégé con la versión 4.0.2. Esta ha sido diseñada basándose en el criterio **top-down** ya que se comenzó con la definición de los conceptos más generales del dominio, en este caso el Proceso de Ingeniería de Requisitos y luego los más específicos como las Actividades, Documentos, Herramientas.



Figura 5: Algunas clases de la Ontología.



Figura 6: Parte de la jerarquía de clases de la Ontología.

Para una vista más completa ver Anexo 1.

### 2.2.5. Definir las propiedades de las clases (slots)

A partir de la lista de términos definida y de su organización, se seleccionaron los términos candidatos a ser propiedades; de esos términos seleccionados se analizaron las relaciones existentes entre ellos y las clases definidas, para finalmente determinar las propiedades que forman parte de la Ontología, para asociar dichas propiedades a la clase correspondiente. Ver anexo 3.

En el Protégé, se crearon las propiedades que forman parte de la Ontología. Dentro de las propiedades existen propiedades cuyo valor corresponde a una o varias instancias de una clase (Object Properties) y otras propiedades cuyo valor corresponde a un tipo de dato (Data Properties).

La clase **Herramientas de Trazabilidad** tiene una propiedad llamada *tiene\_guía\_O\_Apoyo, es\_Manual\_de, insertados\_en, tiene\_Control, tiene\_Manual* y *actualizada*; de igual forma la clase **Documentos** contiene las propiedades *controlDocumentos, tiene\_Control\_Cambios, tiene\_Reglas\_Confidencialidad, tiene\_Ubicación, nombre\_Documento* y *nombre\_Proyecto*. Se ilustra este último ejemplo en la figura 7, por solo mencionar algunos ejemplos de clases que poseen

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

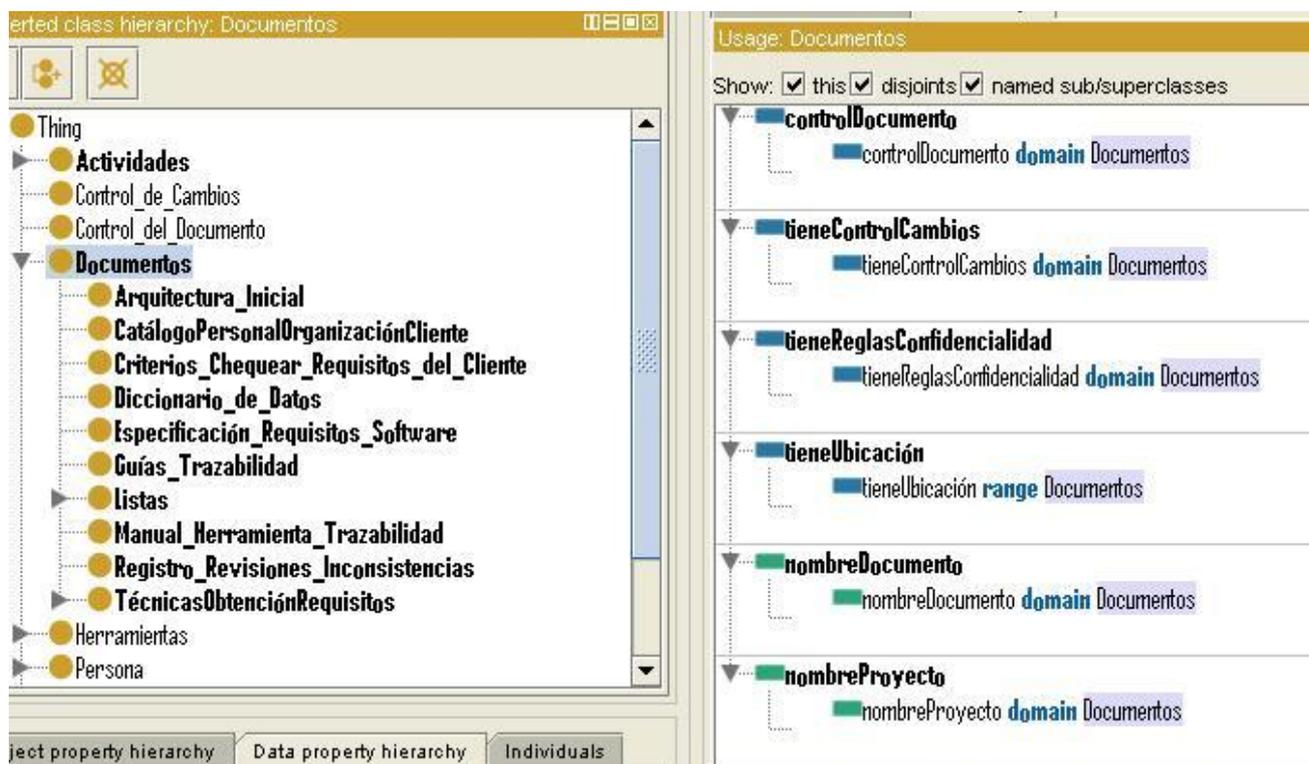


Figura 7: Propiedades de la clase Documentos.

### 2.2.6. Definir las características (facetas) de las propiedades (slots)

A partir de las propiedades de las clases se determinaron las facetas de dichas propiedades como fueron: el tipo de valor, valores admitidos, el número de los valores (cardinalidad), el dominio, el rango, tipo de propiedad y otras características de los valores que los slots pueden tomar.

Por ejemplo, el valor del slot **nombreDocumento** de la clase **Documentos** es una cadena de caracteres, es decir, **nombreDocumento** es un slot con *String* como tipo de valor, ver figura 8.

En la figura 9 se muestra el slot **tieneManual** (como en “**Herramientas\_Trazabilidad tieneManual** de la clase **Manual\_Herramienta\_Trazabilidad**”), donde el dominio de esta propiedad es **Herramientas\_Trazabilidad** y el rango es **Manual\_Herramienta\_Trazabilidad**, además la propiedad **tieneManual** tiene como propiedad inversa **esManualDe**, es decir “**Manual\_Herramienta\_Trazabilidad esManualDe Herramientas\_Trazabilidad**” donde el dominio es **Manual\_Herramienta\_Trazabilidad** y el rango es **Herramientas\_Trazabilidad**.



Figura 8: Ejemplo de Facetas del slot “nombreDocumento” de la clase “Documentos”.



Figura 9: Ejemplo de Facetas, del slot “tieneManual” de la clase “Herramientas\_Trazabilidad” con su propiedad inversa.

Finalmente en el Anexo 1 se muestra el Esquema de la Ontología desarrollada con el Protégé.

## CONCLUSIONES DEL CAPÍTULO

En este capítulo se presentó el diseño de la Ontología siguiendo los pasos descritos anteriormente, quedando como resultado, lo siguiente:

- La Ontología está definida en el dominio de la Ingeniería de Requisitos específicamente en el Proceso Base de Requisitos. Será usada como herramienta para la gestión del conocimiento en sistemas de información basados en Ontologías.
- Las clases y jerarquía de clases se definieron a través del proceso de desarrollo top-down, ya que se comenzó con la definición de los conceptos más generales y se llegó a los más específicos.
- Durante el diseño de la Ontología se especificaron 114 clases y 222 propiedades de las cuales 108 fueron propiedades de tipo de objeto y 114 de tipo de datos.

## Capítulo 3: Validación de la Ontología

### INTRODUCCIÓN

En este capítulo, se valida el esquema de la Ontología diseñada mediante dos vías: teniendo en cuenta la jerarquía de sus conceptos según las descripciones que tengan y comprobar la consistencia de la Ontología, haciendo uso de uno de los razonadores existentes; así como al cumplimiento de las necesidades por las cuales se decidió realizar el diseño de dicha Ontología; esta validación se llevó a cabo mediante un cuestionario.

