



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 3

**IMPLEMENTACIÓN 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.**

AUTOR: ROHANDY FELIPE ALFONSO

TUTOR(A): ING. LISSETTE RODRÍGUEZ VERDECIA

CIUDAD DE LA HABANA

JUNIO, 2011

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo 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 _____.

Rohandy Felipe Alfonso

Lissette Rodríguez Verdecia

Datos de Contacto

Ing. Lissette Rodríguez Verdecia

Centro de trabajo: Universidad de Ciencias Informáticas

Categoría docente: Instructora.

Asignatura que imparte: Práctica Profesional II

Aspirante al grado científico de máster en ciencias.

Línea de investigación: Ingeniería de Requisitos y Ontologías.

Participación en eventos: UCIENCIA 2008, UCIENCIA 2010, FORDES 2010, Informática 10.

Años de graduada: 3 años.

Agradecimientos

Quisiera agradecer a Fidel y a la Revolución por haberme dado la oportunidad de estudiar en esta universidad y convertirme en lo que soy hoy.

A todas las personas que me ayudaron en la realización de este trabajo a mi tutora Lissette Rodríguez, a Aleyda Isabel Labrada, a Delvis Echeverría; sin ellos no hubiese sido posible su culminación.

A los que me escucharon en cada momento de estrés y desánimo: Ari, Lisbe, Yordy, Yen, Ali, JJ, los Pedros, en fin, a todos mis compañeros de grupo en los diferentes cursos y a mis compañeros de apartamento en estos 5 años por soportarme.

Dedicatoria

A mi madre que ha sido la luz de mi vida, que me ha guiado por buenos caminos, que siempre ha confiado en mí, que ha estado a mi lado en las buenas y en las malas, es por ella que seré un profesional para hacer realidad su sueño.

A quien no es mi papá biológico pero que siempre ha estado a mi lado ayudándome, aconsejándome, llevándome por buenos caminos.

A mis abuelos que han sido perseverantes en mi formación, que han vivido por mí, que han sentido cada minuto de este anhelo conmigo como si estuvieran en mi piel.

A mi esposa que ha tenido la paciencia y la dedicación para estar sola en momentos difíciles con nuestro bebé, por estar siempre pendiente de mi escuela y no dejarme hacer locuras.

Por último y no por ser el menos importante sino todo lo contrario, a mi hijo, que es lo más lindo que me ha dado la vida, todo esto es para él.

Resumen

El aumento del desarrollo de software en Cuba trae consigo la necesidad de la certificación de entidades con respecto a la calidad del software. Para ello se pretende realizar un Modelo Cubano de Calidad para el Desarrollo de Aplicaciones Informáticas en la Industria Cubana de Software, el cual tendrá como valor agregado la gestión del conocimiento. Este modelo será apoyado en una ontología como herramienta para la gestión del conocimiento y perfectamente aplicable a la Ingeniería de Software y en particular a la Ingeniería de Requisitos que es una de las áreas claves que integrará el mismo. Por lo que se cuenta con una ontología especificada mediante un mapa conceptual, pero se hace necesaria su implementación con una herramienta automatizada, para que esta pueda ser reutilizada, siendo esto el resultado final de este trabajo.

PALABRAS CLAVES: Gestión del Conocimiento, Ingeniería de Requisitos, Ontologías.

Índice

RESUMEN..... VI

INTRODUCCIÓN..... 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA. 6

 1.1 Introducción al Capítulo..... 6

 1.2 Gestión del Conocimiento 6

 1.3 Herramientas para la Gestión del Conocimiento 7

 1.4 Ingeniería de Requisitos..... 7

 1.5 Ontologías..... 8

 1.5.1 Tendencias de las Ontologías 8

 1.5.2 Definiciones de Ontologías..... 9

 1.5.3 Clasificación de ontologías 11

 1.5.4 Conceptos claves en relación con las ontologías 12

 1.5.5 Aplicaciones y usos de las ontologías 13

 1.6 Lenguajes..... 16

 1.7 Herramientas para construir ontologías 20

 1.8 Metodologías para el Desarrollo de Ontologías 22

 1.8.1 Pasos específicos para el desarrollo de ontologías..... 24

 1.9 Aplicaciones o soluciones a reutilizar..... 26

 1.10 Conclusiones del Capítulo..... 27

CAPÍTULO 2: IMPLEMENTACIÓN DE LA ONTOLOGÍA. 28

 2.1 Introducción al capítulo 28

 2.2 Modelo conceptual 28

 2.3 Formas de Representación de las Ontologías 29

 2.4 Modelo representativo de la ontología 31

 2.5 Desarrollo de la ontología 33

 2.6 Principios seguidos en la construcción de la ontología 43

 2.7 Patrones de diseño utilizados en la construcción de la ontología 44

 2.8 Paradigma de modelado utilizado 44

 2.9 Conclusiones del capítulo 45

CAPÍTULO 3: VALIDACIÓN TEÓRICA DE LA ONTOLOGÍA..... 46

 3.1 Introducción del capítulo 46

 3.2 Validación para ontologías de dominio..... 46

 3.3 Conclusiones del capítulo 53

CONCLUSIONES 54

RECOMENDACIONES.....	55
BIBLIOGRAFÍA.....	56
GLOSARIO DE TÉRMINOS	69

Introducción

En la actualidad 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. Cuando se habla de calidad del software se hace referencia al conjunto de cualidades que determinan su utilidad, “es el grado en que un software cumple con los requisitos especificados: eficiencia, flexibilidad, corrección, mantenimiento, seguridad e integridad”. (2011) Existen una variedad de modelos de Calidad para guiar el proceso de desarrollo de un software a continuación se presentan algunos de ellos: CMMI, orientado a la mejora de procesos en diferentes niveles de madurez, la Norma ISO/IEC 12007, orientada al proceso del ciclo de vida del software, la Metrica3 para el modelado y la implementación, la ISO 15504, modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software.

El creciente y cada vez más acelerado desarrollo de la informática en Cuba ha impuesto el desarrollo de un modelo de calidad que brinde a la Industria Cubana de Software una guía práctica para el mejor desempeño de sus procesos de desarrollo. 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, es el principal responsable de la creación del mismo. 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 existentes.

Según la Real Academia Española de la Lengua el conocimiento es “entendimiento, inteligencia, razón natural”. Pueden encontrarse tantas definiciones como autores se consulten, pero se puede afirmar, que el conocimiento puede ser visto como la evolución natural de los conceptos de datos e información. Por lo que la gestión del conocimiento es la disciplina que promueve la generación, colaboración y utilización del conocimiento para el aprendizaje organizacional, generándole nuevo valor.

Dentro de las áreas claves identificadas y que formará parte del modelo cubano de calidad, se encuentra la Ingeniería de Requisitos. Como herramienta para la gestión del conocimiento y perfectamente aplicable

a la Ingeniería de Software y en particular a la Ingeniería de Requisitos se encuentran las Ontologías. La gestión del conocimiento en la Ingeniería de Requisitos aporta tres aspectos fundamentales: la reutilización de los requisitos, el incremento y medida de la calidad en los requisitos y mejora la comunicación entre los miembros del equipo.

Las ontologías constituyen la herramienta seleccionada que dotará al modelo de un valor agregado en cuanto a la gestión del conocimiento. En términos muy simples una ontología es una representación formal de un conjunto de conceptos asociados a un dominio particular y de las relaciones existentes entre estos conceptos. Una ontología es formal en el sentido de que emplea un lenguaje para su representación. Esta representación puede ser procesada automáticamente. Una ontología está asociada a un dominio determinado cuyos objetos son referidos por los conceptos y relaciones de dicha ontología. En el contexto de la informática existe un número importante de definiciones del término ontología.

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, posibilitan que este conocimiento no sea reutilizado por lo que se hace necesario la representación y organización de estos términos en una ontología. Además no existe una base de conocimientos creada que permita a los especialistas hablar un lenguaje común referente a la propia Ingeniería de Requisitos.

Por lo anteriormente planteado, tenemos como **Planteamiento del problema**:

¿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?

Objeto de estudio

Gestión del Conocimiento en la Ingeniería de Requisitos.

La investigación que se desarrolla tiene como **Objetivo general**:

Implementar la 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 que permita gestionar el conocimiento para que este pueda ser reutilizado.

El mismo tiene como **Campo de acción:**

Ontologías para la gestión del conocimiento en la Ingeniería de Requisitos.

Como **Objetivos específicos** se plantean los siguientes:

- 1- Definir los principales elementos teóricos-metodológicos para el análisis del problema a resolver.
- 2- Implementar la Ontología teniendo en cuenta los términos, relaciones, taxonomía y reglas de inferencias ya definidas asociadas a la Ingeniería de Requisitos.
- 3- Validar teóricamente la Ontología.

Para dar respuesta a la interrogante planteada en el problema científico se parte de la siguiente **Idea a defender:**

La implementación 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, contribuirá a la gestión del conocimiento en esta área.

Posible resultado:

Representación ontológica de la disciplina Ingeniería de Requisitos que garantice la gestión del conocimiento.

Para responder a esto se proponen las siguientes **tareas:**

- 1- Estudio de los diferentes conceptos relacionados con ontologías, gestión del conocimiento e Ingeniería de Requisitos.
- 2- Adquisición de conocimientos acerca de otras aplicaciones o soluciones similares, en aras de su reutilización.
- 3- Documentación de los lenguajes de desarrollo de ontologías y escoger uno de ellos para la implementación de la misma, teniendo en cuenta su futura reutilización en Sistemas de Información basados en Ontologías.
- 4- Descripción del lenguaje y herramienta a utilizar en la implementación de la ontología.

- 5- Implementación de la Ontología teniendo en cuenta los términos, relaciones, taxonomía y reglas de inferencias ya definidas asociadas a la Ingeniería de Requisitos.
- 6- Validación teórica de la Ontología.

Los **Métodos Teóricos** utilizados para la realización de este trabajo son:

Análisis Histórico Lógico: Este método es de suma importancia y es utilizado para realizar un profundo análisis de cómo ha evolucionado la gestión del conocimiento y las ontologías durante los años.

Analítico-Sintético: Este método es utilizado en el análisis de la bibliografía permitiendo el procesamiento de la información y arribando a las conclusiones prácticas y teóricas de la investigación, así como para precisar las herramientas, los lenguajes y las metodologías de desarrollo de software que se utilizarán.

Inductivo-Deductivo: Se utilizó para la realización de un estudio de la Ingeniería de Requisitos y ver todas las posibilidades que nos aporta a la hora de gestionar el conocimiento.

Modelación: Para la creación de abstracciones que explican la realidad, por ejemplo, todos los modelos y diagramas presentados en las diferentes fases de desarrollo de la ontología.

El **Método Empírico** utilizado para la realización de este trabajo es:

Experimental: Para verificar o validar una hipótesis, una teoría o un modelo.

Estructura del Trabajo de Diploma:

Para una mejor comprensión del Trabajo de diploma, el contenido ha sido desglosado en introducción, 3 capítulos, conclusiones, recomendaciones, referencias bibliográficas, los anexos, mediante los cuales se percibirá con mayor facilidad la investigación realizada y por último un glosario de términos.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA: Se plasman los conceptos fundamentales asociados al dominio del problema, se hace un análisis crítico acerca de otras aplicaciones o soluciones similares en aras de su reutilización, y la solución propuesta. También se exponen los lenguajes de desarrollo de ontologías y se escoge uno para la implementación de la misma teniendo en cuenta su futura reutilización

en Sistemas de Información basados en Ontologías. Además se detalla la herramienta a utilizar en la implementación de la ontología.

CAPÍTULO 2 IMPLEMENTACION DE LA ONTOLOGIA: Se describe la implementación de la misma teniendo en cuenta los términos, relaciones, taxonomía y reglas de inferencias ya definidas asociadas a la Ingeniería de Requisitos.

CAPÍTULO 3 VALIDACION DE LA ONTOLOGIA: En este capítulo se presenta la validación teórica realizada a la ontología.

Capítulo 1: Fundamentación teórica.

1.1 Introducción al Capítulo

En el presente capítulo se brinda una visión general de algunos conceptos relacionados con la gestión del conocimiento, la Ingeniería de Requisitos y las ontologías necesarios para comprender el tema que se aborda en este trabajo de investigación. Se muestran además, algunos lenguajes y herramientas utilizadas en la generación de ontologías, las metodologías para el desarrollo de las mismas, así como los beneficios que reporta el aprendizaje de ontologías. Se tratan además algunas ontologías ya hechas para analizar que se puede reutilizar de ellas.

1.2 Gestión del Conocimiento

El conocimiento está en la mente de cada persona por lo que no siempre está presente cuando la organización lo necesita; es por ello que surge la Gestión del Conocimiento para que todos tengan acceso a él y este pueda ser reutilizado o enriquecido. Varios autores han hecho estudios acerca de la gestión del conocimiento y cada uno ha emitido un concepto diferente aquí van algunos:

Nonaka & Takeuchi: “Es la capacidad de una organización para crear nuevo conocimiento, diseminarlo a través de la organización y expresarlo en productos, servicios y sistemas”. (Quezada, 2009)

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”. (Quezada, 2009)

ONU: “Capacidad colectiva para adquirir y crear conocimiento y ponerlo a un uso productivo para el bien común. [...] acción concertada para profundizar la comprensión y para gestionar y compartir conocimiento mucho más útil”. (Quezada, 2009)

Todas estas definiciones, concuerdan en que la Gestión del Conocimiento es la disciplina que promueve la generación, colaboración y utilización del conocimiento para el aprendizaje organizacional, generándole nuevo valor.

1.3 Herramientas para la Gestión del Conocimiento

Existe una gran variedad de herramientas inmersas en los sistemas para administración del conocimiento. Se pueden mencionar algunas como son:

- Bases de datos.
- Las redes neuronales.
- La implementación de Intranets y Extranets para la gestión de información en las organizaciones.
- Minería de datos.
- 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. Se escoge para el desarrollo de este trabajo el uso de ontologías porque son herramientas eficaces para modelar, compartir y reutilizar el conocimiento, contribuyendo a la toma de decisiones y haciendo inferencias a partir del conocimiento representado en estos sistemas.

1.4 Ingeniería de Requisitos

Algunas definiciones recientes son: “La 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” (Pressman, 2006); “La ingeniería de requisitos es el proceso de desarrollar una especificación de software. Las especificaciones pretenden comunicar las necesidades del sistema del cliente a los desarrolladores del sistema”. (Sommerville, 2005)

La ingeniería de requisitos comprende la definición de requisitos y la elaboración del modelo conceptual del sistema. La necesidad de una ingeniería de requisitos dentro de la ingeniería del software es imprescindible para obtener productos de calidad. La calidad no puede implementarse en las etapas finales de los procesos de desarrollo sino que es una característica intrínseca al propio producto,

iniciándose en las especificaciones de los productos. Este inicio de las actividades de calidad en la ingeniería de requisitos es la base para el aseguramiento de la calidad del proyecto por lo que se escogió esta como una de las áreas claves para el Modelo Cubano de Calidad a elaborar.

1.5 Ontologías

1.5.1 Tendencias de las Ontologías

La historia de la Ontología se remonta por los antiguos griegos, con Parménides como primera cabeza visible. Su historia es accidentada y ambigua, no existe un hilo conductor que permita guiar fácilmente. Sin embargo, se sabe que es distinta de la metafísica, que tiene un camino mucho más definido, sólido y claro. Dado el amplio mercado de opiniones, existe un sinnúmero de definiciones y caracterizaciones de las Ontologías, unas débiles, otras fuertes; que intentan abordar la cuestión con diferentes grados de éxito.

Con el auge en el mundo de las informáticas, surgen también las Ontologías en informática, que vienen de la filosofía de la representación de la realidad a través del conocimiento. Se originan del campo de la metafísica. Desde la aparición de la inteligencia artificial, estas Ontologías cobran una especial importancia en este campo, ya que se puede dotar de información a los sistemas; pero no una información cualquiera, sino una información que representa de forma conceptual el conocimiento.

En la actualidad, las Ontologías se están aplicando en áreas heterogéneas. Aunque quizás se les conoce más por su papel en el desarrollo de nuevos servicios en la web basados en la descripción del significado de los contenidos de las redes o portales de internet (web semántica), también se están utilizando para el desarrollo de mecanismos que faciliten la comunicación entre las personas y las máquinas por medio del lenguaje natural (procesamiento del lenguaje natural [PLN]). En el contexto sanitario se están manejando cada vez más, para diversas tareas, como la de recuperación de información, la de búsqueda de respuesta en fragmentos de texto que resuelven preguntas y la de extracción de información desde textos narrativos y no desde los campos estructurados de las bases de datos, o bien para la clasificación de documentos de forma automática.

1.5.2 Definiciones de Ontologías

En los últimos años el desarrollo de ontologías ha estado moviéndose del dominio de los laboratorios de Inteligencia Artificial a los escritorios de los expertos de un dominio dado. Las ontologías han llegado a ser comunes en el World-Wide Web. Las ontologías en la Web van desde grandes taxonomías que categorizan sitios Web (tales como en Yahoo!) a categorizaciones de productos para vender y sus características (tales como Amazon.com).

Según el diccionario de la Real Academia de la Lengua Española (DRAE) el término Ontología es la “Parte de la metafísica que trata del ser en general y de sus propiedades transcendentales”. En la rama de la Informática: “Una Ontología define los términos a utilizar para describir y representar un área de conocimiento. Las Ontologías son utilizadas por las personas, las bases de datos, y las aplicaciones que necesitan compartir un dominio de información (un dominio es simplemente un área de temática específica o un área de conocimiento, tales como medicina, fabricación de herramientas [...], etc.). Las Ontologías incluyen definiciones de conceptos básicos del dominio, y las relaciones entre ellos, que son útiles para los ordenadores [...]. Codifican el conocimiento de un dominio y también el conocimiento que extiende los dominios. En este sentido, hacen el conocimiento reutilizable”. (2004)

“Una ontología es una especificación explícita de una conceptualización” (Gruber, 1993). El término proviene de la filosofía, donde una ontología es un recuento sistemático de la existencia. En sistemas de Inteligencia Artificial (IA), lo que existe es lo que puede ser representado. Cuando el conocimiento de un dominio se representa mediante un formalismo declarativo, el conjunto de objetos que puede ser representado se llama universo del discurso. Estos conjuntos de objetos y las relaciones que se establecen entre ellos, son reflejados en un vocabulario con el cual representamos el conocimiento en un sistema basado en conocimiento”.