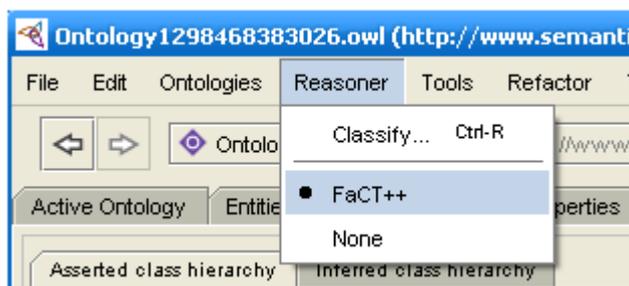
### 3.1. Validación mediante el razonador FACT++

Protégé incluye dos tipos de razonadores (clasificadores o razonadores): FACT y Pellet.

- **FaCT** (Fast Classification of Terminologies): desarrollado por Ian Horrocks en la Universidad de Manchester, es de código abierto, puede ser usado para chequear el grado de satisfacción de las descripciones. Permite reglas transitivas, inversas, restricciones cualificadas y jerarquías, entre otros. Es lo suficientemente expresivo para soportar y razonar sobre cualquier base de conocimiento. Las características de FaCT son muy similares a las de RACER. FaCT implementa dos razonadores, uno para la lógica con roles transitivos, roles funcionales y jerarquía de roles; y otro para la lógica con roles inversos y restricciones de cardinalidad numéricas. Actualmente es el razonador empleado por defecto en el editor de Ontologías OilEd y Protégé para clasificar los conceptos en una jerarquía según las descripciones que tengan y comprobar la consistencia de la Ontología. En estos momentos existe una nueva versión de FaCT denominada FaCT++ que optimiza los algoritmos utilizados en el FaCT original.
- **PELLET**: es un razonador para OWL de código abierto basado en Java, implementado y distribuido por el grupo de investigación Mindswap de la Universidad de Maryland. Mediante su uso es posible validar, comprobar la consistencia de Ontologías, clasificar la taxonomía. Algunas de las funcionalidades principales de este razonador son las de:
  - ❖ **Comprobación de la consistencia**: Comprueba que no existen contradicciones en la Ontología. La semántica de OWL define una especificación formal para la definición de la consistencia en una Ontología mediante el uso Pellet.

- ❖ **Corrección de los conceptos:** Verifica si es posible que se definan instancias para una clase; si no es así, definir una instancia para una clase provocaría que toda la Ontología fuera inconsistente.
- ❖ **Clasificación:** Observa la relación entre cada clase y crea la jerarquía de clases completa.
- ❖ **Realización:** Encuentra las clases más específicas a las que pertenece una instancia; en otras palabras, determina la clase a la que pertenece cada uno de los individuos.

De los razonadores antes mencionados se propone el uso de FACT++ para la validación de la Ontología debido a que posee licencia libre, todavía tiene mantenimiento, sigue en desarrollo (actualmente se está investigando cómo aumentar la capacidad de razonamiento del motor de inferencia) y tiene soporte de OWL. También trabaja de forma eficiente con Ontologías de tamaño grande y mediano. Además, Protégé en la versión que se utilizó para el diseño de la Ontología, trae por defecto al razonador Fact++. Ver figura 10.



**Figura 10: Uso del razonador FACT++ desde Protégé.**

La Ontología puede ser enviada al razonador para deducir automáticamente la clasificación de jerarquía y también para comprobar la consistencia lógica de la Ontología. En Protégé, la jerarquía de clases que se diseña es nombrada asserted hierarchy (jerarquía afirmada) y la jerarquía de clases que se calcula automáticamente por el razonador es nombrada inferred hierarchy (jerarquía inferida).

Cuando se elige al razonador, la ventana de la jerarquía inferida se abre en la parte superior de la ventana de jerarquía existente. Si una clase se ha clasificado y validado correctamente entonces el nombre de la clase aparecerá con el color negro en la jerarquía inferida. Si una clase ha sido declarada incompatible<sup>26</sup>, incoherente<sup>27</sup> o inconsistente<sup>28</sup> su icono o su nombre completo se resaltará en rojo.

<sup>26</sup> Impedimento o tacha legal para ejercer una función determinada, o para ejercer dos o más cargos a la vez. Según la definición RAE.

<sup>27</sup> No tiene conexión, ni relación o unión de unas cosas con otras. Según la definición RAE.

<sup>28</sup> Falta de coherencia entre las partículas de una masa o los elementos de un conjunto. Según la definición RAE.

En esta Ontología tras aplicarle el razonador Fact++ ocurrieron varios errores como:

### Clases Incompatibles:

Este ejemplo se puede observar en la figura 11, donde en la jerarquía de clases, *Actividades\_Intermedias* debe aparecer como una subclase de *Actividades\_Fin* y como una subclase de *Actividades\_Inicio*. Esto significa que *Actividades\_Intermedias* es una *Actividades\_Fin* y una *Actividades\_Inicio* y que todos los individuos que son miembros de la clase *Actividades\_Intermedias* también son miembros de la clase *Actividades\_Fin* y miembros de la clase *Actividades\_Inicio*. Intuitivamente esto es incorrecto ya en este caso, una actividad no puede estar al mismo tiempo en el inicio y fin.



Figura 11: Verificación de compatibilidad con el razonador FACT++ desde Protégé.

En la figura anterior se puede apreciar como el razonador resalta en color rojo las clases que ha detectado como incompatibles.

Sin embargo, hay que recordar que los nombres de las clases han sido puestos por los diseñadores de la Ontología y para el razonador las denominaciones no tienen significado; por ende el razonador no puede determinar que las clases son incompatibles basándose en sus nombres.

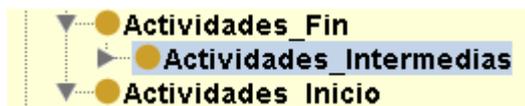
La razón real por la que se ha detectado la clase *Actividades\_Intermedias* incompatible es debido a que sus superclases *Actividades\_Fin* y *Actividades\_Inicio* son disjuntos entre sí, es decir, individuos que son miembros de la clase *Actividades\_Fin* no pueden ser miembros de la clase *Actividades\_Inicio* y viceversa.

La solución a este error es quitar la declaración de clases disjuntas entre la clase *Actividades\_Inicio* y la clase *Actividades\_Fin*, como se muestra en la figura 12.



Figura 12: Solución al error de clases incompatibles.

Ahora debe notarse que *Actividades\_Intermedias* ya no es incompatible, lo que significa que los individuos que son miembros de la clase *Actividades\_Intermedias* también son miembros de la clase *Actividades\_Inicio* y *Actividades\_Fin*. Ver figura 13.

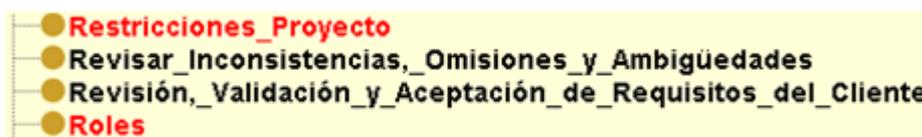


**Figura 13: Resultados positivos al verificar la compatibilidad con el razonador FACT++ desde Protégé.**

### Errores de Escritura:

Este error es el más común, debido a que al nombrar una clase no se debe incluir caracteres especiales como arrobas (@), comas (,), punto y coma (;) espacios en blancos, paréntesis (), diéresis (ü) y guiones (-) por solo mencionar algunos ejemplos.

Al invocar al razonador estos errores son resaltados en color negro. Ejemplos de ellos se pueden apreciar en la figura 14, donde se evidencia el uso de la coma (,) seguido de las palabras “*Inconsistencias*” y “*Revisión*” así como las diéresis (ü) en la palabra “*Ambigüedades*”.