Otra definición de ontología es la presentada por Borst, la cual redefine la anterior dada por Gruber: “Una ontología es una especificación formal de una conceptualización compartida” (Tallarico). En esta definición el término formal se refiere a la necesidad de disponer de ontologías comprensibles por las máquinas. Esta definición enfatiza la necesidad de consenso en la conceptualización. Por otro lado el término

compartida se refiere al tipo de conocimiento contenido en las ontologías, esto es, conocimiento consensuado y no privado.

Una ontología es una representación formal de un conjunto de conceptos asociados a un dominio particular y de las relaciones existentes entre estos conceptos. Es formal en el sentido de que emplea un lenguaje para su representación. Esta representación puede ser procesada automáticamente. Una ontología está asociada a un dominio determinado cuyos objetos son referidos por los conceptos y relaciones de dicha ontología.

En Cuba se están dando los primeros pasos en cuanto a ontologías se refiere, un ejemplo de ello es la realización de este trabajo. El uso de ontologías en el desarrollo de sistemas contribuye en una mejora en la calidad del producto final, ya que ellas pueden ayudar a evitar problemas como:

- Inconsistencias entre ontologías implícitas en sistemas informáticos.
- Conflictos entre conceptos ontológicos e implementaciones.
- Conflictos entre ontologías de sentido común y conceptos básicos no incluidos en el software.

Traen objetivos como:

- Compartir la comprensión de estructuras de información entre personas o agentes de software. Esto debe revertir de forma positiva y casi necesaria en la extracción y recuperación de información, desde fuentes con contenidos conectados temáticamente.
- Permitir la reutilización del conocimiento perteneciente a un dominio en la elaboración de una nueva ontología o su utilización en la solución de un problema.
- Permite hacer explícitos los supuestos de un dominio. Esta aseveración puede conducir a conclusiones muy interesantes para la representación del conocimiento más allá de consideraciones técnicas, operativas e informáticas.
- Separa el conocimiento de un dominio del conocimiento que se puede tener de un proceso. El conocimiento que se representa en una ontología se puede aplicar en diferentes áreas de conocimiento independientemente del proceso que se modele.

- Hace posible analizar el conocimiento de un campo, por ejemplo en lo que se refiere al estudio de los términos y relaciones que lo configuran ya sea formalmente o no.

La construcción de ontologías lleva implícito que cada término y cada relación entre términos se definan formalmente. Los conceptos se describen explícitamente para entender su significado, mediante acuerdos ontológicos. Con ello un usuario que desee reutilizar una ontología desarrollada por otros, puede conseguir la información de todos los conceptos que soporta, su taxonomía y los axiomas. A diferencia de los tesauros, las ontologías poseen una mayor variedad de relaciones entre conceptos. Estas relaciones se crean a propósito del modelo conceptual existente en el dominio que se está formalizando.

1.5.3 Clasificación de ontologías

Guarino clasifica las ontologías de acuerdo con su dependencia y relación con una tarea específica desde un punto de vista (Guarino, 1997):

- Ontologías de Alto Nivel o Genéricas:

Describen conceptos más generales. En relación con los sistemas informáticos, estas ontologías describirían conceptos básicos. Por ejemplo: una teoría describiría partes y todos, y sus relaciones con la topología.

- Ontologías de Dominio:

Describen un vocabulario relacionado con un dominio genérico. Por ejemplo, podría ser una descripción de datos y entidades relacionados con la sensorización remota con un ambiente urbano.

- Ontologías de Tareas o de Técnicas básicas:

Describen 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:

Describen conceptos que dependen tanto de un dominio específico como de una tarea específica y, generalmente son una especialización de ambas. Se propone que este tipo de ontologías nazca a partir de una combinación de ontologías de niveles superiores.

Guarino marca otra distinción y diferencia, ontologías refinadas y no-refinadas, o lo que es lo mismo, off-line y on-line (Guarino, 1998):

➤ No refinada:

Tiene un número mínimo de axiomas y su objetivo es ser compartida por usuarios que concurren sobre una determinada visión del mundo. Tienen una mayor capacidad de ser compartidas y deberían ser utilizadas on-line para dar soporte en funcionalidad de sistemas de información.

➤ Refinada:

Precisa de un lenguaje de alta expresividad y tiene un gran número de axiomas. Deberían ser usadas off-line y solamente para referencia.

Van Heist clasifica las ontologías de acuerdo a la cantidad y tipo de estructura de la conceptualización (Using Explicit Ontologies in KBS Development, 1996):

➤ Ontologías terminológicas:

Especifican los términos que son usados para representar el conocimiento en el universo del discurso. Suelen ser usadas para unificar vocabulario en un campo determinado.

➤ Ontologías de información:

Especifican la estructura de almacenamiento de bases de datos. Ofrecen un marco para el almacenamiento estandarizado de información.

➤ Ontologías de modelado de conocimiento:

Especifican conceptualizaciones del conocimiento. Contienen una rica estructura interna y suelen estar ajustadas al uso particular del conocimiento que describen.

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

1.5.4 Conceptos claves en relación con las ontologías

En el tema de las ontologías existen una gran cantidad de conceptos y términos propios de ellas, aquí se listan los más importantes:

- Clase: Es un objeto que define una categoría. Describe conceptos en el dominio del discurso.
- Subclase: Es en sí misma una clase, pero que es hija de alguna otra clase.

- Clase jerárquica (class hierarchy): La compuesta por una colección de clases conectadas por relaciones "es un tipo de".
- Casos (instances): Ejemplos específicos pertenecientes a alguna clase, esto es, objetos de una clase.
- Roles o Propiedades (slots): Propiedades de cada concepto que describen varias características y atributos del concepto. Ayudan a definir las características de las clases.
- Facetas: Se utilizan para definir qué tipo de valor puede contener un slot particular, valores permitidos, número de valores, etc. También se denominan restricciones de roles.
- Valor: Describe una propiedad que se aplica a alguna clase o caso.
- Tipo: Define el tipo de valor (como cadena de caracteres, número, booleano, etc.)
- Cardinalidad: Define cuántos valores puede tener un slot individual (máximo y mínimo).
- Herencia: Es el proceso por el cual las subclases e instancias de alguna clase heredan propiedades y valores definidos más arriba en la jerarquía.
- Variable: Espacio vacío que puede llenarse preguntando a clases e instancias. Cada variable comienza con un signo de interrogación.
- Relación: Nuevo conocimiento que se obtiene por deducción, partiendo del conocimiento que se encuentra en la ontología. Las relaciones utilizan variables.

1.5.5 Aplicaciones y usos de las ontologías

En la actualidad existen diferentes campos en los que se están realizando proyectos en los que se aplican Ontologías como en:

- **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, las tendencias de los mismos 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 también la personalización de la información a mostrar al usuario según sus intereses.
- **Búsqueda de información en Internet:** 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 del conocimiento¹, ya que en él existen numerosos recursos, desde artículos de investigación² hasta bases de datos de hospitales, pasando por bases de datos públicas describiendo interacciones entre proteínas y genes³. Toda esta 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, consistente y coherente a los recursos digitales.
- **Web Semántica:** Es una Web extendida, dotada de mayor significado, en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Al dotar a la Web de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de una infraestructura común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla. Esta Web extendida y basada en el significado, se apoya en el desarrollo de ontologías preferiblemente desarrolladas mediante el lenguaje OWL.
- **Sistemas de información basados en ontologías:** Los sistemas de información (SI) son esencialmente artefactos de conocimiento que capturan y representan el conocimiento sobre ciertos dominios. Las ontologías están llegando a ser una herramienta fructífera en la

¹ NCBO: <http://www.bioontology.org/>

² PubMed: <http://www.ncbi.nlm.nih.gov/entrez/query.fcgi>

³ UniProt: <http://www.ebi.uniprot.org/index.shtml>

investigación y desarrollo de la disciplina de los SI. Esto ha llevado a la noción de SI basados en ontologías (SIBO), un concepto que, aunque en una fase preliminar de desarrollo, abre nuevas maneras de pensar sobre las ontologías y los SI en conjunción unas con otros, y cubre las dimensiones estructurales, las dimensiones temporales de los SI e involucra tanto a los desarrolladores como a los usuarios de los SI. Las ontologías y los SIBO están desarrollándose y aplicándose en una variedad de áreas de aplicación emergentes tales como modelización de empresas, diagnósticos, toma de decisión, planeamiento y adaptación, modelado de procesos y sistemas.

Las Ontologías se usan para favorecer la comunicación entre personas, organizaciones y aplicaciones, lograr la interoperabilidad entre sistemas informáticos, razonar automáticamente y para la ingeniería de software. 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.

Las Ontologías serán imprescindibles en el desarrollo de la Web Semántica y en los futuros sistemas de gestión empresarial, porque permitirán que las aplicaciones estén de acuerdo en los términos que usan cuando se comunican. Mediante ellas, será mucho más fácil recuperar información relacionada temáticamente, aún cuando no existan enlaces directos entre las páginas web. También permiten que los sistemas interoperen. Dos sistemas son interoperables si pueden trabajar conjuntamente de una forma automática, sin esfuerzo por parte del usuario. Resultan muy útiles para facilitar 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. Una de las aplicaciones más importantes del razonamiento automático es la validación de datos.

En la Ingeniería del Software, las Ontologías ayudan a la especificación de los sistemas de software. Como la falta de un entendimiento común conduce a dificultades en identificar los requisitos y especificaciones del sistema que se busca desarrollar, las Ontologías facilitan el acuerdo entre desarrolladores y usuarios.

1.6 Lenguajes

Las Ontologías para ser expresadas requieren de un lenguaje lógico y formal. En la inteligencia artificial se han desarrollado numerosos lenguajes para este fin, algunos basados en la lógica de predicados, como KIF y Cycl que ofrecen poderosas primitivas de modelado, y otros basados en frames (taxonomías de clases y atributos), que tienen un mayor poder expresivo, pero menor poder de inferencia; e incluso existen lenguajes orientados al razonamiento. Todos estos lenguajes han servido para desarrollar otros lenguajes aplicables a la Web. En un lenguaje de Ontologías se pretenderá un alto grado de expresividad y uso.

Dentro de los principales lenguajes de Ontologías para explotar las características de la web se destacan los siguientes. Todos ellos están basados en la capacidad sintáctica de XML (Lenguaje de Etiquetado Extensible).

SHOE: Simple HTML Ontology Extensions. Fue el primer lenguaje de etiquetado para diseñar Ontologías en la Web. Nació antes de que se ideara la Web Semántica. Las Ontologías y las etiquetas se incrustaban en archivos HTML. Este lenguaje permite definir clases y reglas de inferencia, pero no negaciones o disyunciones. A su albur se desarrollaron muchos editores, buscadores, API (Interfaz de Programación de Aplicaciones), etc.; el proyecto fue abandonado a medida que se desarrollaron OIL y DAM; aunque también existe una socialización de este lenguaje en XML.

OML: Fue parcialmente desarrollado con base en SHOE por la Universidad de Washington, y fue considerado una serialización de SHOE, puesto que comparten múltiples características. Su modelo de conocimiento son los grafos conceptuales.

XOL: Lenguaje para intercambio de Ontologías basado en XML, fue desarrollado tomando como base OML y Ontolingua, por parte de investigadores del área bioinformática en los Estados Unidos.

OIL: Ontology Inference Layer, derivado en parte de SHOE, fue impulsado también por el proyecto de la Unión Europea On-To-Knowledge. Utiliza la sintaxis del lenguaje XML y está definido como una extensión de RDFS. Se basa tanto en la lógica descriptiva (declaración de axiomas) como en los sistemas basados en frames (taxonomías de clases y atributos). OIL posee varias capas de sub-lenguajes, entre ellas

destaca la capa base que es RDFS, a cada una de ellas añade alguna funcionalidad y mayor complejidad. La principal carencia de este lenguaje es la falta de expresividad para declarar axiomas.

RDF y RDF (S): XML aporta su poder sintáctico para soportar el desarrollo de RDF (Marco para la Descripción de Recursos) y RDF(S) (RDF esquema), como una herramienta orientada a compartir conocimiento en la WEB.

El modelo de datos de RDF consiste en tres tipos de objetos: Recursos, los cuales son expresiones RDF y están descritos mediante (Identificadores Uniformes del Recurso) URI's; Propiedades, que describen las características, relaciones, y atributos o aspectos específicos de los recursos; y las Instrucciones que asignan un valor a una propiedad en un recurso específico. El modelo de datos RDF no hace ninguna asunción sobre la estructura de un documento que contiene información RDF, permitiendo que las instrucciones puedan aparecer en cualquier orden dentro de una ontología. Tampoco provee primitivas de modelado para definir las relaciones entre propiedades y recursos. Esta limitación es solucionada mediante el lenguaje para describir vocabulario, conocido como RDF Schema, el cual se extiende con primitivas basadas en el paradigma de KR de Frames.

La combinación RDF y RDFS, es conocida como RDF(S). EL RDFS provee los mecanismos para establecer las relaciones entre propiedades y recursos. Las primitivas de RDFS son agrupadas en: clases, propiedades, contenedores de clases y propiedades, colecciones y propiedades de utilidad. RDF(S) provee las primitivas básicas, necesarias para modelar ontologías, existe un balance adecuado entre sus capacidades de expresividad y razonamiento. Su desarrollo se realizó buscando un núcleo estable que pudiera ser fácilmente extendido. RDFS es usado ampliamente por diferentes herramientas como Protege, Mozilla y Amaya.

OWL: Web Ontology Language o Lenguaje de Ontologías para la Web, es un lenguaje de etiquetado semántico para publicar y compartir Ontologías en la Web. Se trata de una recomendación del W3C, y puede usarse para representar Ontologías de forma explícita, es decir, permite definir el significado de términos en vocabularios y las relaciones entre estos términos. En realidad, OWL es una extensión del lenguaje RDF y emplea las tripletas de RDF, aunque es un lenguaje con más poder expresivo que este. Se trata de un lenguaje diseñado para usarse cuando la información contenida en los documentos

necesita ser procesada por programas o aplicaciones, en oposición a situaciones donde el contenido solamente necesita ser presentado a los seres humanos. Al igual que OIL, OWL se estructura en capas que difieren en la complejidad y puede ser adaptado a las necesidades de cada usuario, al nivel de expresividad que se precise y a los distintos tipos de aplicaciones existentes (motores de búsqueda, agentes, etc.).

Una ontología en OWL, inicia con la declaración del nodo raíz de RDF. En este nodo podemos incluir el espacio de nombres (namespaces) para RDF, RDF(S) y OWL. Sin son utilizados tipos de datos pertenecientes a XML Schema, puede ser incluido su espacio de nombres. El orden en el cual aparecen las definiciones no es relevante, lo usual es definir la cabecera y posteriormente sus términos. En su cabecera se puede incluir: la documentación de la ontología, las ontologías que son importadas, la versión con la información que permita controlar su compatibilidad con las demás versiones. Los conceptos son conocidos como clases, cada uno de los cuales puede contener diversos elementos dependiendo de la capa del lenguaje que se esté utilizando. Existen 3 sublenguajes de OWL, los cuales van creciendo respecto al nivel de expresión:

OWL Lite: Útil para la creación de jerarquías y restricciones simples, sólo permite valores de cardinalidad 0 y 1, pierde en expresividad.

OWL DL (Description Logic): es el lenguaje más sencillo he indicado para los usuarios que requieren el máximo de expresividad y decibilidad (todos los cálculos acaban en un tiempo finito). Una clase puede ser a la vez subclase de muchas clases, no puede ser una instancia de otra clase.

OWL Full: Máximo nivel de expresión y la libertad sintáctica de RDF. Permite expresiones de segundo orden, pero sin decibilidad. Por ejemplo, una clase puede ser tratada simultáneamente como una colección de individuos y como un individuo por sí mismo. Es el más completo por lo que se necesita mucho poder computacional para poder hacer inferencias es por eso que se dice que no tiene garantía computacional.

A continuación se muestra una tabla comparativa entre los lenguajes ontológicos más usados en la web para la representación del conocimiento. Como se aprecia OWL es uno de los más completos al igual que

el DAML+OIL.

Tabla 1: Comparación de lenguajes ontológicos.

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

De los diferentes lenguajes que existen para representar el conocimiento se seleccionó OWL debido a que es el más completo de los lenguajes utilizados para la representación del conocimiento en la web. Es una evolución de los lenguajes DAML+OIL y el RDF-S. Es abierto y extensible y recomendado por W3C. Además por ser una evolución de los anteriores incorpora una serie de primitivas para enriquecer la expresividad de los elementos y sus relaciones.

La utilización de este lenguaje permite publicar y compartir la ontología en la web, hacer tan explícita la misma como se quiera, definiendo el significado de los términos de peso en el dominio como son las clases *Artefactos*, *Documentos*, *Herramientas*, *Roles* y todas sus clases hijas respectivamente y estableciendo las relaciones entre estos términos demostrando mayor poder expresivo. Estas relaciones se refieren a la jerarquía de clases lograda después de definir las clases padres e hijas, y a las propiedades que tienen las mismas. Este lenguaje es utilizado además porque permite procesar toda esta información del dominio que anteriormente se encontraba en documentos. Otra ventaja que proporciona

es que es adaptable a las necesidades del usuario y al nivel de expresividad que se precise, lo que permite que se utilice el OWL DL siendo este el más sencillo he indicado para lograr el máximo de expresividad y decibilidad (todos los cálculos acaban en un tiempo finito). Otra de las facilidades que brinda es que una clase puede ser a la vez subclase de muchas clases pero no puede ser instancia de otra clase, tal es el caso de la clase *Acciones_Correctivas* la cual en el *Proceso_Base_Requisitos* es una subclase de *Tickets* pero la instancia *Acciones_Correctivas* es de tipo *Acciones_Correctivas* solamente.

1.7 Herramientas para construir ontologías

De acuerdo al estudio realizado sobre el estado del arte se han podido identificar una serie de herramientas indispensables para el trabajo con ontologías. Estas herramientas se organizan en las categorías siguientes: Desarrollo, Evaluación, Combinación e integración, Herramientas de anotación, Almacenamiento y consulta y Aprendizaje. Pero se tratarán solo las de desarrollo ya que son las que interesan para dar cumplimiento al trabajo. Este grupo incluye herramientas que pueden ser usadas para construir una ontología o reutilizar una ya existente.

Herramientas de desarrollo de ontologías

Ontolingua

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

OntoStudio

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

WebOnto

Fue diseñada para facilitar la navegación, creación y edición cooperativa de ontologías representadas sin sufrir problemas de interfaz. Las principales características son: la gestión gráfica de ontologías, la

generación automática de instancias a partir de definiciones de clases y la inspección de elementos, la herencia de propiedades y el chequeo de consistencia.

OntoEdit

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

Protégé

Es un editor de Ontologías escrito en Java, gratuito y de código abierto. Tras él existe una gran comunidad de desarrolladores y de usuarios universitarios, empresariales y gubernamentales. Actualmente permite trabajar con RDFS y dispone de una extensión para OWL. Su sencillez y su buena documentación lo hacen ideales para los principiantes en Ontologías.

Este editor incluye además SPARQL (Protocol and RDF Query Language). Se trata de un lenguaje normalizado por el RDF Data Access Working Group (DAWG) del Word Wide Web Consortium (W3C). Es una tecnología clave en el desarrollo de la Web Semántica que se constituyó como recomendación oficial del W3C el 15 de Enero de 2008. Ofrece a los desarrolladores y usuarios finales un camino para presentar y utilizar los resultados de búsquedas a través de una gran variedad de datos, siendo útil para recuperar y organizar la información. También proporciona un camino de integración sobre recursos diferentes. Permite extraer información en diversas formas, extraer subgrafos RDF y construir nuevos grafos RDF basados en la información de los grafos consultados.

La arquitectura de Protégé se divide en dos partes: modelo y vista de usuario. La primera parte define un metamodelo compuesto por clases, atributos, restricciones e instancias. Las clases son los conceptos del dominio del discurso (por ejemplo *Artefactos*), los atributos describen propiedades de las clases (por ejemplo *tieneDescripción*), las instancias proporcionan particularizaciones de las clases (por ejemplo

Arquitectura_de_Información) y las restricciones proporcionan información adicional para poder realizar las relaciones entre instancias. El metamodelo, en sí, es una ontología que representan clases, atributos, etc. La parte de vista de usuario es el espacio en la herramienta de modelado que permite la definición de las clases, atributos, restricciones e instancias de un dominio en particular. Además de las anteriores características de la vista de usuario, Protégé tiene la capacidad de generar interfaces gráficas, a partir de la ontología de dominio, que permiten al usuario instanciar la ontología en casos particulares (por ejemplo la de este trabajo que se refiere a la Ingeniería de Requisitos).

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. Además incorpora el manejo de plug-ins que permiten a los programadores externos ampliar las funcionalidades de la herramienta. Igualmente, dada la arquitectura de implementación de Protégé, existen diversas formas para acceder a ésta por medio de lenguajes de programación.

De las herramientas estudiadas se trabajará con Protégé por ser un editor de ontologías escrito en Java, gratuito y de código abierto. Dispone de una extensión para OWL. Es sencillo y tiene buena documentación lo que lo hace fácil para principiantes en Ontologías. Posee diferentes medios para exportar la ontología tales como: XML, RDFS, OWL y bases de datos relacionales, entre otros. Además incorpora el manejo de plug-ins que les permiten a los programadores externos ampliar las funcionalidades de la herramienta.

1.8 Metodologías para el Desarrollo de Ontologías

El proceso de construir una ontología es algo más que una actividad de ingeniería. Cada grupo de desarrollo usualmente sigue sus propios principios, criterios de diseño y fases de desarrollo. La falta de métodos y normas generalizados para la definición de una ontología tanto al interior de los equipos de trabajo, como entre ellos, obstaculiza el desarrollo de ontologías compartidas y consensuadas, la extensión de ontologías desarrolladas por otros grupos, y su rehusó en otras ontologías y aplicaciones. Las metodologías proporcionan un conjunto de directrices que indican cómo hay que llevar a cabo las actividades identificadas en el proceso de desarrollo, qué técnicas son las más apropiadas en cada actividad y qué produce cada una de ellas. Entre las metodologías existentes encontramos:

- La metodología de Uschold y King se basa en la experiencia de construir el Enterprise Ontology, el cual incluye un conjunto de ontologías para modelado empresarial. Ellos proponen los siguientes pasos: (1) Identificar el propósito y el alcance de la ontología; (2) Construir la ontología capturando conocimiento, codificando conocimiento e integrando tal conocimiento con ontologías existentes; (3) Evaluar la ontología; (4) Documentación; y (5) Lineamientos para cada fase.
- La metodología de Gruninger y Fox: Esencialmente requiere construir un modelo lógico del conocimiento que será especificado en la ontología, este modelo no se construye directamente. Primero, son descritas informalmente las especificaciones que tienen que ser encontradas por la ontología identificando un conjunto de asuntos competentes, y su descripción es entonces formalizada en un lenguaje basado en el cálculo de predicados de primer orden. Las preguntas de competencia son las bases para una caracterización rigurosa del conocimiento que tiene que cubrir la ontología, estas especifican el problema y lo que constituye una buena solución al problema. Con un mecanismo de composición y descomposición, las preguntas de competencia y sus respuestas pueden ser usadas para responder preguntas de competencia más complejas en otras ontologías, permitiendo la integración de las mismas.
- El marco de trabajo METHONTOLOGY permite la construcción de ontologías en el nivel del conocimiento. Incluye: la identificación del proceso de desarrollo de la ontología, un ciclo de vida propuesto y la metodología como tal. El proceso de desarrollo de la ontología identifica las tareas que deben realizarse cuando se construye una ontología (planeación, control, control de calidad, especificación, adquisición de conocimientos, conceptualización, integración, formalización, implementación, evaluación, mantenimiento, documentación y administración de la configuración). El ciclo de vida basado en prototipos evolutivos identifica las etapas que atraviesa la ontología durante su tiempo de vida. Finalmente, la metodología en sí misma, especifica los pasos que se deben seguir para desarrollar cada actividad, las técnicas usadas, los productos que se producirán y como serán estos evaluados. La fase principal en el proceso de desarrollo de una ontología usando el enfoque de METHONTOLOGY es la fase de conceptualización.
- Otra metodología de este tipo (orientada a proceso) es On-To-Knowledge (OTK), la cual contiene identificación de metas y objetivos, que son logrados con herramientas de soporte para el manejo de conocimiento. Las distintas fases que promueven son: estudio de factibilidad, fase inicial,

refinamiento, inferencia, evaluación, y aplicación-evolución.

- La metodología Kactus construye la Ontología sobre una base de conocimiento por medio de un proceso de abstracción. 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.
- La metodología Sensus es un enfoque top-down para derivar Ontologías específicas del dominio a partir de grandes Ontologías. En esta metodología se identifican un conjunto de términos semilla 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 profundo de las diversas metodologías para la construcción de ontologías y saber que todas son ideales para el diseño de la ontología; se decidió utilizar una guía de pasos específicos que existen para el desarrollo de Ontologías (ver apartado 1.8.1), la cual aparece en varias bibliografías relacionadas al diseño y construcción de Ontologías. Estos pasos son más específicos, concretos y entendibles que los pasos que las metodologías proponen, permitiendo la reducción de tiempo y aumentando la rapidez en todo el proceso de diseño y especificación de la Ontología. Son como la síntesis, recopilación o una vinculación de los mejores pasos de las distintas metodologías existentes. Sin dejar de mencionar que aún no hay una metodología estandarizada o aceptada como principal, por la mayoría de los especialistas en el desarrollo de ontologías.

1.8.1 Pasos específicos para el desarrollo de ontologías

A continuación se citan los pasos específicos para la creación de una Ontología (Natalya F. Noy, 2005):

- 1. Definir el dominio y alcance o ámbito de la Ontología:** Se refiere a definir el dominio que cubrirá la Ontología, el uso y la aplicación final; definir a qué tipo de preguntas responderá, recolectar la información necesaria, entre otros. La definición del dominio y el alcance de la misma, sería respondiendo preguntas del tipo ¿Cuál es el dominio que la Ontología debe cubrir?, ¿Para qué se

va a usar la Ontología?, ¿A qué preguntas debe dar respuesta la Ontología?, ¿Quién va a usar y a mantener la Ontología? Una de las formas de determinar el ámbito de la Ontología es preparar una lista de preguntas no exhaustiva, que la base de conocimientos basada en la Ontología debe ser capaz de responder, lista que recibe el nombre de 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 Ontolingua⁴ o en DAML ontology library⁵.
- 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. Hay diferentes aproximaciones: ascendente, descendente y mixto.
- 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 el tipo de valor, los valores permitidos, el número de valores (cardinalidad), así como otras características de los valores que la propiedad puede tener.
- 7. Creación de instancias:** Es el último paso de este proceso. Definir una instancia individual de una clase requiere: a) elegir una clase; b) crear una instancia individual de esa clase, y c) rellenar las propiedades con valores.

⁴ <http://www.ksl.stanford.edu/software/ontolingua/>

⁵ <http://www.daml.org/ontologies/>

1.9 Aplicaciones o soluciones a reutilizar

Ontología para la Calidad de los Servicios Web.

En dicha ontología se describen los modelos en base a los cuales se analiza la Calidad de Software de Servicios Web. Proporciona un marco general para la descripción y utilización de cualquier modelo de calidad para evaluar Servicios Web y se realiza mediante el lenguaje OWL-Services.

Ontología para la Medición del Software.

Se considera una ontología semi-formal pues está basada en (diagramas de clases UML, tablas y texto) REFSENO⁶. A través de esta ontología se establecen y clarifican los conceptos y relaciones relevantes relacionadas con la medición del software, a partir de los cuales se puede definir la información necesaria que hay que recoger y establecer para llevar a cabo un proceso de medición adecuado y efectivo. Además, se proporciona una forma de comunicación en una empresa en el contexto de sus procesos de medición, basada en el uso de una terminología común, lo que facilita no sólo el entendimiento entre miembros del equipo responsables de realizar dichas actividades de medición, sino también la posibilidad de registrar los resultados de dicho proceso de una forma consistente e integrada.

Ontología basada en OWL para la Representación de CMMI.

La ontología incluye 309 clases y 4 propiedades de clases.

Estas Ontologías están definidas para dominios de Calidad, Ingeniería de software, entre otros, que en realidad para esta investigación no son de gran importancia. No se encontraron Ontologías relacionadas con el tema por lo que es totalmente nuevo el dominio que se desea abarcar en la Ontología de este trabajo. No obstante es bueno destacar que a pesar de no reutilizarse ninguna ontología existente se revisaron las bibliotecas de ontologías reusables en la Web, tales como: Ontolingua⁷, la biblioteca de ontologías DAML⁸, ontologías comerciales como UNSPSC⁹, RosettaNet¹⁰ y DMOZ¹¹. Estas ontologías

⁶ REFSENO es un formalismo diseñado para la representación de ontologías de ingeniería del software mediante tablas, texto y opcionalmente diagramas; por lo que las ontologías obtenidas son semi-formales.

⁷ <http://www.ksl.stanford.edu/software/ontolingua>

⁸ <http://www.daml.org/ontologies>

⁹ <http://www.unspsc.org>

¹⁰ <http://www.rosettanet.org>

¹¹ <http://www.dmoz.org>

dieron un gran aporte a la investigación en lo que respecta a la visión del modelado de dominios, así como el diseño.

1.10 Conclusiones del Capítulo

En este capítulo se expusieron los principales conceptos relacionados con esta investigación. Fueron estudiados los aspectos relacionados con las Ontologías y la gestión del conocimiento llegando a la conclusión de que no existe herramienta tan eficaz para modelar, compartir, reutilizar conocimiento y contribuir a la toma de decisiones, como las Ontologías. De los tipos de Ontologías que se estudiaron, se optó por la Ontología de Dominio por sus características propicias y perfectamente aplicables al dominio donde se desea crear la Ontología, en este caso es, sobre el de la Ingeniería de Requisitos. Se mostraron las principales herramientas y metodologías utilizadas para construir y representar Ontologías, donde se ha decidido utilizar:

- La herramienta Protégé por ser un editor de Ontologías escrito en Java, gratuito y de código abierto. Permite construir, modificar y consultar ontologías. Es sencillo y tiene buena documentación lo que lo hace ideal para principiantes en Ontologías.
- Existen diversas metodologías para guiar el proceso de desarrollo de ontologías, pero no existe una estandarizada por los especialistas, 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: Implementación de la ontología

2.1 Introducción al capítulo

En este capítulo se tratarán aspectos relacionados con la implementación de la ontología partiendo de una serie de mapas conceptuales. Se abordará además en las formas de representación de las mismas. Se mostrarán las clases, las propiedades y las instancias.

2.2 Modelo conceptual

Se han publicado diferentes criterios sobre el concepto de los mapas conceptuales, uno de ellos precisamente definido por Novak, su creador, publicado en su texto “Aprendiendo a aprender”, define el mapa o modelo conceptual como una técnica que representa simultáneamente, una estrategia de aprendizaje, un método para captar lo más significativo de un tema y un recurso esquemático para representar un conjunto de significados conceptuales, incluidos en una estructura de proposiciones.

Los mapas conceptuales proporcionan un resumen esquemático de lo aprendido, ordenado de una manera jerárquica. El conocimiento está organizado y representado en todos los niveles de abstracción, situando los más generales e inclusivos en la parte superior o izquierda, y los más específicos y menos inclusivos en la parte inferior o derecha. Las relaciones entre ellos se explicitan mediante líneas que unen sus cajas respectivas. Las líneas, a su vez, tienen palabras asociadas que describen cuál es la naturaleza de la relación que liga los conceptos.

Se parte de una serie de mapas conceptuales a partir de los cuales se desarrollará la ontología. Estos mapas fueron separados para su mejor comprensión y entendimiento favorecidos por las facilidades que nos brinda la herramienta CompendiumLD. Posteriormente se extraerán de estos las clases a desarrollar en la herramienta Protégé. El primero y más general se refiere al proceso de Ingeniería de Requisitos que comprende a su vez el Proceso Base de Requisitos y el Proceso de Administración de Requisitos siendo estos los principales procesos a tratar. Figura 1

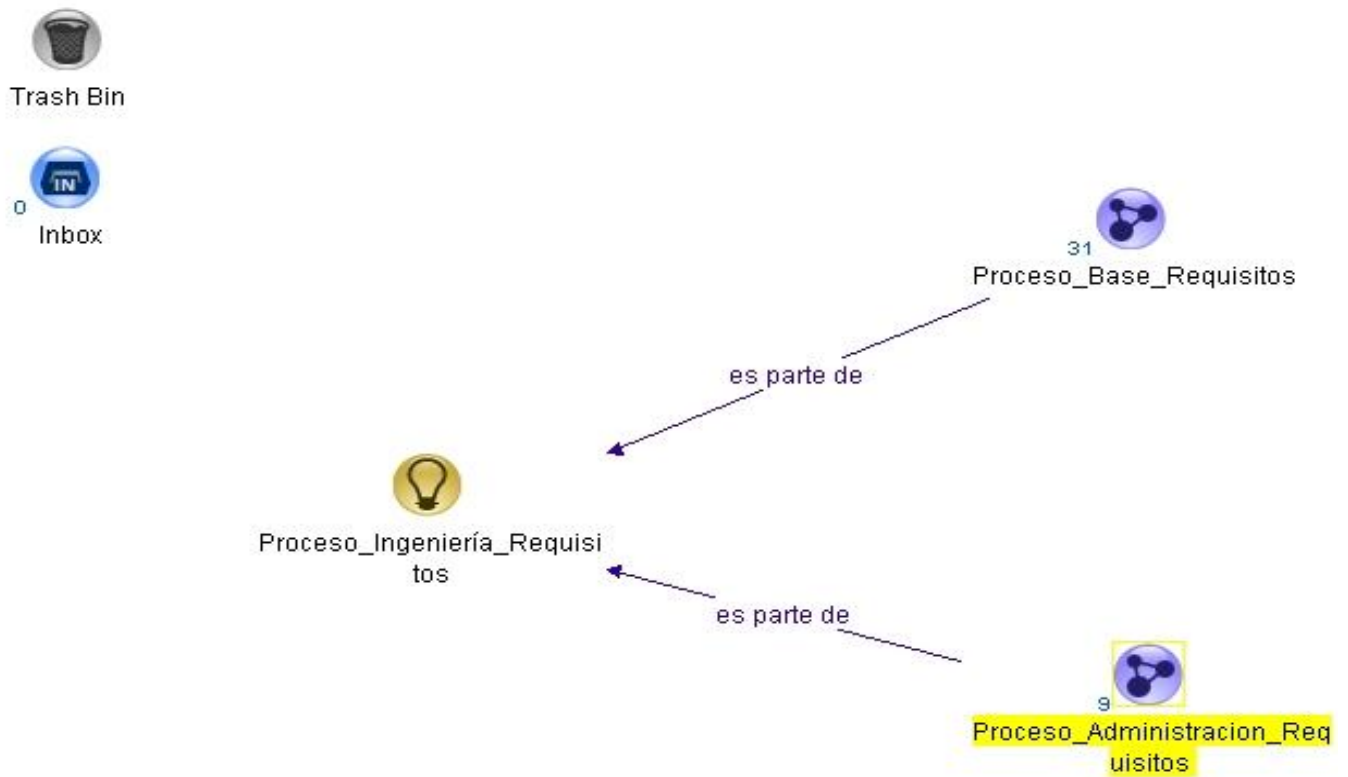


Figura 1: Proceso de Ingeniería de Requisitos.

Estos mapas conceptuales van desglosando todo el conocimiento implícito en el Proceso de Ingeniería de Requisitos. El resto de ellos se encuentran en los anexos. Ya el conocimiento está en estos modelos solo resta llevarlo al Protégé.

2.3 Formas de Representación de las Ontologías

El conocimiento inferido se puede representar de diferentes maneras, como figuras, diagramas, entre otros; pero la forma más precisa para expresar una Ontología, y por ende la más popularizada en el mundo informático es el uso de ficheros XML+ RDF + OWL. Aquí se muestran algunos ejemplos de representaciones ontológicas.