**Figura 14: Verificación de la escritura en las clases con el razonador FACT++ desde Protégé (Uso de las comas (,) y de las diéresis (ü)).**

Para solucionar estos errores de escritura, se renombra la clase en el diseño o se corrige directamente desde el código generado por el diseño de la Ontología, como se puede apreciar en las figuras 15 y 16.

```
>u    uuu    <!-- http://www.semanticweb.org/ontologies/2011/
#Revisión_Validación_y_Aceptación_de_Requisitos_del_Cliente"/
maDecisi&#243;n;nyComunicarlaAInteresados -->00    <owl:Datatyp
```

**Figura 15: Solución de errores desde el código (uso de la coma”,”).**

```
cos"/>0    <rdfs:range rdt:resource="&xsd:boolean
"#Revisar_Inconsistencias_Omisiones_y_Ambigüedades"/>
"#seEspecificanRequisitosDelCliente">0    <rdfs:d
```

**Figura 16: Solución de errores desde el código (uso de las diéresis en la palabra “*Ambigüedades*”).**

En el Anexo 4 se muestra el resultado positivo al solucionar todos los errores, inconsistencias, e incoherencia emitidas por el razonador quedando sin errores la Ontología.

### 3.2. Validación a través del Proceso Base de Requisitos

Este tipo de validación se empleó para verificar que justamente el diseño de la Ontología que se realizó, cumple con lo definido en el Proceso Base de Requisitos a partir del cual se definieron los conceptos de dicho dominio, así como las clases y propiedades.

Al culminar el esquema de la Ontología se le circuló a la Msc. Kariné Ramos Blanco con más de tres años de experiencia en el tema de Ingeniería de Requisitos y creadora del Proceso Base de Requisitos, un resumen con las propuestas principales de clases, propiedades y relaciones que se definieron; así como un cuestionario con preguntas con el objetivo de validar el diseño de la Ontología.

En el cuestionario, que aparece en el Anexo 5, se evaluó la estructura, arquitectura y diseño de la Ontología. Además se valoró la organización de los términos y conceptos, así como el vocabulario empleado y la relación entre las clases y jerarquía de clases. También se tuvo en cuenta la posibilidad real del uso y aplicación de la Ontología en el Modelo Cubano de Calidad.

En la siguiente tabla se refleja los resultados obtenidos en cada una de las preguntas del cuestionario.

**Tabla 4: Resultados del cuestionario de validación.**

Pregunta \ Evaluación	Si	No	No Sé
¿La estructura, arquitectura y diseño de la Ontología modelan el dominio para el cual fue creada?	✓		
¿Los términos y conceptos representados en la Ontología están organizados sin redundancias y consistentes con los requerimientos para los cuales fue creada?	✓		
¿El vocabulario empleado para representar el conocimiento cubre completamente los conceptos y relaciones del dominio?	✓		
¿Es posible el uso de la Ontología en el Modelo Cubano de Calidad de Software?	✓		
¿Considera correcta la jerarquía de clases y la relación que existe entre las clases que componen el esquema de la Ontología?	✓		
Considera que el modelo del Proceso Base de Requisitos se ve reflejado	✓		

en el esquema de la Ontología propuesta.			
--	--	--	--

### **CONCLUSIONES DEL CAPÍTULO**

En este capítulo se realizó la validación del Esquema de la Ontología a través de razonadores y de su correspondencia con el Proceso Base de Requisitos, arribando a los siguientes resultados:

- Se escogió el razonador FACT++, por ser el razonador que trae la herramienta Protégé por defecto, es decir, es de código abierto, todavía tiene mantenimiento y sigue en desarrollo. Es un buen razonador para la TBox de una Ontología y fácil su utilización.
- Tras aplicarse el razonador se encontraron cuatro errores de escritura en las clases y dos errores por clases incompatibles, los cuales fueron solucionados en su totalidad quedando la Ontología libre de errores detectables por el razonador.
- Se elaboró y aplicó un cuestionario a una especialista en la disciplina de la Ingeniería de Requisitos con el objetivo de validar el diseño de la Ontología en correspondencia al Proceso Base de Requisitos. Luego de aplicada esta encuesta, la Ontología quedó validada para el dominio al cual hace referencia, con un cien por ciento de aceptación.

## **Conclusiones Generales**

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

- Se obtuvo el diseño de una Ontología que modela la información en el dominio de la Ingeniería de Requisitos a partir del Proceso Base de Requisitos definido para el Modelo Cubano de Calidad para el desarrollo de aplicaciones informáticas. La representación gráfica se realizó a partir de pasos específicos y se utilizó Protégé como herramienta de modelado.
- La Ontología diseñada provee la posibilidad de representar los principales conceptos relacionados con el dominio en cuestión y está elaborada con una sintaxis comprensible por los ordenadores que facilita almacenar ese conocimiento de forma tal que pueda ser utilizado por aplicaciones informáticas y sistemas de información basados en Ontologías.
- La propuesta fue validada ejecutando diferentes iteraciones con el razonador Fact++ lo que permitió evaluar satisfactoriamente la información representada en la Ontología en cuanto a las clases incompatibles y errores de escritura en los nombres de los términos.
- El esquema ontológico diseñado fue validado además, desde el punto de vista de su concordancia con el Proceso Base de Requisitos entregado para su representación. Esta validación fue realizada por la creadora del proceso anteriormente mencionado con un cien por ciento de aceptación.

## **Recomendaciones**

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

- Implementar el agente semántico que permita consultar la información disponible en esta Ontología para lograr su perfeccionamiento y puesta en práctica.
- Aplicar la propuesta de manera flexible, ajustándola a las necesidades del entorno.
- Integrar la Ontología con las restantes áreas que forman parte del Modelo Cubano de Calidad.

## Referencias Bibliográficas

1. *Gestion del Conocimiento. MODELO ANDERSEN*. [En línea] [Citado el: 7 de Junio de 2011.] [http://www.gestiondelconocimiento.com/modelos\\_arthur.htm](http://www.gestiondelconocimiento.com/modelos_arthur.htm).
2. **Matos, Guillermo; Chalmeta, Ricardo; Coltell , Oscar**. Scielo. Información tecnológica. *Scielo. Información tecnológica*. [En línea] Centro de Información Tecnológica, 2011. [Citado el: 6 de Junio de 2011.] [http://www.scielo.cl/scielo.php?pid=S0718-07642006000200011&script=sci\\_arttext](http://www.scielo.cl/scielo.php?pid=S0718-07642006000200011&script=sci_arttext).
3. **Escorial, Eduardo Bueno Presidente de Euroforum**. LA GESTIÓN DEL CONOCIMIENTO: NUEVOS PERFILES PROFESIONALES. *LA GESTIÓN DEL CONOCIMIENTO: NUEVOS PERFILES PROFESIONALES*. [En línea] 24 de Junio de 2009. [Citado el: 07 de Junio de 2011.] <http://www.sedic.es/bueno.pdf>.
4. Gestión de Conocimiento y Producción Ajustada. *Gestión de Conocimiento y Producción Ajustada*. [En línea] 2010. [Citado el: 07 de Junio de 2011.] [http://www.elprisma.com/apuntes/ingenieria\\_industrial/gestiondelconocimientoyproduccionajustada/default4.asp](http://www.elprisma.com/apuntes/ingenieria_industrial/gestiondelconocimientoyproduccionajustada/default4.asp).
5. **Ramírez Céspedes, Zulia**. *Las ontologías como herramienta en la Gestión del Conocimiento*. Ciudad de La Habana : s.n.
6. **Bosch Mela**. *Las ontologías del web semántico: su uso como espacios para la pluralidad y la diversidad. 1er foro Social de Información, Documentación y Bibliotecas*. [En línea] 2008. [Citado el: 8 de junio de 2011.] <http://www.inforosocial.org/ponencias/eje02/12.pdf>.
7. **Gruber. T. R.** *A Translation Approach to Portable Ontology Specifications*. s.l.: Knowledge Acquisition, 2003, págs. 199-220.
8. **Tramullas , Jesús**. *Agentes y ontologías para el tratamiento de información: clasificación y recuperación en Internet*. [En línea] [Citado el: 9 de junio de 2011.] <http://www.tramullas.com/papers/isko99.pdf>.
9. **Lozano Tello, Adolfo**. *Ontologías en la Web Semántica. I Jornadas de Ingeniería Web*. [En línea] 2009. [Citado el: 9 de junio de 2011.] [www.inf.udec.cl/andrea/cursos/retrieval/web.pdf](http://www.inf.udec.cl/andrea/cursos/retrieval/web.pdf).