- Mediante un Fichero XML a través del lenguaje OWL en este caso. A continuación, en la Figura 2, se muestra un fragmento de la Ontología representada mediante el lenguaje OWL en un fichero.

```

"http://www.semanticweb.org/ontologies/2011/1/Ontology1298468383026.c
...
acepiRealizadoPor --><< <owl:ObjectProperty
rdf:about="#acepiRealizadoPor">< <rdfs:domain
rdf:resource="#Alcanzar_Compromiso_Entre_Partес_Involucradas"/><
<rdfs:range rdf:resource="#RolesAnalista"/>< <rdfs:range
rdf:resource="#RolesCliente"/>< <rdfs:range
rdf:resource="#RolesJefe_Proyecto"/>< </owl:ObjectProperty>< <<
<!--
...
Descripci#243;nSistema --><< <owl:DatatypeProperty
rdf:about="#Descripci#243;nSistema">< <rdfs:domain
rdf:resource="#Vista_del_sistema"/>< <rdfs:range
rdf:resource="#xsd:string"/>< </owl:DatatypeProperty>< <<<
<!--
...
Acciones_Correctivas --><< <owl:Class
rdf:about="#Acciones_Correctivas">< <rdfs:subClassof
rdf:resource="#Proceso_Base_Requisitos"/>< <rdfs:subClassof
rdf:resource="#Proceso_de_Administraci#243;n_Requisitos"/><
</owl:Class>< <<< <!--
...
Aceptada --><< <owl:Thing rdf:about="#Aceptada">< <rdf:type
rdf:resource="#Estado"/>< </owl:Thing>< <<< <!--

```

Figura 2: Ontología representada mediante el lenguaje OWL en un fichero.

➤ Mediante un Gráfico

Otra forma de representar las Ontologías es mediante un gráfico como se muestra en la Figura 3, donde está representado un ejemplo de clases y subclases de la Ontología propuesta como solución al problema planteado en la investigación.

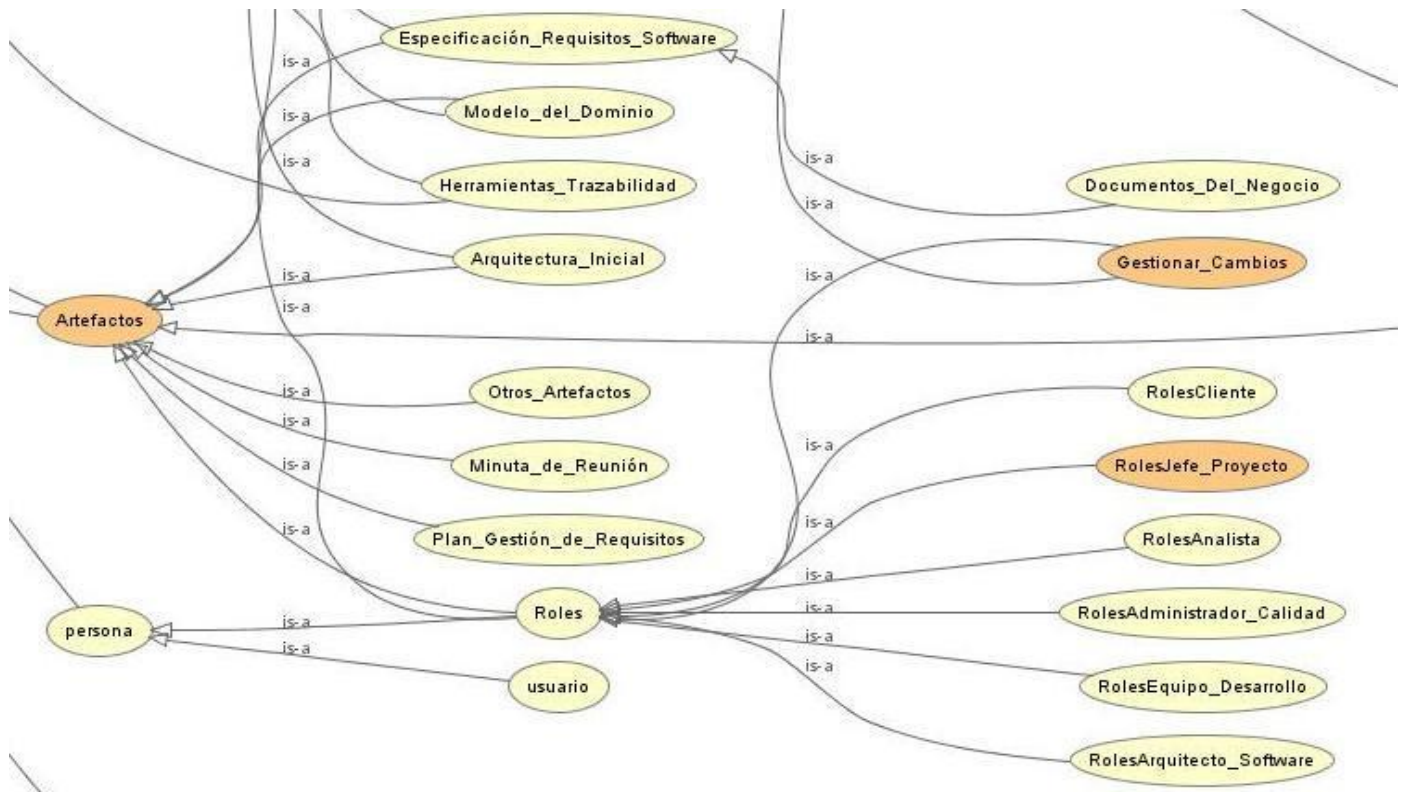


Figura 3: Algunas clases y subclases de la Ontología propuesta para solución.

En este trabajo se escogió la representación de la Ontología a través de un gráfico porque realizar la representación mediante modelos y diagramas es hacerlo de manera precisa, entendible y económica. También permite 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 código al respecto.

2.4 Modelo representativo de la ontología

Este es el modelo ontológico (Figura 4) realizado en la Herramienta Protégé versión 4.0 a partir de los modelos conceptuales anteriormente expuestos. No podemos olvidar que la ontología es un modelo de la realidad y esta debe demostrarlo así. El proceso de diseño y construcción de la ontología es iterativo y continuará a través del ciclo de vida de la misma.

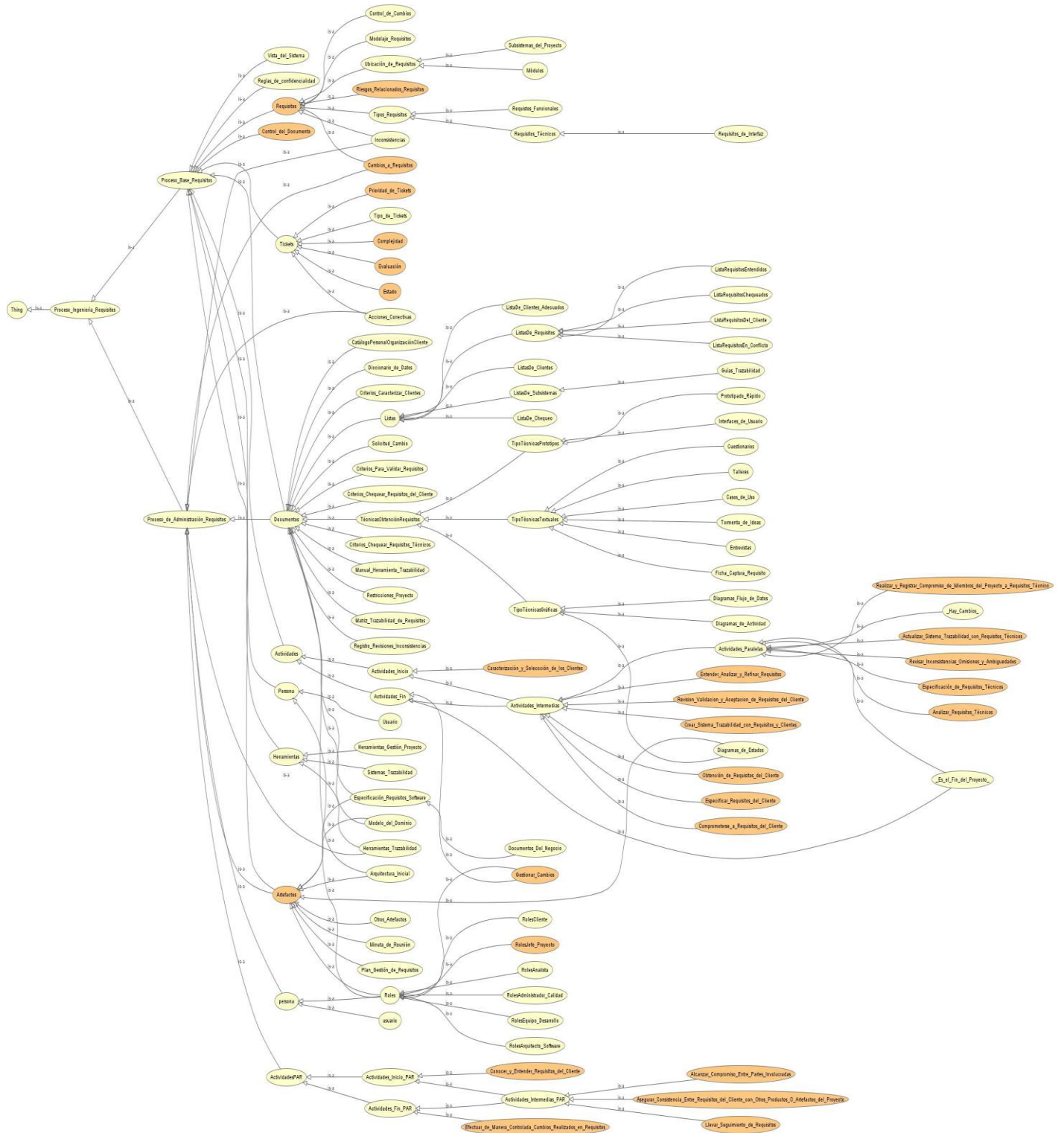


Figura 4: Modelo de la ontología.

2.5 Desarrollo de la ontología

En consecuencia con la metodología utilizada se comienza definiendo el dominio y el alcance de la ontología, para ello se ha definido el dominio de la Ingeniería de Requisitos y será usada para brindar información sobre la Administración de Requisitos y el Proceso Base de Ingeniería de Requisitos del Modelo Cubano de Calidad para el Desarrollo de Aplicaciones Informáticas en la Industria Cubana de Software. También provee quienes son los roles de esos procesos, las actividades a desarrollar y los diferentes documentos y artefactos que se generan. 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. Esta ontología será usada por todo el personal de calidad del software y la mantendrán los ingenieros del conocimiento de cada aplicación.

Preguntas de competencia Una forma de determinar el alcance de la ontología es elaborar una serie de preguntas 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 preguntas de competencia, son 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 rol pertenece la solicitud de cambios?

¿Qué herramienta de trazabilidad se puede utilizar para crear sistemas 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 entrar los requisitos del cliente y los requisitos 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?

¿Con la evaluación del impacto de los cambios qué tipo de actividad se ejecuta?

¿Quiénes son los responsables del registro de peticiones de cambios?

Acorde a estas preguntas se pueden elaborar muchas más y así la ontología incluiría 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.

Es totalmente nuevo el dominio que se desea abarcar en la Ontología de este trabajo, por ende no se encontraron Ontologías relacionadas al mismo. Las encontradas están definidas para dominios de calidad e Ingeniería de Software que no son de importancia para esta investigación. Las métricas de idoneidad no pudieron ser utilizadas al no existir ontologías a reutilizar.

Es útil escribir una lista con los términos más relevantes que tratará la ontología (Tabla 2) para dar explicación a un usuario, sin preocuparse del recubrimiento entre los conceptos que representan, relaciones entre los términos o cualquier propiedad que los conceptos puedan tener, o si son clases o propiedades. En estos momentos no es tan importante tener la lista integral de todos los términos ni las relaciones o propiedades que presenten, eso se determinará en pasos posteriores a esta etapa de diseño. ¿Cuáles son los términos a tratar?

Tabla 2: Lista de términos relevantes.

	Términos		Términos		Términos
1	Proceso Base de Requisitos	10	Jefe de Proyecto	19	Especificación de Requisitos de Software
2	Proceso de Administración de Requisitos	11	Arquitecto	20	Control de cambios
3	Roles	12	Artefactos	21	Sistema de Trazabilidad
4	Actividades	13	Actividades de Inicio	22	Herramienta de Trazabilidad
5	Cliente	14	Actividades de Fin	23	Listas
6	Analista	15	Actividades Intermedias	24	Cambios
7	Catálogo	16	Restricciones	25	Técnicas
8	Proceso de Ingeniería de Requisitos	17	Documentos	26	Requisitos
9	Reglas de confidencialidad	18	Vista del sistema	27	Tickets
28	Control del documento	29	Herramientas	30	Personas
31	Requisitos técnicos	32	Requisitos funcionales	33	Manual de herramientas

A continuación se comienzan a definir las clases. Estas pueden ser de dos tipos: clases abstractas y clases concretas. Las clases concretas pueden tener instancias directas y las clases abstractas no incluyen instancias. Del mapa conceptual se seleccionan los términos que describen objetos que tienen

existencia independiente, los cuales serán las clases de la ontología, está compuesta por 21 clases genéricas de las cuales 4 son concretas y el resto abstractas. Ver tabla 3

Tabla 3: Clases de la Ontología.

	Clases
1	Proceso_Ingeniería_Requisitos
2	Proceso_Base_Requisitos
3	Actividades
4	Artefactos
5	Control_del_Documento
6	Documentos
7	Herramientas
8	Persona
9	Reglas_de_Confidencialidad
10	Requisitos
11	Tickets
12	Vista_del_Sistema
13	Proceso_de_Administración_Requisitos
14	Acciones_Correctivas
15	Actividades_PAR
16	Artefactos
17	Cambios_a_Requisitos
18	Documentos
19	Herramientas_Trazabilidad
20	Inconsistencias
21	Personas

Se organizan las clases en una taxonomía¹² jerárquica¹³, se determina si una instancia de una clase es instancia de alguna otra clase. Las clases se diseñaron basándose en el criterio top-down ya que se comenzó con la definición de los conceptos más generales del dominio hasta los más específicos. La taxonomía jerárquica está compuesta por 221 clases.

Protégé en cualquiera de sus secciones brinda 3 marcos de trabajo. Las clases se crearon como se muestra a continuación: después de tener abierta la herramienta se selecciona la pestaña *Classes* y se marca la clase *thing* que viene por defecto ya que todas las clases heredan de ella y para que se activen las funcionalidades que nos brinda el software. En el primer marco de trabajo se encuentra en su parte superior 3 botones que permiten adicionar subclases, crear clases hermanas y eliminar clases

¹² Ciencia que trata de los principios, métodos y fines de la clasificación.

¹³ Relacionado con la organización por grados de importancia.

respectivamente. Para crear una clase hija por ejemplo *Proceso_Ingeniería_Requisitos*, después de tener marcada la clase *thing* se selecciona el botón *Add subclass*, se muestra un cuadro de diálogo para entrar el nombre de dicha clase y al aceptar queda creada la misma (Figura 5). Para crear clases hermanas por ejemplo *Proceso_Base_Requisitos* se marca la clase ya definida *Proceso_de_Administracion*, se va al botón *Add sibling class* el q muestra un cuadro de diálogo para entrar el nombre de la nueva clase que sería *Proceso_Base_Requisitos* y después de aceptar quedan creadas por fin las clases hermanas (Figura 6). Para eliminar la clase *Proceso_Base_Requisitos* basta con detenerse encima de ella y marcar el botón *Delete selected classes* el cual muestra una pantalla para informar que se borrará la clase y todo lo relacionado con ella (Figura 7).

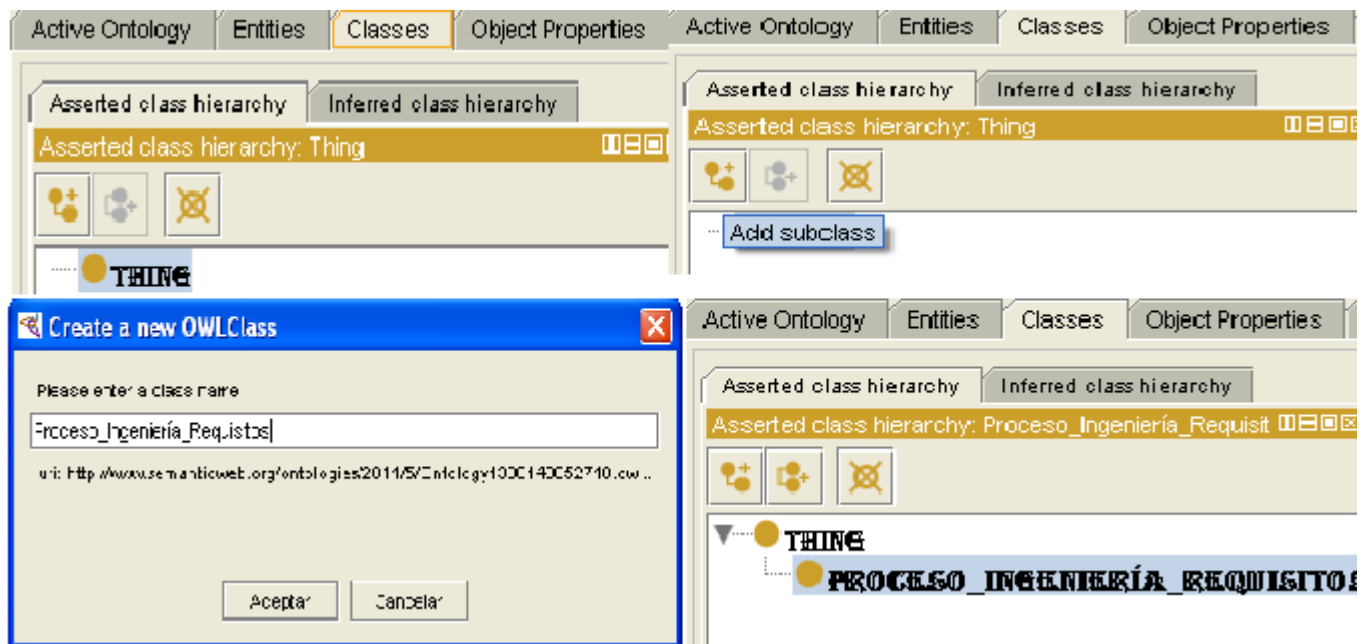


Figura 5: Crear la clase hija *Proceso_Ingeniería_Requisitos* en Protégé.

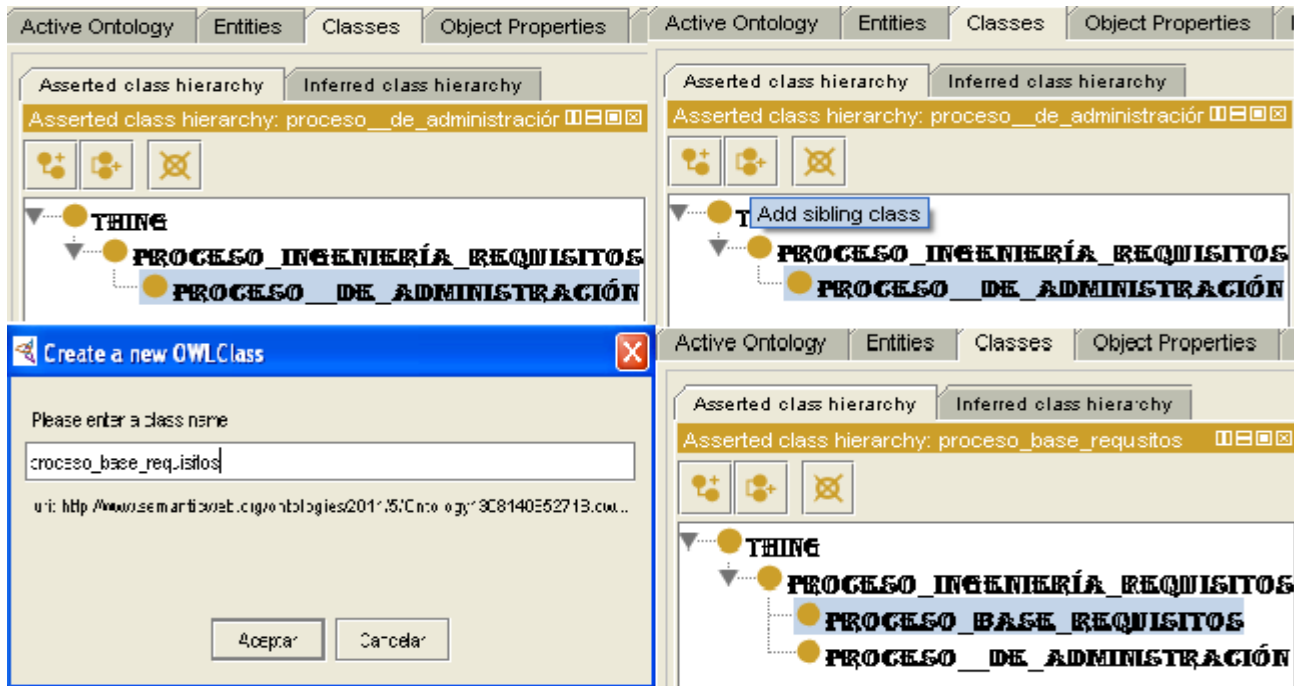


Figura 6: Crear la clase hermana Proceso_Base_Requisitos en Protégé.

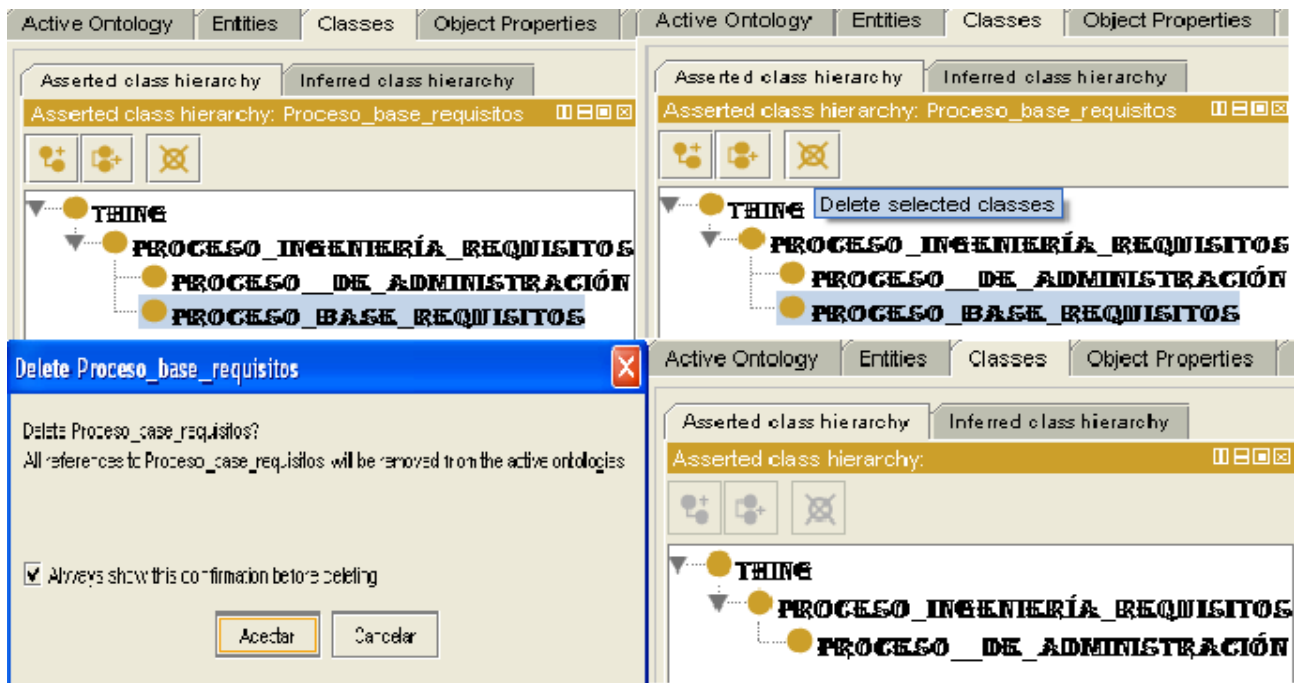


Figura 7: Eliminar la clase Proceso_Base_Requisitos en Protégé.

Las clases aisladas no entregan suficiente información para responder a las preguntas de competencia. Después de definir las clases, se describe la estructura interna de los conceptos. Para ello hay que contenerse encima de la clase a describir, se pasa al segundo plano de trabajo ubicado en la parte superior derecha, se selecciona *Annotations Add* que muestra una ventana donde se encuentran varios aspectos, en el caso de la conceptualización se selecciona *isDefinedBy* y se introduce el concepto. Mostrándonos como resultado el concepto de la clase cada vez que nos detengamos en ella (Figura 8). Todas estas anotaciones se pueden editar o eliminar.

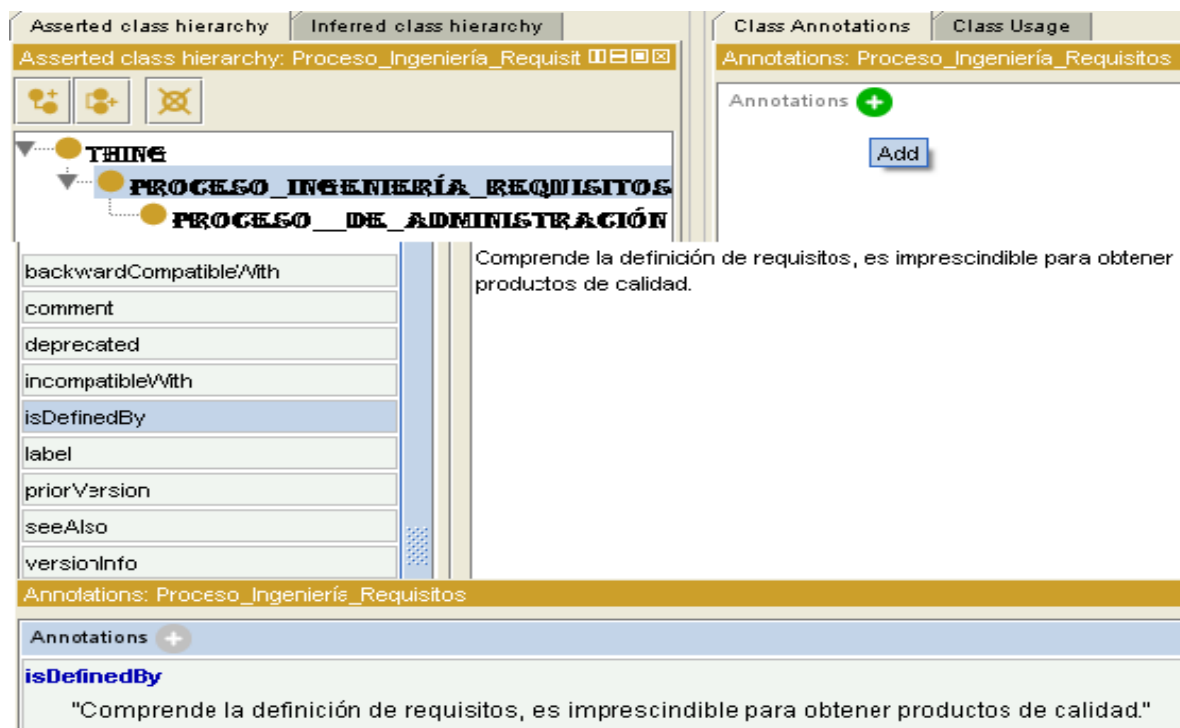


Figura 8: Definir el concepto de la clase Proceso_Ingeniería_Requisitos en Protégé.

A partir de la lista de términos definida y de su organización, se seleccionan los términos candidatos a ser propiedades. De esos términos seleccionados se analizan las relaciones existentes entre ellos y las clases definidas, para finalmente determinar las propiedades que forman parte de la ontología y asociar dichas propiedades a la clase correspondiente.

En Protégé-OWL, se crean las propiedades que forman parte de la ontología. Dentro de las propiedades que se crean hay propiedades cuyo valor corresponde a una o varias instancias de una clase llamadas

Propiedades de Objetos (Object Properties) y hay propiedades cuyo valor corresponde a un tipo de dato conocidas como Propiedades de Datos (Data Properties). De esta forma las propiedades describen las clases a las cuales pertenecen. La ontología está compuesta por 107 propiedades de objetos y 114 propiedades de datos.

Tanto las propiedades de objeto como las de datos se implementan de la misma forma. El marco de trabajo de la izquierda posee 3 botones, el primero para crear propiedades, el segundo para crear subpropiedades y el tercero para eliminarlas. Para establecer la propiedad de objeto *tieneartefacto* de la clase *Artefactos* hay que ir a la pestaña *Object Properties* y seleccionar el primer botón *Add property* del marco de trabajo de la izquierda el cual muestra una nueva ventana para introducir el nombre de la propiedad y al este ser aceptado queda creada la misma (Figura 9).

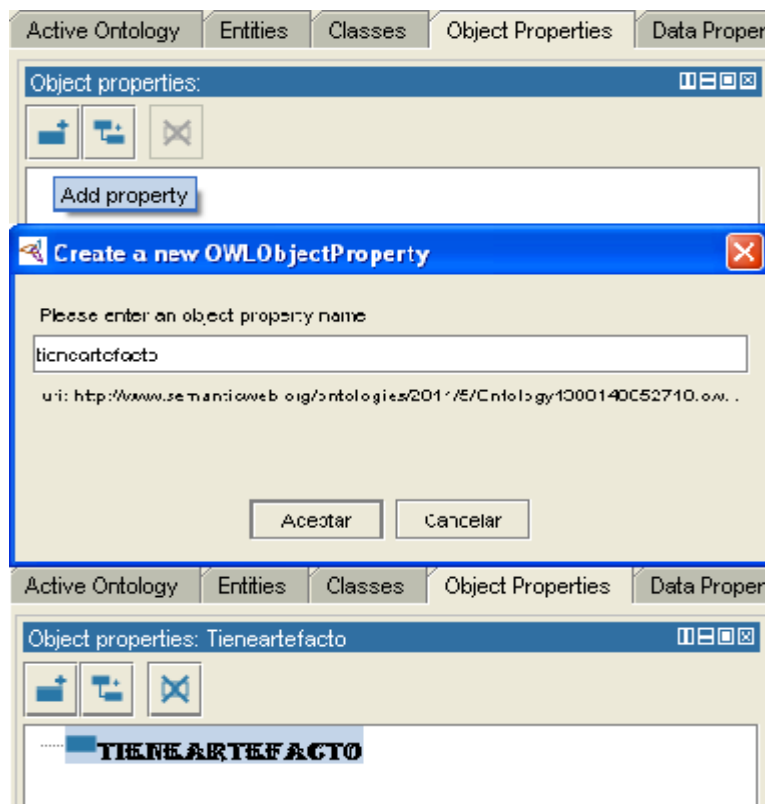


Figura 9: Crear la propiedad de objeto tieneartefacto de la clase Artefactos en Protégé.

Los slots o propiedades pueden tener diferentes facetas o características que describen el tipo de valor, los valores admitidos, el número de los valores (cardinalidad), el dominio, el rango, el tipo de propiedad y

otras características de los valores que los slots pueden tomar. A continuación se describen varias facetas comunes:

➤ Cardinalidad: define cuantos valores puede tener una propiedad. 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: describe slots con valores numéricos (algunas veces los tipos de valores Float e Integer son usados por ser más específicos).
- ✓ Boolean: son simples banderas sí o no.
- ✓ Enumerated: una lista específica de valores admitidos para el slot.
- ✓ Instance: admiten la definición de relaciones entre individuos. Los slots con este tipo de valor 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 $P1$).

En cuanto a la especificación de las características de la propiedad *tieneartefacto* correspondiente a la clase *Artefactos* se va al tercer marco de trabajo, seleccionando primeramente el tipo de propiedad, en este caso *Funcional*; en el caso del *Domain Add* este muestra una ventana para seleccionar la(s) clase(s) que describen la propiedad, en este caso *Tickets*; se pasa a *Ranges Add* que muestra también una nueva

ventana para marcar la propiedad en la que se está trabajando y seleccionar la(s) clase(s) permitidas, en este caso *Artefactos*, y así sucesivamente con el resto de las facetas que en esta propiedad no fueron necesarias definir (Figura 10).

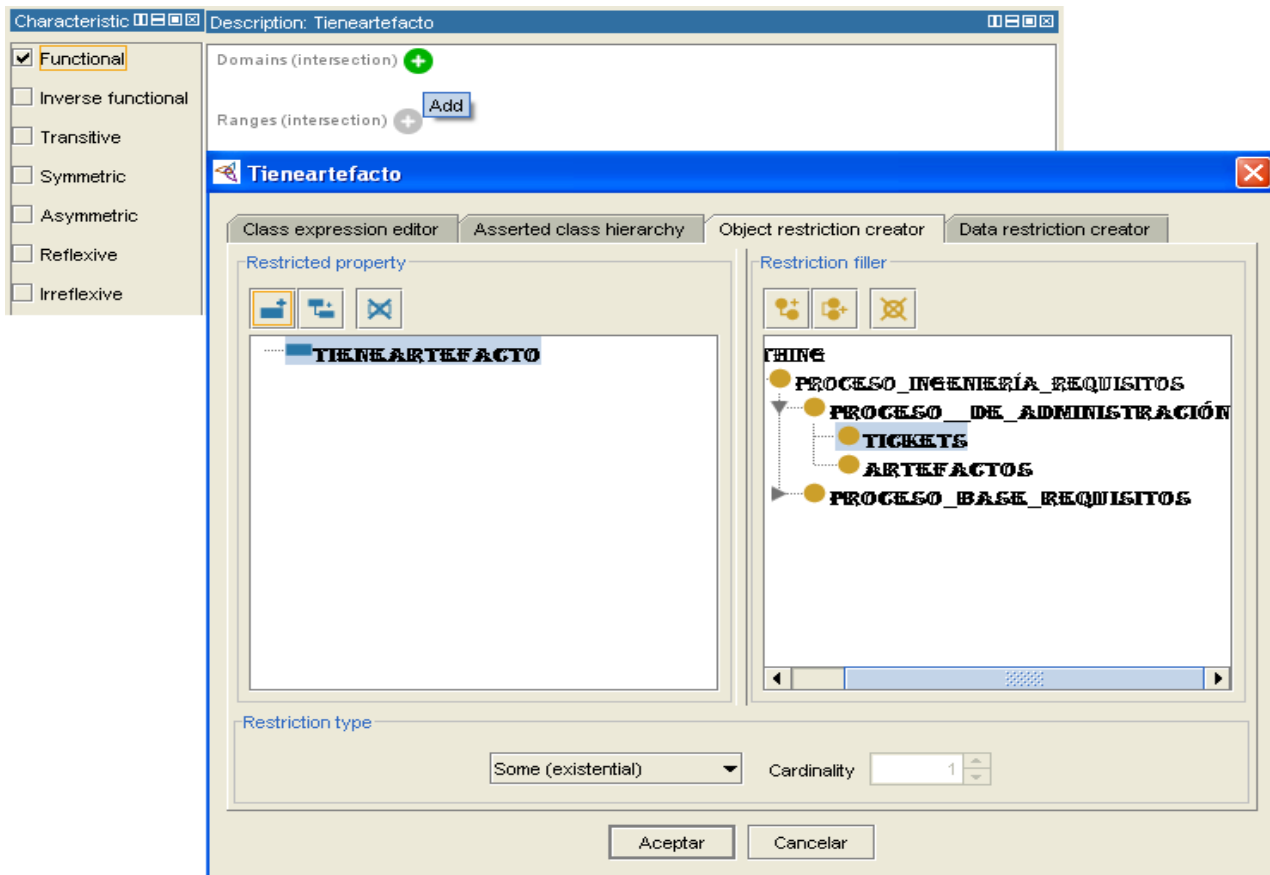


Figura 10: Definir las características de la propiedad de objeto tiene artefacto de la clase Artefactos en Protégé.