10. **Swartout , B; Patil, R; Knight, K; Russ , T;**. Towards Distributed Use of Large-Scale Ontologies. AAAI'97 Spring Symposium Series on Ontological Engineering. 1997, págs. 138-142.
11. **Bernaras, A; Laresgoti, I; Corera, J;**. Building and Reusing Ontologies for Electrical Network Applications". European Conference of Artificial Intelligence (ECAI'96). Budapest. Hungary : Wiley and Sons Ltd, 1996, págs. 298-302.
12. **Mizoguchi, R.** Knowledge Acquisition and Ontology. Proceedings of the KB&KS'93. Tokyo : s.n., 1993, págs. 121-128.
13. **CSD.** *FOG: GENERACIÓN DE ONTOLOGÍAS DIFUSAS*. [En línea] CSD S.A. [Citado el: 6 de febrero de 2011.] <http://www.csd.com.es/FOG/index.html?seccion=inicio>.
14. **Van Heijst; Schreiber; Wieling.** *Using Explicit Ontologies*. s.l. : International Journal of Human an Computer Studies.
15. **Guarino.** *Formal Ontology and Information Systems*. Trento, Italy : Proc. Of the 1st International Conference.
16. **NONAKA; TAKEUCHI.** *The Knowledge-Creating Company*. Oxford. 2008.
17. **Carrera.** Adquisición semi-automática del conocimiento: una arquitectura preliminar. *Adquisición semi-automática del conocimiento: una arquitectura preliminar*. [En línea] Universidad de Vigo, 2007. [Citado el: 9 de Diciembre de 2010.] <http://www.grupocole.org/cole/library/ps/Car2007a.pdf>.
18. *IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology*. [En línea] [Citado el: 24 de Mayo de 2008.] <http://standards.ieee.org/findstds/standard/610.12-1990.html>.
19. **Pressman, Roger S.** *Ingeniería de Software: Un enfoque práctico*. Sexta edición. México DF : McGraw Hill , 2006.
20. **Ortas , A.** *Aproximación a la Ingeniería de Requerimientos*. Uruguay : Universidad ORT Uruguay, 2001.

## Bibliografía

1. *Gestión del Conocimiento. MODELO ANDERSEN*. [En línea] [Citado el: 7 de Junio de 2011.] [http://www.gestiondelconocimiento.com/modelos\\_arthur.htm](http://www.gestiondelconocimiento.com/modelos_arthur.htm).
2. **Matos, Guillermo; Chalmeta, Ricardo; Coltell , Oscar**. Scielo. Información tecnológica. *Scielo. Información tecnológica*. [En línea] Centro de Información Tecnológica, 2011. [Citado el: 6 de Junio de 2011.] [http://www.scielo.cl/scielo.php?pid=S0718-07642006000200011&script=sci\\_arttext](http://www.scielo.cl/scielo.php?pid=S0718-07642006000200011&script=sci_arttext).
3. **Escorial, Eduardo Bueno Presidente de Euroforum**. LA GESTIÓN DEL CONOCIMIENTO: NUEVOS PERFILES PROFESIONALES. *LA GESTIÓN DEL CONOCIMIENTO: NUEVOS PERFILES PROFESIONALES*. [En línea] 24 de Junio de 2009. [Citado el: 07 de Junio de 2011.] <http://www.sedic.es/bueno.pdf>.
4. Gestión de Conocimiento y Producción Ajustada. *Gestión de Conocimiento y Producción Ajustada*. [En línea] 2010. [Citado el: 07 de Junio de 2011.] [http://www.elprisma.com/apuntes/ingenieria\\_industrial/gestiondelconocimientoyproduccionajustada/default4.asp](http://www.elprisma.com/apuntes/ingenieria_industrial/gestiondelconocimientoyproduccionajustada/default4.asp).
5. **Ramírez Céspedes, Zulia**. *Las ontologías como herramienta en la Gestión del Conocimiento*. Ciudad de La Habana : s.n.
6. **Bosch Mela**. *Las ontologías del web semántico: su uso como espacios para la pluralidad y la diversidad. 1er foro Social de Información, Documentación y Bibliotecas*. [En línea] 2008. [Citado el: 8 de junio de 2011.] <http://www.inforosocial.org/ponencias/eje02/12.pdf>.
7. **Gruber. T. R.** *A Translation Approach to Portable Ontology Specifications*. s.l. : Knowledge Acquisition, 2003, págs. 199-220.
8. **Tramullas , Jesús**. *Agentes y ontologías para el tratamiento de información: clasificación y recuperación en Internet*. [En línea] [Citado el: 9 de junio de 2011.] <http://www.tramullas.com/papers/isko99.pdf>.
9. **Lozano Tello, Adolfo**. *Ontologías en la Web Semántica. I Jornadas de Ingeniería Web*. [En línea] 2009. [Citado el: 9 de junio de 2011.] [www.inf.udec.cl/andrea/cursos/retrieval/web.pdf](http://www.inf.udec.cl/andrea/cursos/retrieval/web.pdf).

10. **Swartout , B; Patil, R; Knight, K; Russ , T;**. Towards Distributed Use of Large-Scale Ontologies. AAAI'97 Spring Symposium Series on Ontological Engineering. 1997, págs. 138-142.
11. **Bernaras, A; Laresgoti, I; Corera, J;**. Building and Reusing Ontologies for Electrical Network Applications". European Conference of Artificial Intelligence (ECAI'96). Budapest. Hungary : Wiley and Sons Ltd, 1996, págs. 298-302.
12. **Mizoguchi, R.** Knowledge Acquisition and Ontology. Proceedings of the KB&KS'93. Tokyo : s.n., 1993, págs. 121-128.
13. **CSD.** *FOG: GENERACIÓN DE ONTOLOGÍAS DIFUSAS*. [En línea] CSD S.A. [Citado el: 6 de febrero de 2011.] <http://www.csd.com.es/FOG/index.html?seccion=inicio>.
14. **Van Heijst; Schreiber; Wieling.** *Using Explicit Ontologies*. s.l. : International Journal of Human an Computer Studies.
15. **Guarino.** *Formal Ontology and Information Systems*. Trento, Italy : Proc. Of the 1st International Conference.
16. **NONAKA; TAKEUCHI.** *The Knowledge-Creating Company*. Oxford. 2008.
17. **Carrera.** Adquisición semi-automática del conocimiento: una arquitectura preliminar. *Adquisición semi-automática del conocimiento: una arquitectura preliminar*. [En línea] Universidad de Vigo, 2007. [Citado el: 9 de Diciembre de 2010.] <http://www.grupocole.org/cole/library/ps/Car2007a.pdf>.
18. *IEEE Std 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology*. [En línea] [Citado el: 24 de Mayo de 2008.] <http://standards.ieee.org/findstds/standard/610.12-1990.html>.
19. **Pressman, Roger S.** *Ingeniería de Software: Un enfoque práctico*. Sexta edición. México DF : McGraw Hill , 2006.
20. **Ortas , A.** *Aproximación a la Ingeniería de Requerimientos*. Uruguay : Universidad ORT Uruguay, 2001.
21. **SOFTWARE, INFRAESTRUCTURA PRODUCTIVA DIRECCIÓN DE CALIDAD DE. PRÁCTICAS RECOMENDADAS PARA LA ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE.** 2009.
22. **Sanz, Ismael y Jiménez Ruiz, Ernesto.** *Ontologías en Informática*. 2007.