OWL permite definir a los individuos para hacer valer las propiedades de ellos y también se pueden utilizar en descripciones de las clases. Para la creación de los individuos hay que ir a la pestaña *Individuals*. En el marco de trabajo de la izquierda solo aparecen dos botones, el de agregar y el de eliminar instancias. Para insertar por ejemplo el objeto *Arquitectura_de_Información* se pincha el botón *Add individual* el que muestra un cuadro de diálogo para teclear el nombre y al aceptarlo queda creado el individuo (Figura 11). Para eliminar una instancia se selecciona la que se desea excluir y se pulsa el botón *Delete individual(s)* quedando eliminada la misma. Las instancias o individuos también tienen sus características, en el caso

de la Arquitectura_de_Información se iría al *Types Add* para especificar a qué clase pertenece, en este caso a *Artefactos* (Figura 12).

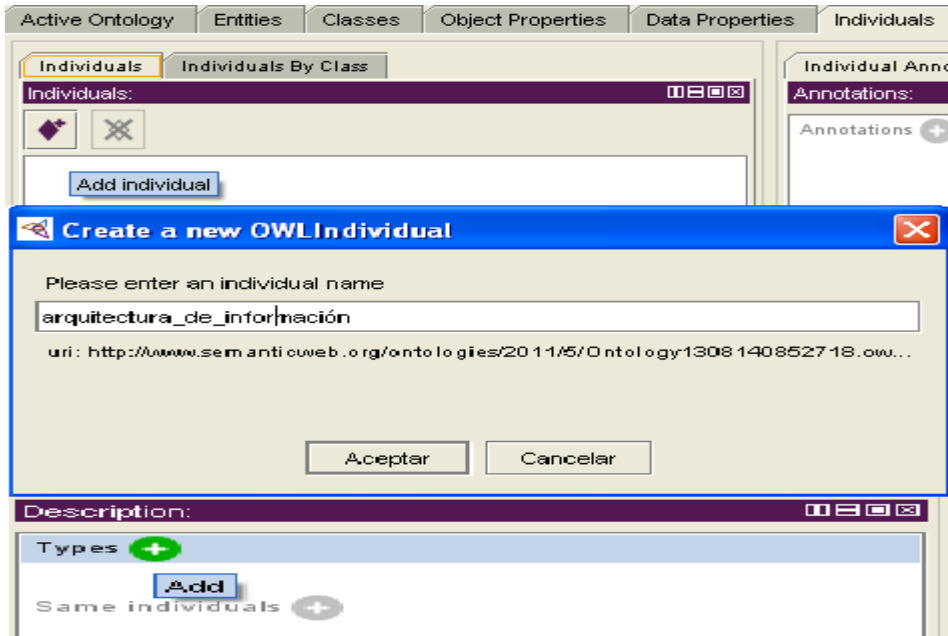


Figura 11: Crear el individuo Arquitectura_de_Información en Protégé.

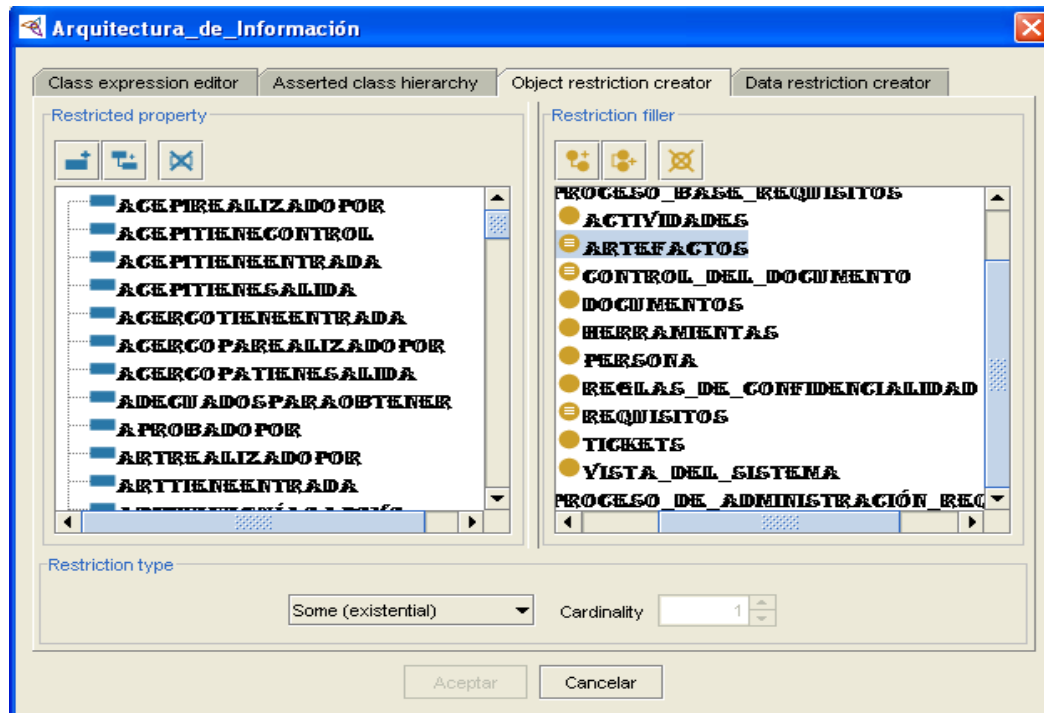


Figura 12: Definir las características del individuo Arquitectura_de_Información en Protégé.

2.6 Principios seguidos en la construcción de la ontología

Aquí se resumen un conjunto de principios que han demostrado ser útiles en el desarrollo de esta ontología.

- Claridad y Objetividad, planteado por (Gruber, 93) pero referenciado de (Breis, 2003), se evidencian en la ontología en los conceptos definidos y en las clases codificadas ya que se utilizó un lenguaje natural.
- Coherencia, planteado por (Gruber, 93) pero referenciado de (Breis, 2003), se nota en las definiciones de los conceptos y en las reglas de inferencia establecidas para responder a las preguntas de competencia, ya que son consistentes por estar basadas en bibliografías consultadas referentes a la Ingeniería de Requisitos.
- Máxima Extensibilidad Monótona, planteado por (Gruber, 93) pero referenciado de (Breis, 2003), este principio se observa en algunas clases establecidas que no poseen definiciones porque se infiere su contenido de su propio nombre, como ejemplo tenemos a las clases *Control_del_Documento*, *Vista_del_Sistema* y *Cambios_a_Requisitos*.
- Mínimos compromisos Ontológicos, planteado por (Gruber, 93) pero referenciado de (Breis, 2003), en este sentido la ontología especifica tan poco como sea posible acerca del significado de sus términos, solo se describen conceptualmente los términos con mayor peso de conocimiento en el dominio, ya sean los referentes a las clases *Artefactos*, *Documentos*, *Actividades*, *Roles* y *Requisitos*, dando a las partes comprometidas con la ontología la libertad para especializar e instanciar la ontología como sea requerido.
- Principio de distinción ontológica, planteado por Borgo y referenciado de (Breis, 2003), este se pone de manifiesto en que las clases representadas en la ontología son disjuntas.
- Minimizar la distancia semántica entre conceptos hermanos planteado por Arpidez y referenciado de (Breis, 2003), se pone en práctica a medida de que los conceptos similares son agrupados y representados como subclases de una clase y deberían ser definidos usando las mismas primitivas, mientras que los conceptos con menos similitud son representados aparte en la jerarquía. Por ejemplo las diferentes actividades que se realizan en el *Proceso_Base_de_Requisitos* son definidas como hijas de la clase padre *Actividades*.

2.7 Patrones de diseño utilizados en la construcción de la ontología

Entre los patrones de diseño existentes y utilizados en la construcción de la ontología se encuentra el de Lógica, el cual se usó para expresar a través de un vocabulario natural todas las clases, conceptos y relaciones definidas, teniendo esto gran impacto sobre los resultados y la eficiencia de los procedimientos de razonamiento de la ontología. El uso de este patrón constituye una buena práctica en el diseño de la ontología y sirve de gran apoyo al de Razonamiento, que se utilizó también ya que la ontología cuenta con reglas de inferencias que posteriormente serán utilizadas junto a consultas para establecer motores de razonamiento. Se empleó además el de Reingeniería ya que se conocía el modelo destino que era la ontología pero el modelo origen constituía un modelo no ontológico como son los modelos conceptuales. Además el proceso de construcción de una ontología es iterativo lo que hace utilizar este patrón mientras tenga vida útil la ontología. Otro patrón utilizado fue el de Presentación, el cual hace afirmar que es legible la ontología y por ende tiene un gran poder de usabilidad dándole una evaluación satisfactoria a la misma.

2.8 Paradigma de modelado utilizado

Durante la historia, los tres principales paradigmas para el modelado de ontologías han sido los sistemas basados en redes o redes semánticas, los sistemas basados en marcos o frames y los sistemas basados en lógica, en particular, las llamadas lógicas descriptivas. Este trabajo se basó en el tercer paradigma de representación del conocimiento, o sea, las lógicas descriptivas, que proporcionan fundamentos lógicos rigurosos a los sistemas basados en marcos. La Lógica Descriptiva (DL) es una familia de formalismos lógicos para la representación y el razonamiento sobre conjuntos de conceptos, sus relaciones y las instancias o clases concretas (KB).

Actualmente los sistemas de edición de Ontologías basados en lógicas descriptivas utilizan OWL como lenguaje para la representación del conocimiento. Tal es el caso de este trabajo que utiliza OWL DL porque proporciona una sintaxis amigable. Utilizar este paradigma y este lenguaje permiten usar también a Protégé-OWL que sigue la misma filosofía que Protégé-Frames en la que existen diferentes pestañas para la edición de los objetos de la ontología: “Classes”, “Properties” e “Individuals”.

Este paradigma en la pestaña “Classes” de OWL proporciona la vista de las propiedades e instancias que tengan como dominio directo la clase seleccionada o sean heredadas de alguna de las clases padres.

También indica si la clase es disjunta. La pestaña para las propiedades permite caracterizar a cada una de ellas de forma independiente y basándonos en cada una de las facetas existentes siempre que sea necesario, tales como anotaciones (*Annotations*), rango (*Range*), dominio (*Domains*), las propiedades inversas (*Inverse Properties*), entre otras.

2.9 Conclusiones del capítulo

En este capítulo se desarrolló la implementación de la ontología, extrayendo el conocimiento para la misma de los mapas conceptuales, se obtuvieron: las clases, la jerarquía de clases, las definiciones de las mismas, las propiedades y las instancias. 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 a los más específicos. Se tuvieron en cuenta los diferentes pasos que propone la metodología para prevenir errores en todo el transcurso del diseño. Se obtuvo finalmente el Modelo de la ontología que es la vista más detallada de la misma que nos brinda la herramienta Protégé. Se obtuvo también el código de la ontología que es indispensable para la realización de trabajos futuros.

Capítulo 3: Validación teórica de la ontología

3.1 Introducción del capítulo

Lograr una validación satisfactoria para la propuesta realizada, con el objetivo de comprobar que esta constituye una solución al problema planteado en la investigación, es sin lugar a dudas una de las grandes satisfacciones para cualquier investigador. En este capítulo se realizará la validación de la ontología mediante un Modelo para Validar Ontologías de Dominio que consta de 4 fases descritas en el epígrafe siguiente.

3.2 Validación para ontologías de dominio

Algunos objetivos del desarrollo de ontologías son: a) proporcionar una estructura de conocimiento común en un dominio, b) facilitar la reutilización del conocimiento y c) analizar el conocimiento. Estos objetivos se alcanzarán sólo si la ontología es de calidad y para garantizarlo es necesario que en cada fase del ciclo de vida de desarrollo, se evalúen los resultados parciales.

Es posible valorar la calidad de una ontología examinando un conjunto mínimo de criterios como son: que el vocabulario utilizado para representar el conocimiento tenga cobertura suficiente del corpus (conocimiento experto, textos y otras fuentes), que la ontología esté escrita de manera correcta, sin errores y conforme a las reglas del lenguaje utilizado, que la estructura taxonómica que organiza los conceptos y términos del dominio sea completa, sin redundancias y consistente y que satisfaga los requerimientos para los cuales fue creada y, de manera particular, que las preguntas de competencia sean respondidas adecuadamente. Este Modelo de Validación cuenta con 4 fases, las que se explican seguidamente:

Fase 1. Uso correcto del lenguaje.

Las actividades que permiten evaluar este criterio son: el lenguaje utilizado cumple con los estándares para desarrollos ontológicos ya que se utilizó OWL. Es un lenguaje sólido (cualquier expresión puede ser derivada a partir del conocimiento codificado) y completo (cualquier expresión que esté lógicamente implícita en la base de conocimiento puede ser derivada). De esta forma, se pueden aplicar métodos de razonamiento sobre la ontología de manera satisfactoria. También es importante resaltar que la escritura

está libre de errores o defectos, garantizando su futura utilización de manera exitosa, para ello en cada fase del ciclo de desarrollo se utilizó el marco de chequeo que provee el editor Protégé-OWL (Figura 13).

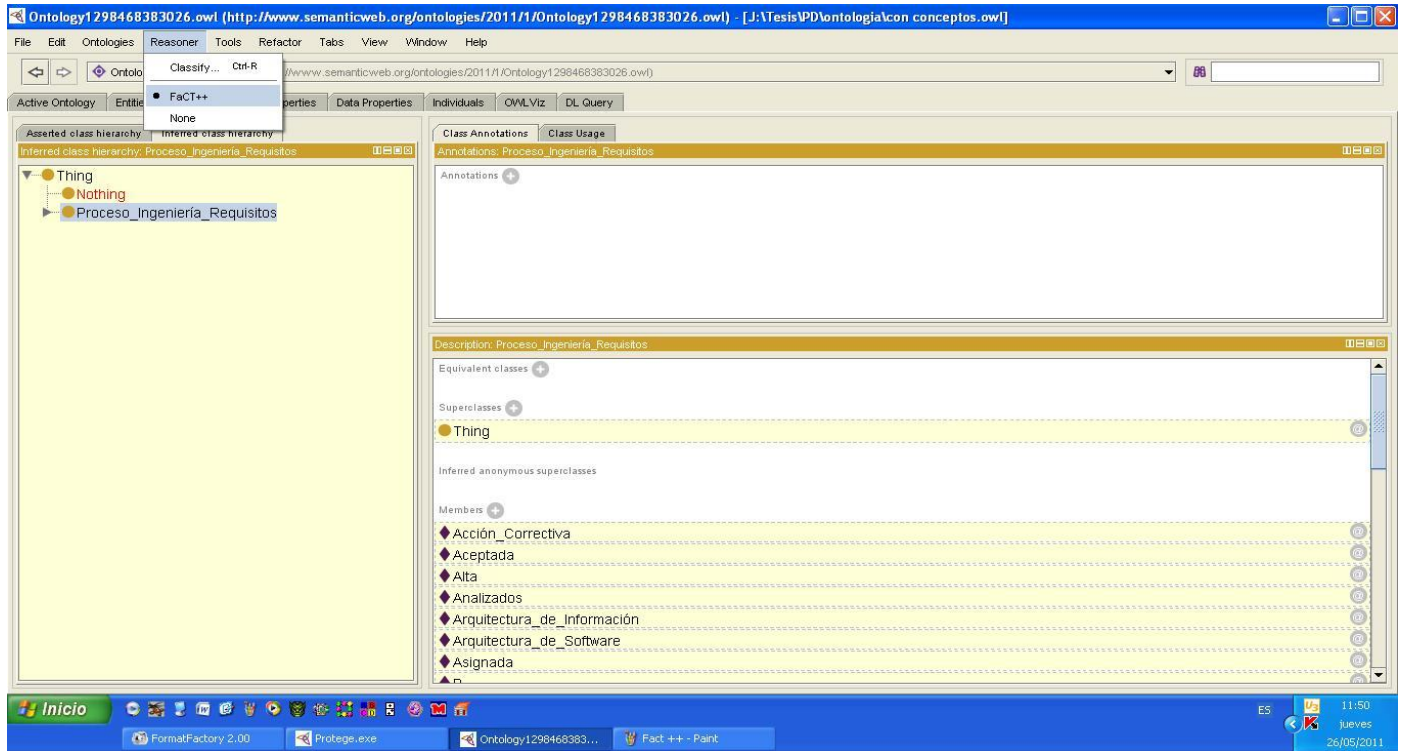


Figura 13: Marco de chequeo del Protégé-OWL.

Fase 2. Exactitud de la estructura taxonómica.

La evaluación taxonómica considera el chequeo de inconsistencias, completitud y redundancia de los términos de la taxonomía. En esta fase no se encontraron en la misma inconsistencias tales como: conceptos que no pertenecen a una clase en particular, ausencia de conceptos relevantes del dominio, clases e instancias con diferentes nombres pero definiciones similares o clases definidas como generalizaciones o especializaciones de sí mismas.

Fase 3. Validez del vocabulario.

Aquí se chequea que los términos codificados en la ontología existan y sean significativos en otras fuentes de conocimiento independientes, como por ejemplo, el conocimiento contenido en el corpus del dominio,

entendiéndose por corpus, al conjunto más extenso y ordenado posible de datos o textos científicos, literarios, etc., que pueden servir de base a una investigación. Las actividades realizadas en esta fase fueron las siguientes: se analizó el corpus del dominio identificando los términos significativos a partir de los documentos; por otra parte se evaluó el vocabulario considerando medidas de calidad de resultados usadas en escenarios de recuperación de información (búsqueda de documentos), tales como la precisión y la exhaustividad.

Calcular la Precisión nos brinda el porcentaje de los términos de la ontología que aparecen en el corpus con relación a la cantidad total de términos de la ontología, utilizando la siguiente expresión:

Precisión = $CO-C / COnto$ donde $CO-C$ es la cantidad de términos que se solapan entre la ontología y el corpus y la $COnto$ es la cantidad total de términos de la ontología.

Precisión = $204/204 = 1$ Esto nos demuestra que el 100% de los términos existentes en la ontología se encuentran en el corpus del dominio.

Calcular la Exhaustividad nos brinda el porcentaje de términos del corpus que aparecen en la ontología con relación al total de términos en el corpus, utilizando la siguiente expresión:

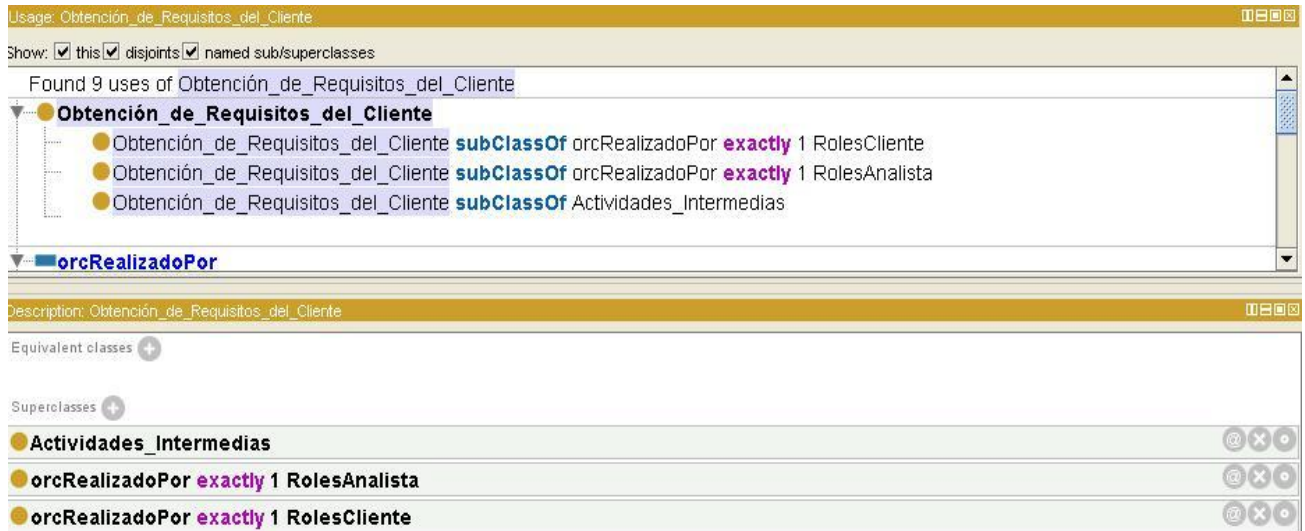
Exhaustividad = $CO-C / CCorp$ donde $CCorp$ representa la cantidad total de términos del corpus.

Exhaustividad = $204/296 = 0.69$ Esto nos demuestra que el 69% de los términos del corpus del dominio aparecen en la ontología.

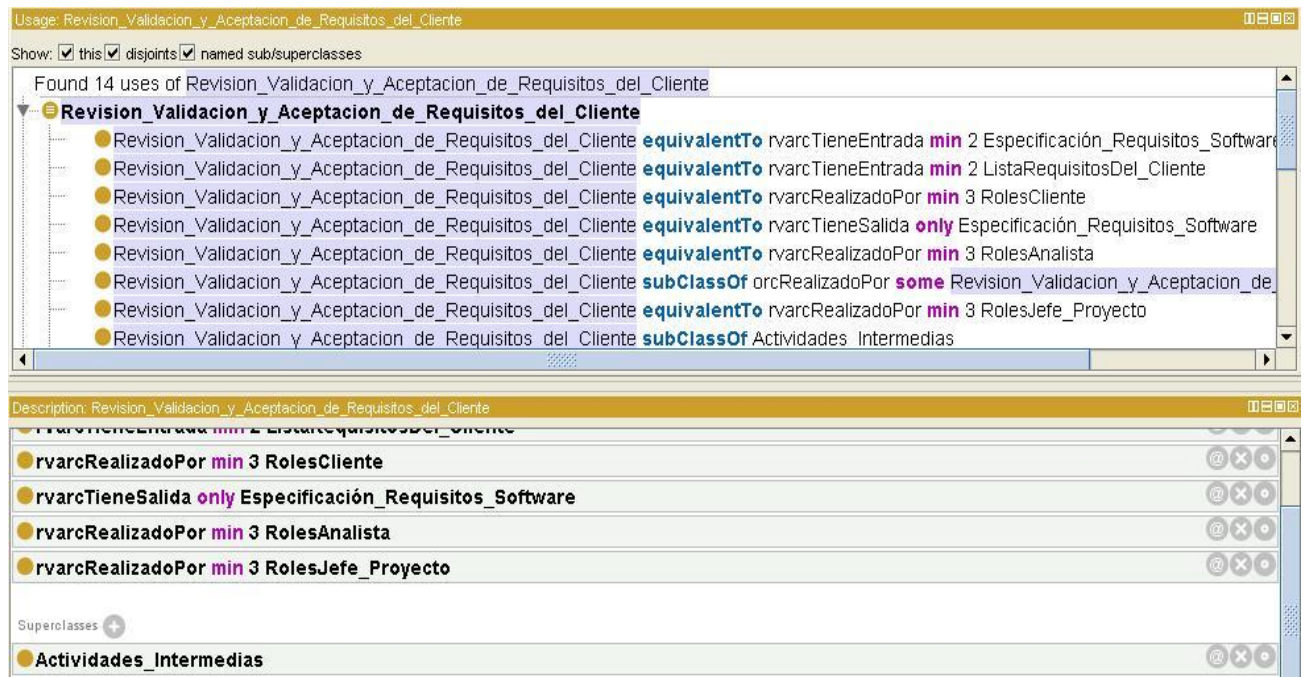
Fase 4. Adecuación a requerimientos.

En esta fase se debe cumplir con la metodología y con las preguntas de competencia. Se cumplieron todos los pasos planteados en la metodología utilizada que fue la de Pasos Específicos y las preguntas de competencia fueron resueltas mediante las reglas de inferencia. Se muestran a continuación:

1- ¿Quiénes son los responsables de la obtención de los requisitos del cliente?



2- ¿Con qué actividad está relacionada la actividad de Validar los requisitos?



3- ¿A qué rol le pertenece la Solicitud de cambios?

The screenshot shows the Protege interface for the class **Gestionar_Cambios**. The top panel, titled "Usage: Gestionar_Cambios", lists 17 uses of the class. The bottom panel, titled "Description: Gestionar_Cambios", shows equivalent classes.

Usage: Gestionar_Cambios

Show: this disjoints named sub/superclasses

Found 17 uses of Gestionar_Cambios

- Gestionar_Cambios **equivalentTo** hcgcTieneGuíaOApoyo **only** Herramientas_Trazabilidad
- Gestionar_Cambios **equivalentTo** hcgcTieneSalida **only** Control_de_Cambios
- Gestionar_Cambios **equivalentTo** hcgcTieneEntrada **only** Solicitud_Cambio
- Gestionar_Cambios **equivalentTo** hcgcRealizadoPor **min** 4 RolesAnalista
- Gestionar_Cambios **equivalentTo** hcgcRealizadoPor **min** 4 RolesAdministrador_Calidad
- Gestionar_Cambios **equivalentTo** hcgcRealizadoPor **min** 4 RolesJefe_Proyecto
- Gestionar_Cambios **subClassOf** Actividades_Fin
- Gestionar_Cambios **equivalentTo** hcgcRealizadoPor **min** 4 RolesArquitecto_Software
- Gestionar_Cambios **subClassOf** Roles

Description: Gestionar_Cambios

Equivalent classes

- hcgcTieneGuíaOApoyo **only** Herramientas_Trazabilidad
- hcgcTieneSalida **only** Control_de_Cambios
- hcgcTieneEntrada **only** Solicitud_Cambio
- hcgcRealizadoPor **min** 4 RolesAnalista
- hcgcRealizadoPor **min** 4 RolesAdministrador_Calidad
- hcgcRealizadoPor **min** 4 RolesJefe_Proyecto
- hcgcRealizadoPor **min** 4 RolesArquitecto_Software

4- ¿Qué herramienta de trazabilidad se puede utilizar para crear sistema de trazabilidad con requisitos y clientes?

The screenshot shows the Protege interface for the class **Crear_Sistema_Trazabilidad_con_Requisitos_y_Clientes**. The top panel, titled "Usage: Crear_Sistema_Trazabilidad_con_Requisitos_y_Clientes", lists 8 uses of the class. The bottom panel, titled "Description: Crear_Sistema_Trazabilidad_con_Requisitos_y_Clientes", shows equivalent classes.

Usage: Crear_Sistema_Trazabilidad_con_Requisitos_y_Clientes

Show: this disjoints named sub/superclasses

Found 8 uses of Crear_Sistema_Trazabilidad_con_Requisitos_y_Clientes

- Crear_Sistema_Trazabilidad_con_Requisitos_y_Clientes **equivalentTo** tieneManual **some** Manual_Herramienta_Trazabilidad
- Crear_Sistema_Trazabilidad_con_Requisitos_y_Clientes **subClassOf** Actividades_Intermedias

crtrcTieneEntrada

- crtrcTieneEntrada **domain** Crear_Sistema_Trazabilidad_con_Requisitos_y_Clientes

cstrcRealizadoPor

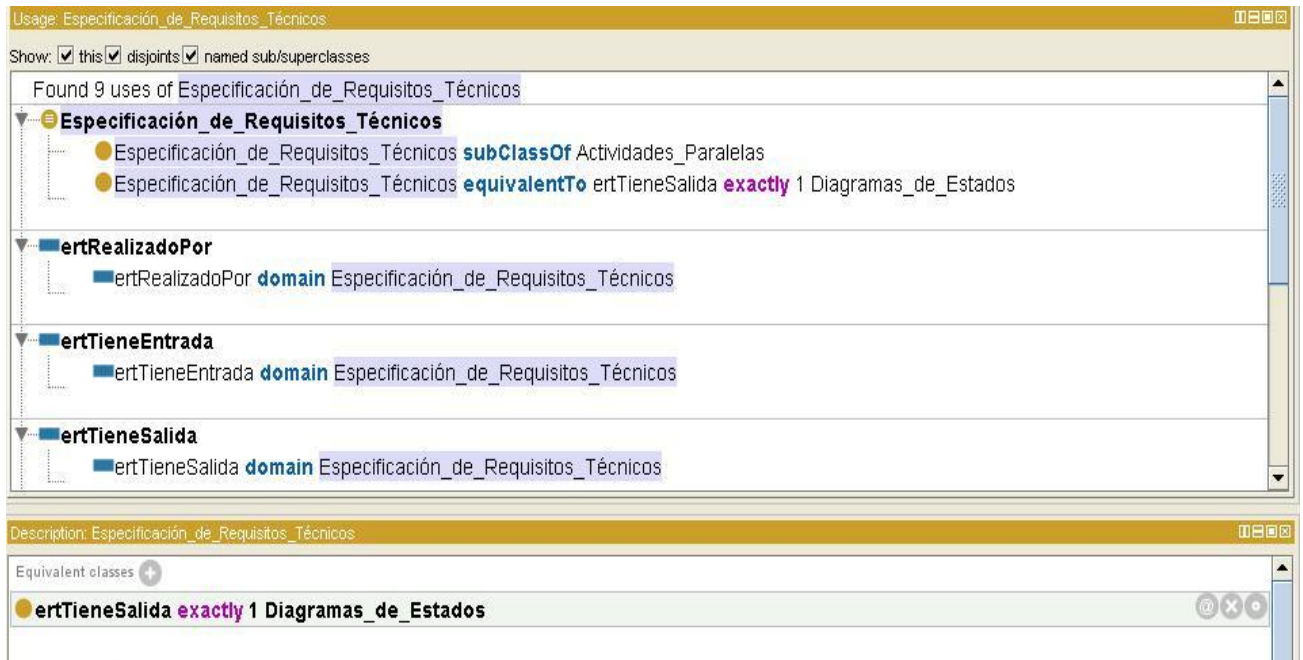
- cstrcRealizadoPor **domain** Crear_Sistema_Trazabilidad_con_Requisitos_y_Clientes

Description: Crear_Sistema_Trazabilidad_con_Requisitos_y_Clientes

Equivalent classes

- tieneManual **some** Manual_Herramienta_Trazabilidad

5- ¿Cuáles son los diagramas que pueden ser modelados a partir de la especificación de los requisitos técnicos?



6- ¿Cuáles son los riesgos de los requisitos que se obtienen al entrar los requisitos del cliente y los requisitos técnicos?



- 7- ¿Quiénes son los encargados de asegurar la consistencia entre los requisitos del cliente con otros productos o artefactos del proyecto?

Usage: Asegurar_Consistencia_Entre_Requisitos_del_Cliente_con_Otros_Productos_O_Artefactos_del_Proyecto

Show: this disjoint named sub/superclasses

Found 6 uses of Asegurar_Consistencia_Entre_Requisitos_del_Cliente_con_Otros_Productos_O_Artefactos_del_Proyecto

- Asegurar_Consistencia_Entre_Requisitos_del_Cliente_con_Otros_Productos_O_Artefactos_del_Proyecto **equivalentTo** **acercopaRealizadoPor min 2 RolesAnalista**
- Asegurar_Consistencia_Entre_Requisitos_del_Cliente_con_Otros_Productos_O_Artefactos_del_Proyecto **equivalentTo** **acercopaRealizadoPor min 2 RolesAdministrador_Calidad**
- Asegurar_Consistencia_Entre_Requisitos_del_Cliente_con_Otros_Productos_O_Artefactos_del_Proyecto **subClassOf** **Actividades_Intermedias_PAR**

acercopaRealizadoPor

- acercopaRealizadoPor **domain** Asegurar_Consistencia_Entre_Requisitos_del_Cliente_con_Otros_Productos_O_Artefactos_del_Proyecto

Description: Asegurar_Consistencia_Entre_Requisitos_del_Cliente_con_Otros_Productos_O_Artefactos_del_Proyecto

Equivalent classes +

- acercopaRealizadoPor **min 2 RolesAnalista**
- acercopaRealizadoPor **min 2 RolesAdministrador_Calidad**

- 8- ¿Quiénes llevan el seguimiento de los requisitos?

Usage: Llevar_Seguimiento_de_Requisitos

Show: this disjoint named sub/superclasses

Found 6 uses of Llevar_Seguimiento_de_Requisitos

- Llevar_Seguimiento_de_Requisitos **subClassOf** **Actividades_Intermedias_PAR**
- Llevar_Seguimiento_de_Requisitos **equivalentTo** **IsrRealizadoPor only RolesEquipo_Desarrollo**

IsrRealizadoPor

- IsrRealizadoPor **domain** Llevar_Seguimiento_de_Requisitos

IsrTieneControl

- IsrTieneControl **domain** Llevar_Seguimiento_de_Requisitos

Description: Llevar_Seguimiento_de_Requisitos

Equivalent classes +

- IsrRealizadoPor **only RolesEquipo_Desarrollo**

- 9- ¿Qué tipo de actividad se ejecuta con la evaluación del impacto de los cambios?
10- ¿Quiénes son los responsables del registro de peticiones de cambios?

The screenshot displays the 'Usage' window of an ontology editor for the class 'Efectuar_de_Manera_Controlada_Cambios_Realizados_en_Requisitos'. The window title is 'Usage: Efectuar_de_Manera_Controlada_Cambios_Realizados_en_Requisitos'. Below the title, there are checkboxes for 'this', 'disjoints', and 'named sub/superclasses', all of which are checked. The main content area shows 'Found 9 uses of Efectuar_de_Manera_Controlada_Cambios_Realizados_en_Requisitos'. The uses are listed as follows:

- Efectuar_de_Manera_Controlada_Cambios_Realizados_en_Requisitos **subClassOf** Actividades_Fin_PAR
- Efectuar_de_Manera_Controlada_Cambios_Realizados_en_Requisitos **equivalentTo** emccrrRealizadoPor **min** 3 RolesJefe_Proyecto
- Efectuar_de_Manera_Controlada_Cambios_Realizados_en_Requisitos **equivalentTo** emccrrRealizadoPor **min** 3 RolesEquipo_Desarrollo
- Efectuar_de_Manera_Controlada_Cambios_Realizados_en_Requisitos **equivalentTo** emccrrRealizadoPor **min** 3 RolesCliente

Below the list, the 'emccrrRealizadoPor' class is expanded to show its domain: 'emccrrRealizadoPor **domain** Efectuar_de_Manera_Controlada_Cambios_Realizados_en_Requisitos'. The 'emccrrTieneControl' class is also visible but not expanded.

The bottom section of the window shows the 'Description: Efectuar_de_Manera_Controlada_Cambios_Realizados_en_Requisitos' and a list of 'Equivalent classes' with a plus sign icon:

- emccrrRealizadoPor **min** 3 RolesJefe_Proyecto
- emccrrRealizadoPor **min** 3 RolesEquipo_Desarrollo
- emccrrRealizadoPor **min** 3 RolesCliente

3.3 Conclusiones del capítulo

En este capítulo se trató la validación de la solución al problema planteado. Se realizó mediante el Modelo de Validación para Ontologías de Dominio demostrando que se utilizó el lenguaje correcto, que la estructura taxonómica no contiene inconsistencias ni redundancias, que la ontología posee un alto porcentaje de precisión y exhaustividad y que se cumplió con la metodología y las preguntas de competencia.

Conclusiones

Con la realización del presente trabajo se adquirieron conocimientos teóricos sobre las ontologías y la gestión del conocimiento, conceptos fundamentales que permitieron interpretar la propuesta de solución. Se realizó un análisis acerca de los lenguajes, herramientas y metodologías a utilizar para facilitar el proceso de desarrollo de la ontología; con el uso de estos elementos se alcanzó como resultado una ontología que solucionó la problemática planteada. Se le dió cumplimiento al desarrollo de la propia ontología creando las clases, la jerarquía de clases, las propiedades de las mismas, las instancias y definiendo los conceptos de las clases. Se demostró que la ontología está correctamente diseñada, logrando así que tenga la menor cantidad de errores posibles demostrando su validez y el cumplimiento de los objetivos propuestos.

Recomendaciones

- Continuar enriqueciendo la ontología para que su dominio se haga más extenso.
- Realizar consultas a la misma para aumentar sus funcionalidades.
- Integrarla a un sistema de información basado en ontologías.

Bibliografía

Anna V. Zhdanova, Uwe Keller. 2005. *Choosing an Ontology Language*. s.l. : Engineering and Technology, 2005.

Antonio Martín, Sonsoles Celestino, Adela Valdenebro y Julia Mensaque. *ONTOLOGÍAS E INTELIGENCIA ARTIFICIAL PARA LA RECUPERACIÓN EFICIENTE DEL CONOCIMIENTO*. Sevilla, España : s.n.

Ascaño, Marta Isabel Blaquier. *Una Ontología para el Proceso Jurídico*. Ciudad de la Habana : s.n.

Breis, Jesualdo Tomás Fernández. 2003. *Un Entorno de Integración de Ontologías para el Desarrollo de Sistemas de Gestión*. Murcia : s.n., 2003.