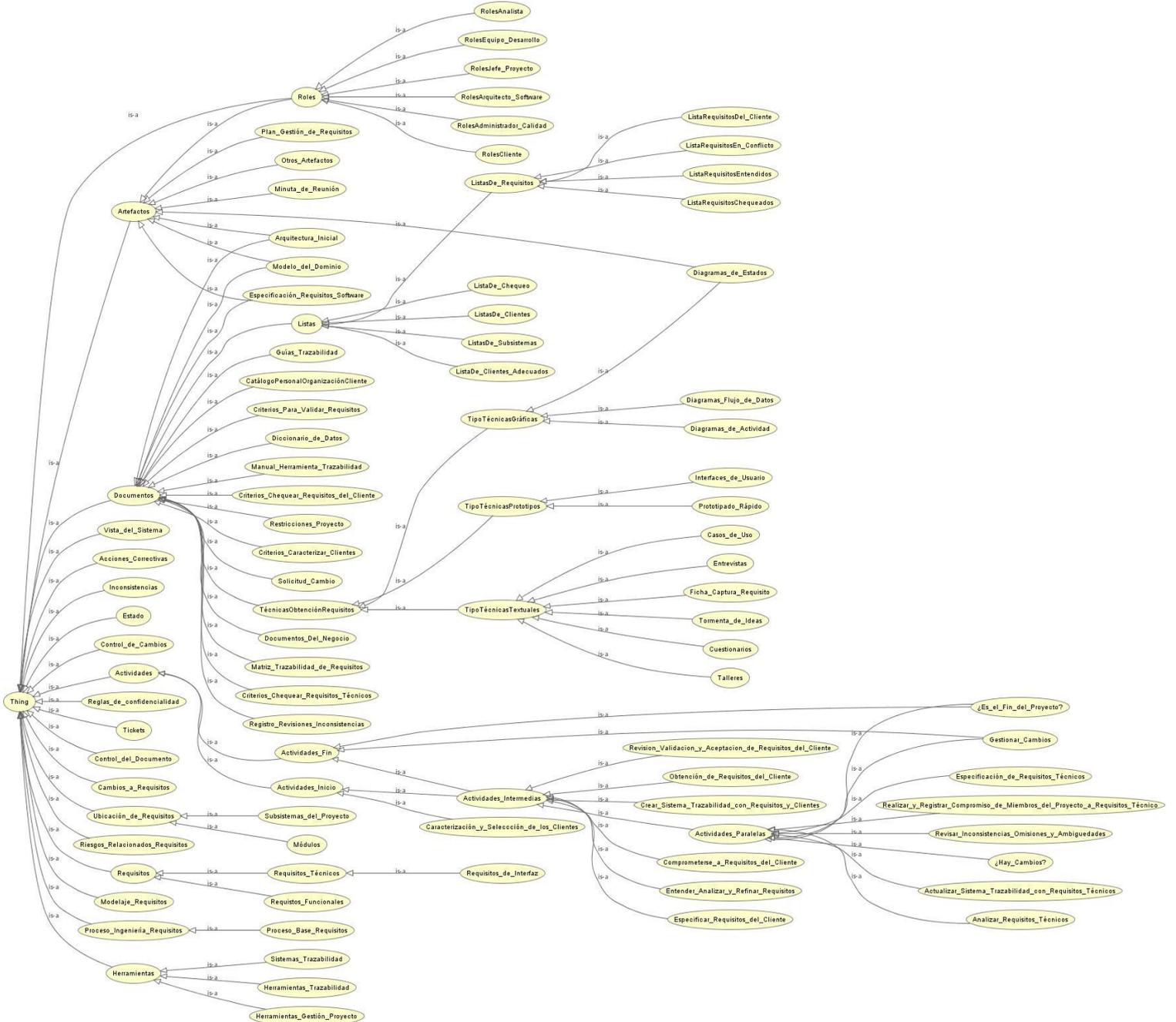
23. **Sánchez Lucas, Esther y Martínez Rubio, Rosa.** *LAS ONTOLOGÍAS Y SU APLICACIÓN EN EL ÁMBITO DE LA DOCUMENTACIÓN.* 2009.
24. **Sánchez, Diana Marcela, Cavero, José María y Marcos, Esperanza.** *Ontologías y MDA: una revisión de la literatura.* España : s.n.
25. **SAMPER ZAPATER, JOSÉ JAVIER.** *ONTOLOGÍAS PARA SERVICIOS WEB SEMÁNTICOS DE INFORMACIÓN DE TRÁFICO: DESCRIPCIÓN Y HERRAMIENTAS DE EXPLOTACIÓN.* Valencia : Universitat de Valencia Servei de Publicacions C/ Artes Gráficas, 13 bajo 46010 València Spain, 2008.
26. **Rodríguez, Claudia Milena, Cano Montaña, William y Martínez, José Mauricio.** Un Estado del Arte . *Un Estado del Arte* . [En línea] Enero de 2010. [Citado el: 5 de Abril de 2011.] [http://www.usbbog.edu.co/Nuestra\\_Universidad/Publicaciones/Ingenium/Semantics.pdf](http://www.usbbog.edu.co/Nuestra_Universidad/Publicaciones/Ingenium/Semantics.pdf).
27. **Rainer, Javier, Gómez, Jaime y Galán, Ramón.** *Ontología para la explotación del conocimiento de un Robots Guía.* Madrid : s.n., 2008.
28. **Pedraza Jiménez, Rafael.** *Generación Semiautomática de Ontologías.* 2007.
29. **Neuland Agüero, Ing Dennis.** *Áreas del aseguramiento de la calidad.* 2009.
30. **Hernández Ramírez, Haliuska y Saiz Noeda, Maximiliano.** *Ontologías mixtas para la representación conceptual de objetos de aprendizaje.* Ciudad de la Habana y España : s.n.
31. **Hernández, M.C. Miriam Zulma Sánchez.** Planificación y Modelado. Procesos de la Ingeniería. *Planificación y Modelado. Procesos de la Ingeniería.* [En línea] [Citado el: 29 de Noviembre de 2010.] <http://www.scribd.com/doc/26752264/PlanificaciOn-y-Modelado-Procesos-de-La-Ingenieria>.
32. **García Peñalvo, Francisco José.** Web Semántica y Ontologías. *Web Semántica y Ontologías.* [En línea] 2 de Noviembre de 2008. [Citado el: 5 de Diciembre de 2011.] <http://zarza.usal.es/~fgarcia/doctorado/iuce/WSemantica.pdf>.
33. **F. Noy, Natalya y L. McGuinness, Deborah.** *Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología.* Stanford University, Stanford : s.n., 2008.
34. **Dríxo.** Kootbox. *kootbox.* [En línea] Wendy Rodríguez, 5 de Enero de 2008. [Citado el: 29 de Noviembre de 2010.] <http://www.kootbox.com/content/view/40/2/>.

35. **Contreras, Jesús y Martínez Comeche, Juan Antonio.** *TUTORIAL ONTOLOGÍAS*. Madrid : Universidad Complutense , 2008.
36. **Chun Pong, Leung.** Ontología combinada superior sugerida. *Ontología combinada superior sugerida*. [En línea] Octubre de 2008. [Citado el: 3 de Diciembre de 2010.] [http://www.encyclopediaspana.com/Ontolog%C3%ADa\\_combinada\\_superior\\_sugerida.html](http://www.encyclopediaspana.com/Ontolog%C3%ADa_combinada_superior_sugerida.html).
37. **Pedraza Jiménez, Rafael ; Codina, Lluís ; Rovira, Cristòfol.** Web semántica y ontologías en el procesamiento de la información documental En: El profesional de la información. *Web semántica y ontologías en el procesamiento de la información documental En: El profesional de la información*. [En línea] 16 de Noviembre-Diciembre de 2007. [Citado el: 9 de Diciembre de 2010.] <http://www.lluiscodina.com/webSemanticaOntologias2007.pdf>.
38. UDLAP. *UDLAP*. [En línea] 29 de Mayo de 2008. [Citado el: 12 de Octubre de 2010.] [http://webcache.googleusercontent.com/search?q=cache:UijS6ADGieYJ:catarina.udlap.mx/u\\_dl\\_a/tales/documentos/mcc/sanchez\\_l\\_se/capitulo\\_4.html+Ontolog%C3%ADas+y+su+representacion+jer%C3%A1rquica&cd=2&hl=es&ct=clnk&gl=cu](http://webcache.googleusercontent.com/search?q=cache:UijS6ADGieYJ:catarina.udlap.mx/u_dl_a/tales/documentos/mcc/sanchez_l_se/capitulo_4.html+Ontolog%C3%ADas+y+su+representacion+jer%C3%A1rquica&cd=2&hl=es&ct=clnk&gl=cu).
39. Tormenta de Ideas. *Tormenta de Ideas*. [En línea] [Citado el: 15 de 01 de 2011.] [http://www.fundibeq.org/opencms/export/sites/default/PWF/downloads/gallery/methodology/tools/tormenta\\_de\\_ideas.pdf](http://www.fundibeq.org/opencms/export/sites/default/PWF/downloads/gallery/methodology/tools/tormenta_de_ideas.pdf).
40. *SISTEMA DE EVOLUCION DE ONTOLOGÍAS PARA LA GESTIÓN DE CONOCIMIENTO*. 2008.
41. **Wittens, Steven ; Stefan, Nagtegaal;** Qué es Ingeniería de Requisitos (IR). *Qué es Ingeniería de Requisitos (IR)*. [En línea] 27 de Marzo de 2008. [Citado el: 5 de Diciembre de 2010.] <http://danielvn7.wordpress.com/2008/03/27/%C2%BFque-es-ingenieria-de-requisitos-ir/>.
42. **Facultad de Ingeniería (UBA).** *Proyectos Informáticos. Administración y Control de Proyectos I*. 2007.
43. **Quispe-Otazu, Rodolfo.** Procesos De La Ingeniería De Requerimientos. *Procesos De La Ingeniería De Requerimientos*. [En línea] 12 de Enero de 2010. [Citado el: 4 de Diciembre de 2010.] <http://www.mitecnologico.com/Main/ProcesosDeLaIngenieriaDeRequerimientos>.
44. **Jaime F. Castillo.** *Perfil del Ingeniero de Requerimientos*. 2010.

45. **Blaquier Ascaño, Dra. Marta Isabel.** *OnProc: Una Ontología para el Proceso Jurídico.* Ciudad de la Habana : s.n.
46. La Arquitectura de Software. *La Arquitectura de Software.* [En línea] [Citado el: 17 de 01 de 2011.] <http://www.arquitecturadesoftware.com/>.
47. *Introducción a la Disciplina de Requisitos de RUP.* 2010.
48. Importancia De La Captura De Requerimientos En El Proceso Productivo Del Proyecto Ciscop De La Facultad Regional De Artemisa. Importance Of The Captu. *Importancia De La Captura De Requerimientos En El Proceso Productivo Del Proyecto Ciscop De La Facultad Regional De Artemisa. Importance Of The Captu.* [En línea] 10 de 11 de 2008. [Citado el: 20 de 01 de 2010.] <http://www.articuloz.com/comunicaciones-articulos/importancia-de-la-captura-de-requerimientos-en-el-proceso-productivo-del-proyecto-ciscop-de-la-facultad-regional-de-artemisa-importance-of-the-captu-636738.html>.
49. **IEEE Std 1233.** *Guía para el desarrollo de Especificaciones de Requerimientos de Sistemas.* 1998.
50. Glossarium BITri. *Glossarium BITri.* [En línea] José María Díaz Nafría, Noviembre de 2009. [Citado el: 23 de octubre de 2010.] <http://sites.google.com/site/glosariobitrum/Home/ontologia>.
51. *Gestión del Conocimiento.* [En línea] Sebastian Schmiege, 12 de mayo de 2009. [Citado el: 25 de Noviembre de 2010.] <http://gestiondelconocimientokm.wordpress.com>.
52. **VÉLEZ LEÓN, PAULO.** *Aproximaciones a la Ontología del Arte.* Buenos Aires : s.n., 2006.

# Anexos

## Anexo 1. Esquema Final de la Ontología.



**Anexo 2. Lista de clases de la Ontología.**

Clases y Subclases
Thing
Proceso_Ingenieria_Requisito
Proceso_Base_Requisitos
Acciones_Correctivas
Actividades
Actividades_Fin
Actividades_Inicio
Actividades_Intermedias
Gestionar_Cambios
Es_El_Fin_Del_Proyecto
Actividades_Paralelas
Comprometerse_A_Requisitos_Del_Cliente
Crear_Sistema_Trazabilidad_Con_Requisitos_Y_Clientes
Entender_Analizar_Y_Refinar_Requisitos
Especificar_Requisitos_Del_Cliente
Obtencion_Requisitos_Del_Cliente
Revision_Validacion_Y_Aceptacion_De_Requisitos_Del_Cliente
Actualizar_Sistema_Trazabilidad_Con_Requisitos_Tecnicos
Analizar_Requisitos_Tecnicos
Especificacion_Requisitos_Tecnicos
Realizar_Y_Registras_Compromiso_De_Miembros_Del_Proyecto_A_Requisitos_Tecnicos
Revizar_Inconsistencias_Omisiones_Y_Ambiguedades
Hay_Cambios
Caracterizacion_Y_Seleccion_De_Los_Clientes
Artefactos
Arquitectura_Inicial
Diagramas_De_Estados
Minuta_De_Reunion
Especificacion_Requisitos_Software
Modelo_Del_Dominio
Otros_Artefactos
Plan_Gestion_De_Requisitos
Roles
RolesAdministrador_Calidad
RolesAnalista
RolesArquitecto_Software
RolesCliente
RolesEquipo_Desarrollo
RolesJefe_Proyecto
Cambios_A_Requisitos
Complejidad
Control_De_Cambios
Control_Del_Documento
Documentos

Arquitectura\_Inicial  
CatalogoPersonalOrganizacionCliente  
Criterios\_Caracterizar\_Clientes  
Criterios\_Chequear\_Requisitos\_Tecnicos  
Criterios\_Chequear\_Requisitos\_Cliente  
Criterios\_Para\_Validar\_Requisitos  
Diccionario\_De\_Datos  
Documentos\_Del\_Negocio  
Especificacion\_Requisitos\_Softqare  
Guias\_Trazabilidad  
Listas  
ListaDe\_Chequeo  
ListaDe\_Clientes\_Adecuados  
ListasDe\_Clientes  
ListasDe\_Requisitos  
ListaRequisitosChequeados  
ListaRequisitos\_Cliente  
ListaRequisitos\_Conflicto  
ListaRequisitosEntendidos  
ListaDe\_Subsistemas  
Manual\_Herramienta\_Trazabilidad  
Matriz\_Trazabilidad\_De\_Requisitos  
Modelo\_Del\_Dominio  
Requisitos\_Revisiones\_Inconsistencias  
Restricciones\_Proyecto  
Solicitud\_Cambio  
TecnicasObtencionRequisitos  
TipoTecnicasGraficas  
Diagramas\_Flujo\_De\_Datos  
Diagramas\_De\_Actividad  
Diagramas\_De\_Estados  
TipoTecnicaPrototipos  
Interfas\_De\_Usuario  
Prototipo\_Rapido  
TipoTecnicasTextuales  
Casos\_De\_Uso  
Cuestionarios  
Entrevistas  
Ficha\_Captura\_Requisitos  
Talleres  
Tormenta\_De\_Ideas  
Estado  
Evaluacion  
Herramientas  
Herramientas\_Gestion\_Proyecto  
Herramienta\_Trazabilidad  
Sistemas\_Trazabilidad  
Inconsistencias  
Modelaje\_Requisitos

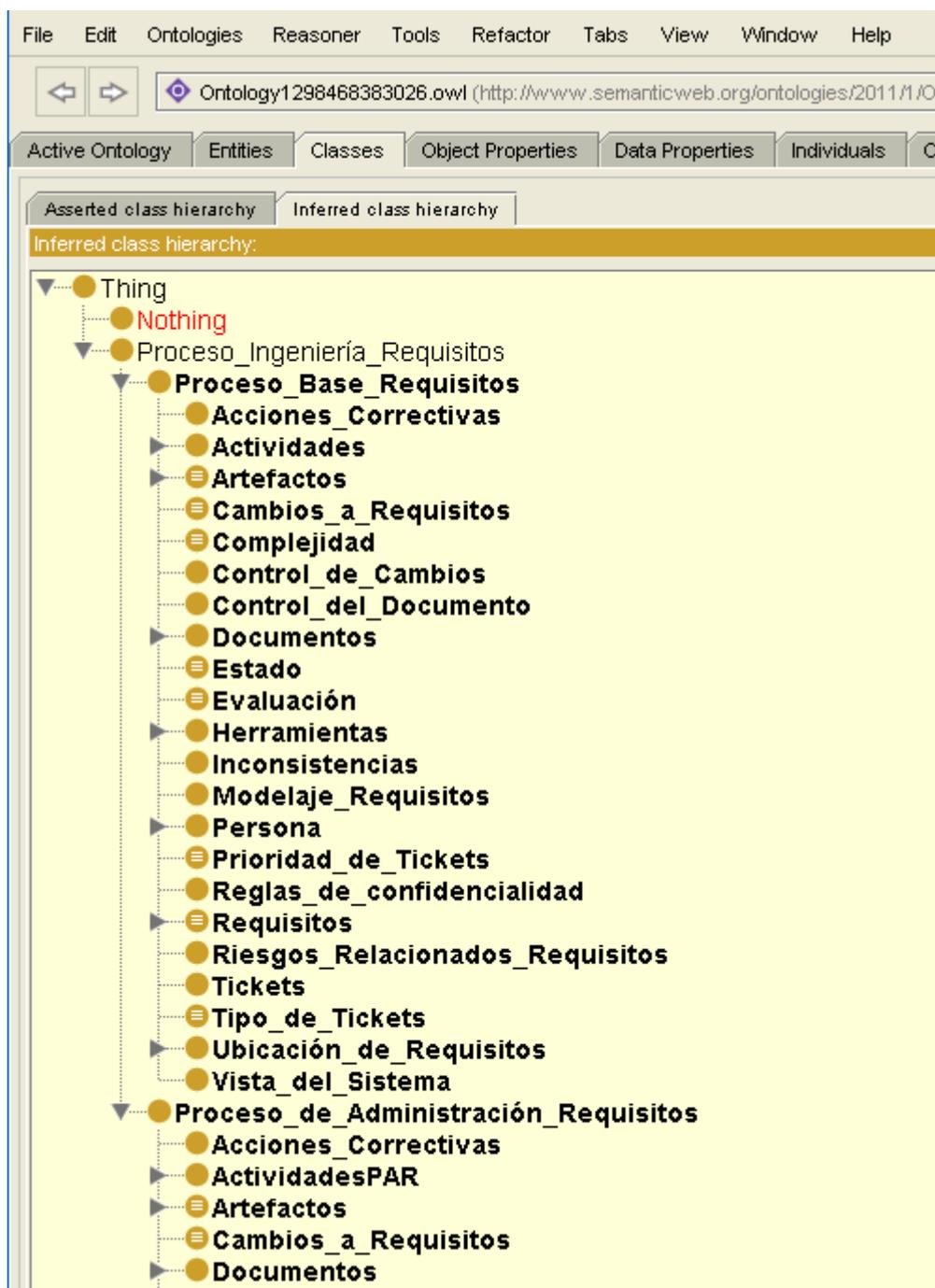
Persona
Usuario
Prioridad_De_Tickets
Regla_De_Confidencialidad
Requisitos
Requisitos_Tecnicos
Requisitos_de_Interfaz
Requisitos_Funcionales
Riesgos_Relacionados_Requisitos
Tickets
Tipo_De_Tickets
Ubicación_De_Requisitos
Modulos
Subsistemas_Del_Proyecto
Vista_Del_Sistema
Acciones_Correctivas
ActividadesPAR
Actividades_Fin_PAR
Actividades_Intermedias_PAR
Alcanzar_Compromiso_Entre_Partes_Involucradas
Asegurar_Consistencia_Entre_Requisitos_Del_Cliente_Con_Otros_Productos_Del_Proyecto
Llvar_Seguimiento_De_Rquisitos
Efectuar_De_ManeraControlada_Cambios_Realizados_En_Requisitos
Actividades_Inicio_PAR
Conocer_Y_Entender_Requisitos_Del_Cliente

### Anexo 3. Lista de Propiedades (Objeto y Datos) de la Ontología.

Propiedades	
Objeto	Datos
acepiRealizadoPor	DescripcionSistema
acepiTieneControl	EsNulo
acepiTieneEntrada	actualizada
acepiTieneSalida	aplicanTecnicasYObtienenRequisitosPreliminares
acercotaTieneEntrada	catalogoRequisitos
acercopaRealizadoPor	chequeado
acercopaTieneSalida	compromisoConProyecto
adecuadosParaObtener	conocimientoMercado
aprobadoPor	conocimientoNegocio
artRealizadoPor	conocimientoInformatica
artTieneEntrada	criterioSelecionMultiple
artTieneGuiaOApoyo	criterioSeleccionUnica
artTieneSalida	definicionesYAcronimos
asignadoA	describirRequisitos
astrtRealizadoPor	descripcionActores
astrtTieneEntrada	descripcionCambio
astrtTieneGuiaOApoyo	descripcionDato

astrtTieneSalida	descripcionEntidad
autorCambio	descripcionInconsistencia
cargo	disponibilidadTiempo
cercRealizadoPor	elementosRevisados
cercTieneControl	esAceptado
cercTieneEntrada	esAprobada
cercTieneSalida	esFirmadaPorPartes
chequeadosPor	esPriorizado
controlDocumentos	esRevisadaLaSolicitud
cecRealizadoPor	evaluacionImpactoDeLosCambios
crcTieneEntrada	fechaFin
crcTieneGuiaOApoyo	fechaInicio
crcTieneSalida	fechaRevision
crtrcTieneEntrada	formaDistribucion
cscRealizadoPor	fuentesInconsistencia
cscTieneEntrada	gradoAutoridad
cscTieneGuiaOApoyo	irAFin
cscTieneSalida	listaCosto
cstrcRealizadoPor	listaRendimiento
cstrcTieneGuiaOApoyo	listaTiempo
cstrcTieneSalida	nombreAtributo
dependeDe	nombreDocumento
dependeDePAR	nombreEntidad
descripcionModulos	nombreHerramienta
distribuidoEn	nombreProyecto
earrRealizadoPor	nombreVistaSistema
earrTieneEntrada	obtienenPrimeraVersionDeRestriccionesDelProyecto
earrTieneGuiaOApoyo	porcentajeRealizado
earrTieneSalida	priorizacionDeRequisitos
elaboradoPor	proveedorRequisitos
emccrrRealizadoPor	queEvidencia
emccrrTieneControl	restriccionesClasesNoValidad
emccrrTieneEntrada	restriccionesClasesValidad
emccrrTieneSalida	seAceptanTicketsYRealizacionDeRequisitos
ercRealizadoPor	seActualizaHerramientaDeTrazabilidad
ercTieneEntrada	seAcuerdanClientesAdecuados
ercTieneGuiaOApoyo	seAsignanRealizacionDeRequisitosDelClienteMedianteTickets
ercTieneSalida	seAsignanRealizacionDeRequisitosTecnicosMedianteAsignacionTicke
ertRealizadoPor	ts
ertTieneEntrada	seAsignaTareaYVerificarCumpliminatoDelCambio
ertTieneSalida	seCaracterizaCadaCliente
esManualDe	seChequeanLosRequisitorTecnicos
expresanPuntoVista	seChequeanLosRequisitosDelCliente
firmadoPor	seCreaCatalogo
hcgcRealizadoPor	seDetectanYMonitoreanLasInconformidades
hcgcTieneEntrada	seDistribuyenRequisitosEnModulosOSubistemas
hcgcTieneGuiaOApoyo	seDocumentaLaSolicitudDeUnCambio
hcgxTieneSalida	seEspecificanRequisitosDelCliente
insertadoEn	seEstableceRealizacionBidireccional

IsrRealizadoPor	seExplicanRequisitosADesorrollarYProveenRequisitosModeladosseFirmaLaEspecificacionDeRequisitos
IsrTieneControl	seGestionaElCambio
IsrTieneEntrada	seIdentificanRequisitosDeInterfaz
IsrTieneSalida	seIdentificanYEvaluanRiesgos
nombrePersona	seInsertanRequerimientosPactadosYClientes
orcRealizadoPor	seModelanLosRequisitos
orcTieneGuiaOApoyo	sePlanificaDistribucionDeTickets
orcTieneSalida	sePriorizaRequisitos
proponen	seRealizaReunion
prototipadoRapido	seRealizaReunionPAR
relacionadoCon	seRefinanRequisitos
requeridoPor	seRegresaAEspecificacionDeRequisitosTecnicos
requeridoPorPAR	seRevisanInconsistencias
resultadosCuestionarios	seRevisanRequisitosEnCoflictosYSeLleganAAcuerdosBilaterales
resultadoEntrevista	seReunenParaValidarEspecificacionDeRequisitos
resultadosTalleres	seReunenYSeleccionanTecnicas
ResultadosTormentaIdeas	seTomaDesicionesYComunicarlaAInteresados
riaRealizadoPor	seleccionInicialDeClientes
riaTieneGuiaOApoyo	sumaTotal
riaTieneSalida	tiempoDedicado
rrcmpTRealizadoPor	tiempoEstimado
rrcmpTieneEntrada	tieneAdscripcion
rrcmpTieneSalida	tieneAlcance
rvarcRealizadoPor	tieneClasificacion
rvarcTieneEntrada	tieneContacto
rvarcTieneSalida	tieneDesarrollo
sirveParaCaracterizar	tieneDescripcion
sonAceptadosPor	tieneEstadoAceptada
tieneArtefacto	tieneEstadoAsignada
tieneControlCambio	tieneEstadoCerrada
tieneEstado	tieneEstadoNueva
tieneEvaluacion	tieneEstadoResuelta
tieneInterfaces	tieneFecha
tieneManual	tieneImpacto
tienePrioridad	tieneIntroduccion
tieneProcedimiento	tieneLenguajeTecnico
tieneReglasConfidencialidad	tieneNumero
tieneResponsables	tieneObjetivo
tieneSeccion	tieneTema
tieneUbicacion	tieneTextoEstructurado
tieneVistaSistema	tieneTipoCambio
tienenUbicacion	tieneTitulo
	tieneTituloReferencia
	tieneVersion
	tipoDato
	indiceContenidos

**Anexo 4. Resultado final de la validación de la Ontología a través del razonador Fact ++.**

---

**Anexo 5. Cuestionario para determinar la evaluación de la Ontología con respecto a la Ingeniería de Requisitos.**

1. ¿La estructura, arquitectura y diseño de la Ontología modelan el dominio para el cual fue creada?

Si.       No.       No sé.

2. ¿Los términos y conceptos representados en la Ontología están organizados sin redundancias y consistente con los requerimientos para los cuales fue creada?

Si.       No.       No sé.

3. ¿El vocabulario empleado para representar el conocimiento cubre completamente los conceptos y relaciones del dominio?

Si.       No.       No sé.

4. ¿Es posible el uso de la Ontología en el Modelo Cubano de Calidad de Software?

Si.       No.       No sé.

5. ¿Considera correcta la jerarquía de clases y la relación que existe entre las clases que componen el esquema de la Ontología?

Si.       No.       No sé.

6. ¿Considera que el modelo del Proceso Base de Requisitos se ve reflejado en el esquema de la Ontología propuesta?

Si.       No.       No sé.

## **Glosario de Términos**

**Conocimiento explícito:** Es aquel que puede ser expresado con palabras y números, y puede ser fácilmente comunicado y compartido bajo la forma de datos, fórmulas científicas, procedimientos codificados o principios universales.

**Conocimiento tácito:** Es la capacidad de la mente humana para dar sentido a la colección de experiencias vividas y a conectar pausas desde el pasado al presente y al futuro. Es aquel conocimiento no visible, muy personal y difícil de formalizar y de comunicar o compartir con otras personas; incluye elementos tales como los puntos de vista subjetivos o las intuiciones.

**Gestión del Conocimiento:** Proceso sistemático de buscar, organizar, filtrar y presentar la información con el objetivo de mejorar la comprensión de las personas en un área específica de interés.

**Ingeniería de Requisitos:** Ayuda a los ingenieros de software a entender mejor el problema en cuya solución trabajarán. Incluye el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, qué es lo que el cliente quiere y cómo interactuarán los usuarios finales con el software.

**Interoperabilidad:** La posibilidad de que distintos tipos de ordenadores, redes, sistemas operativos, y aplicaciones trabajen juntos de forma eficaz, sin comunicación previa, de tal forma que puedan intercambiar información de manera útil y con sentido.

**Metafísica:** Es la parte de la filosofía que trata del ser en cuanto tal, y de sus propiedades, principios y causas primeras. Modo de discurrir con demasiada sutileza en cualquier materia. Cosa que así se discurre.

**Método:** Es una operación o función que está asociada a un objeto y que tiene permiso para manipular los datos del objeto.

**Modelo:** Es la representación real de los datos obtenidos a partir de la información disponible. Ejemplos son el modelo de estructura y el modelo de estilo de que representan la estructura analítica y la información de estilo asociada a un documento. El modelo podría ser un árbol, o un grafo orientado, o cualquier otra.

**Ontolingua:** Es un lenguaje que permite construir, publicar y compartir Ontologías.

**Ontología:** Es una base de datos que describe los conceptos generales o sobre un dominio, algunas de sus propiedades y cómo los conceptos se relacionan unos con otros.

**OWL:** Lenguaje de Ontología Web. Es el acrónimo del inglés Ontology Web Language.

**Razonadores semánticos:** Son aplicaciones computacionales que permiten realizar inferencias y generar conocimiento a partir de un conjunto de axiomas y hechos.

**Taxonomía:** Es un conjunto organizado de términos utilizado para organizar información y pensado principalmente para poder navegar por él. En otras palabras, una taxonomía simplemente exige que sus componentes estén organizados de manera que se puedan "recorrer". Su rasgo fundamental y definitorio es, por tanto, su finalidad de exploración. Son usadas, por ejemplo, para la categorización y la organización de la información en sitios web.

**Web Semántica:** Es una extensión de la web actual que cuenta con información debidamente estructurada, lo que proporciona un significado bien definido. Mejora la forma en la que las máquinas y las personas trabajan en cooperación.

**XML:** El lenguaje extensible de marcas es un metalenguaje que nació como forma restringida del SGML (Standard Generalized Markup Language).