2011. Calidad del Software. *Calidad del Software*. [En línea] SlideShare, 2011. [Citado el: 6 de enero de 2011.] <http://www.slideshare.net/elsuse/calidad-del-software>.

Contreras, Jesús. *TUTORIAL ONTOLOGÍAS*. Madrid : s.n.

Diana Marcela Sánchez, José María Cavero, Esperanza Marcos. *Ontologías y MDA: una revisión de la literatura*. España : s.n.

Esmeralda Ramos, Haydemar Núñez, Roberto Casañas. 2009. Esquema para evaluar ontologías únicas para un dominio de conocimiento. *Esquema para evaluar ontologías únicas para un dominio de conocimiento*. [En línea] Enlace, abril de 2009. [Citado el: 15 de abril de 2011.] http://www.scielo.org.ve/scielo.php?script=sci_arttext&pid=S1690-75152009000100005&lng=es&nrm=iso. ISSN 1690-7515..

EVOLUCIÓN DE LOS LENGUAJES UTILIZADOS EN LA CONSTRUCCIÓN DE LA WEB SEMÁNTICA. 2006. 32, s.l. : UTP, 2006. ISSN 0122-1701.

Fernández López, M, Gómez Pérez, A and Rojas Amaya, M D. 2000. *Ontologies crossed life*. Francia : s.n., 2000.

2004. Frequently Asked Questions on W3C's Web Ontology Language (OWL). *Frequently Asked Questions on W3C's Web Ontology Language (OWL)*. [En línea] 10 de febrero de 2004. [Citado el: 20 de noviembre de 2010.] <http://www.w3.org/2003/08/owlfaq>.

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

Graciela Barchini, Margarita Álvarez, Susana Herrera. 2006. [En línea] 11 de abril de 2006. [Citado el: 10 de mayo de 2011.]

<http://www.google.com/cu/url?sa=t&source=web&cd=1&ved=0CB1QFjAA&url=http%3A%2F%2Fwww.jistem.fea.usp.br%2Findex.php%2Fjistem%2Farticle%2Fdownload%2F10.4301%25252FS1807-17752006000100001%2F42&rct=j&q=sistemas%20de%20informaci%C3%B3n%20basados%20en%20ontolog>.

Grigoris Antoniou, Frank Van Harmelen. *Web Ontology Language: OWL*. Amsterdam : s.n.

Gruber, T. R. 1993. *Translation approach to portable ontologies*. s.l. : Knowledge acquisition, 1993.

Guarino, Nicola. 1998. Formal Ontology and Information Systems. *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems*. s.l. : IOS Press, 1998.

—. **1997.** *Understanding, Building and Using Ontologies*. 1997.

—. Understanding, Building, And Using Ontologies. *Understanding, Building, And Using Ontologies*. [En línea] LADSEB-CNR. <http://ksi.cpsc.ucalgary.ca/KAW/KAW96/guarino/guarino.html>.

2010. Guía Breve de Web Semántica. *Guía Breve de Web Semántica*. [En línea] 6 de mayo de 2010. [Citado el: 10 de mayo de 2011.] <http://www.w3c.es/divulgacion/guiasbreves/websemantica>.

Horridge, Matthew. 2009. *A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools*. Manchester : s.n., 2009.

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

Ismael Sanz, Ernesto Jiménez-Ruiz. 2007. *Ontologías en Informática*. 2007.

Jonás Montilva, Mauricio Rojas. 2010. *Método para la conceptualización en el modelado del negocio en procesos de software*. 2010.

López, M. Fernández. *Overview Of Methodologies For Building Ontologies*. Madrid : s.n.

María Poveda, Mari Carmen Suárez-Figueroa, Asunción Gómez-Pérez. *Malas Prácticas en Ontologías*. Madrid : s.n.

Maroto, Juan Carrion. Gestión del Conocimiento.com. *Gestión del Conocimiento.com*. [En línea] [Citado el: 10 de 11 de 2010.] <http://www.gestiondelconocimiento.com/introduccion.htm>.

2008. Modelos de Gestion de la Calidad del Software. [En línea] 12 de enero de 2008. <http://modelosdegestiondelcalidad.blogspot.com/>.

- Natalya F. Noy, Deborah L. McGuinness. 2005.** *Desarrollo de Ontologías-101: Guía Para Crear Tu Primera Ontología*. California : s.n., 2005.
- Navarro Galindo, José L. and Samos Jiménez, José.** *Una panorámica actual de software para el trabajo con ontologías*. Granada, España : s.n.
- Noman Islam, Abu Zafar Abbasi and Zubair A. Shaikh.** *Semantic Web: Choosing the Right Methodologies, Tools and Standards*. Karachi, Pakistan : s.n.
- Nuñez, Esmeralda Ramos y Haydemar. 2007.** *ONTOLOGÍAS: componentes, metodologías, lenguajes, herramientas y aplicaciones*. Caracas : s.n., 2007.
- Ontoknowledge. *Ontoknowledge*. [En línea] <http://www.ontoknowledge.org/oil>.
- Ontoknowledge. *Ontoknowledge*. [En línea] <http://www.ontoknowledge.org/tools/ontoedit.shtml>.
- Ontolingua. *Ontolingua*. [En línea] <http://www.ksl.stanford.edu/software/ontolingua>.
- Ontotext. [En línea] <http://www.ontotext.com>.
- Ontotext. *Ontotext*. [En línea] <http://www.ontotext.com/kim>.
- Paz, Rosa Maria Torres de.** *El proceso de Ingeniería de Requisitos en el ciclo global del*. Sevilla : s.n.
- Peñalvo, Francisco José García.** *Web Semántica y Ontologías*. Salamanca : s.n.
- Pressman, Roger S. 2006.** *Ingeniería de Software. Un enfoque práctico*. s.l. : Mcgraw Hill, 2006.
- Protege. *Protege*. [En línea] <http://protege.stanford.edu/plugins>.
- Protégé. *Protégé*. [En línea] <http://protege.stanford.edu>.
- Quezada, Jorge Naranjo. 2009.** Gestión del Conocimiento. *Gestión del Conocimiento*. [En línea] 12 de mayo de 2009. <http://gestiondelconocimientokm.wordpress.com/tag/gestion/>.
- Rafael Pedraza-Jiménez, Lluís Codina y Cristófol Rovira.** Web semántica y ontologías en el procesamiento de la información documental. s.l. : DOI.
- Roger L. Costello, David B. Jacobs. 2003.** *A Quick Introduction to OWL Web Ontology Language*. s.l. : The MITRE Corporation, 2003.
- Samper Zapater, José Javier. 2005.** *Tesis Doctoral: Ontologías para Servicios Web Semántico de Información de Tráfico*. Valencia, España : s.n., 2005.
- Sommerville, Ian. 2005.** *Ingeniería de Software*. s.l. : Pearson, 2005.

Tallarico, Marcelo. *Uso de ontologías en tareas de recuperación de información.*

Tello, Adolfo Lozano. 2002. *MÉTRICA DE IDONEIDAD DE ONTOLOGÍAS.* Madrid : s.n., 2002.

UNA PROPUESTA DE METAONTOLOGÍA PARA LA EDUCACIÓN DE REQUISITOS. **Carlos M. Zapata, Gloria L. Giraldo, Jhon E. Mesa. 2010.** 1, Chile : s.n., 2010, Vol. 18 .

Using Explicit Ontologies in KBS Development. **G. Van Heijst, A. T. Schreiber y J. B. Wielinga. 1996.** 1996.

Aquí se muestran el resto de los mapas conceptuales. El primero se refiere a los artefactos, documentos, roles, inconsistencias, etc., correspondientes al Proceso de Administración de Requisitos (Figura 14).

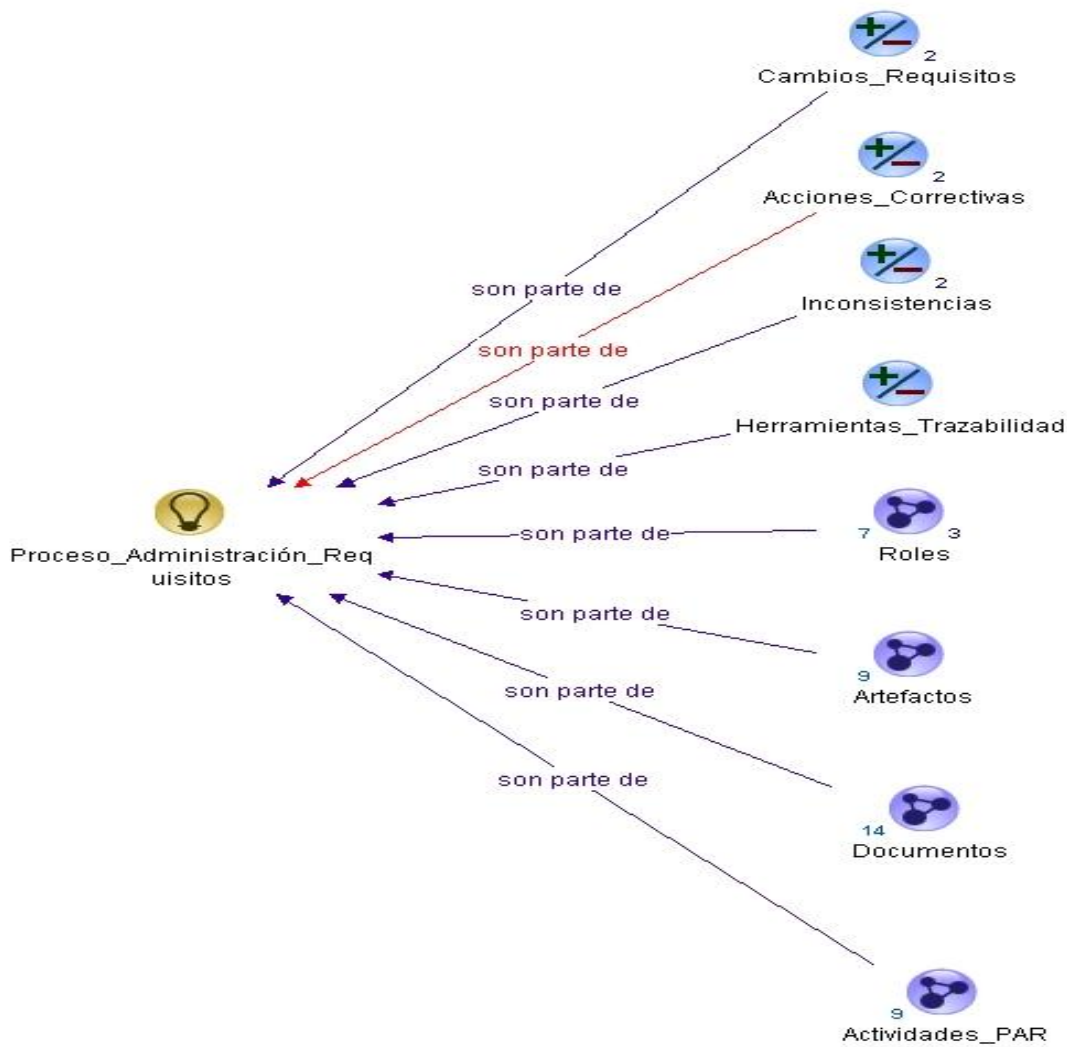


Figura 14: Proceso de Administración de Requisitos.

El que sigue es referente a los artefactos generados en ambos procesos, tanto en el Proceso de Administración de Requisitos como en el Proceso Base de Requisitos (Figura 15).

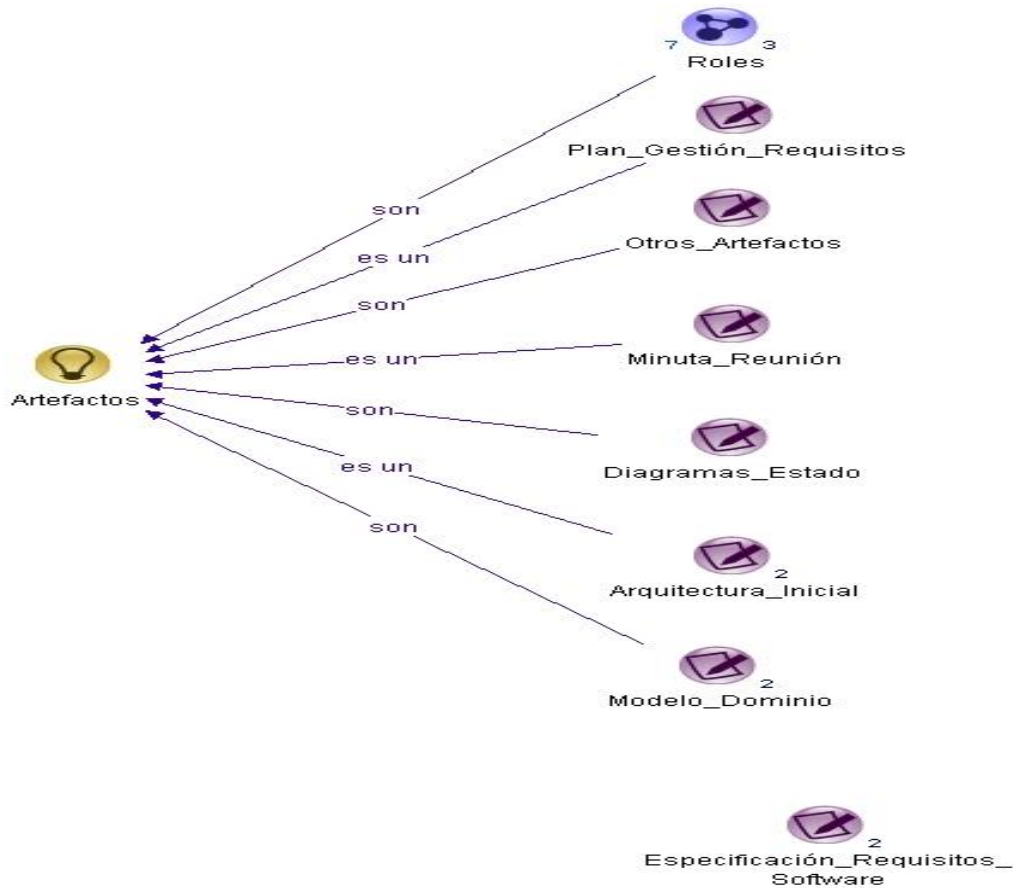


Figura 15: Artefactos.

El que se muestra a continuación brinda los documentos que se realizan también en ambos procesos (Figura 16).

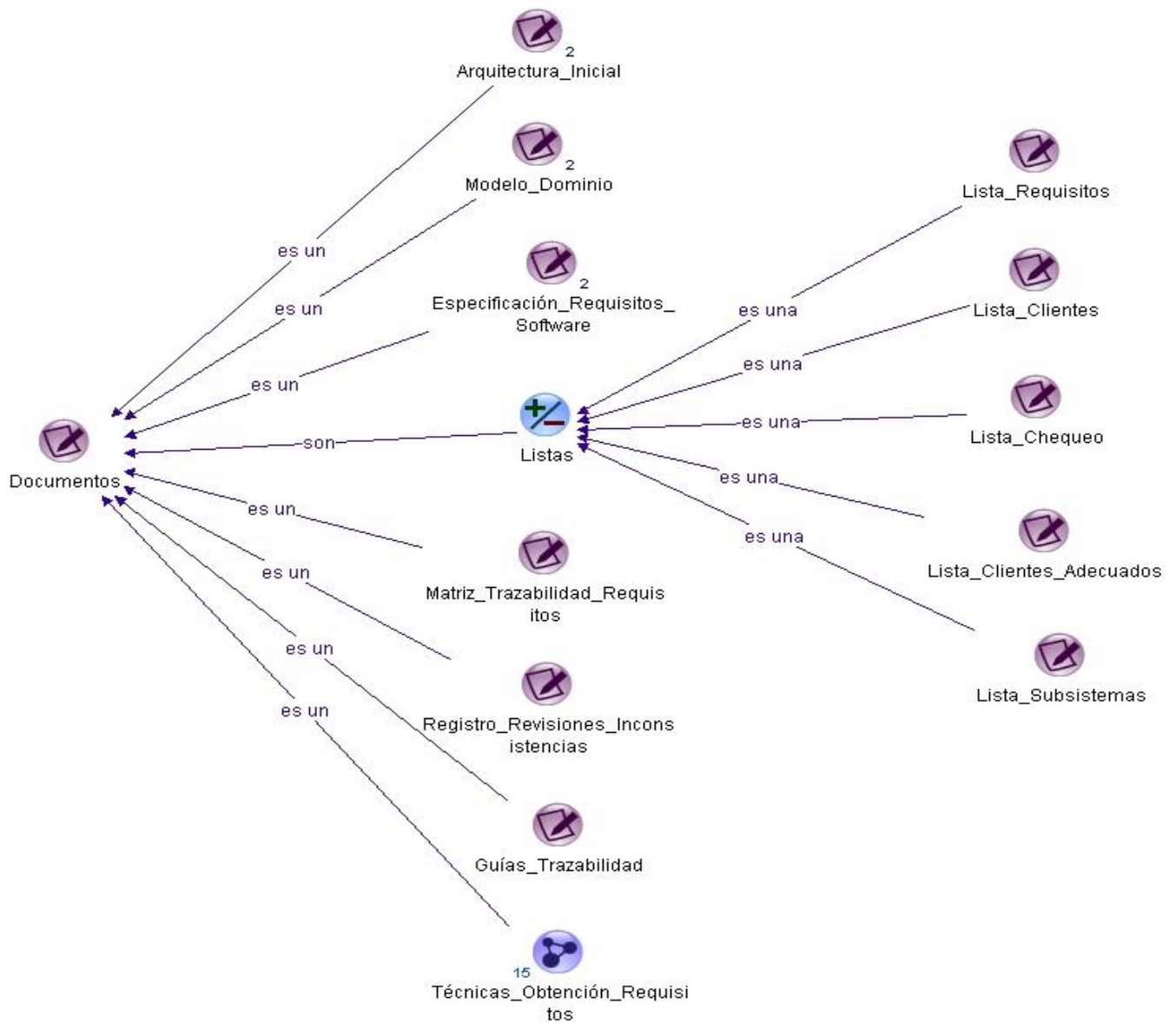


Figura 16: Documentos.

En el mapa conceptual de la Figura 17 se observan las Técnicas de Obtención de Requisitos correspondientes a los documentos a realizar.

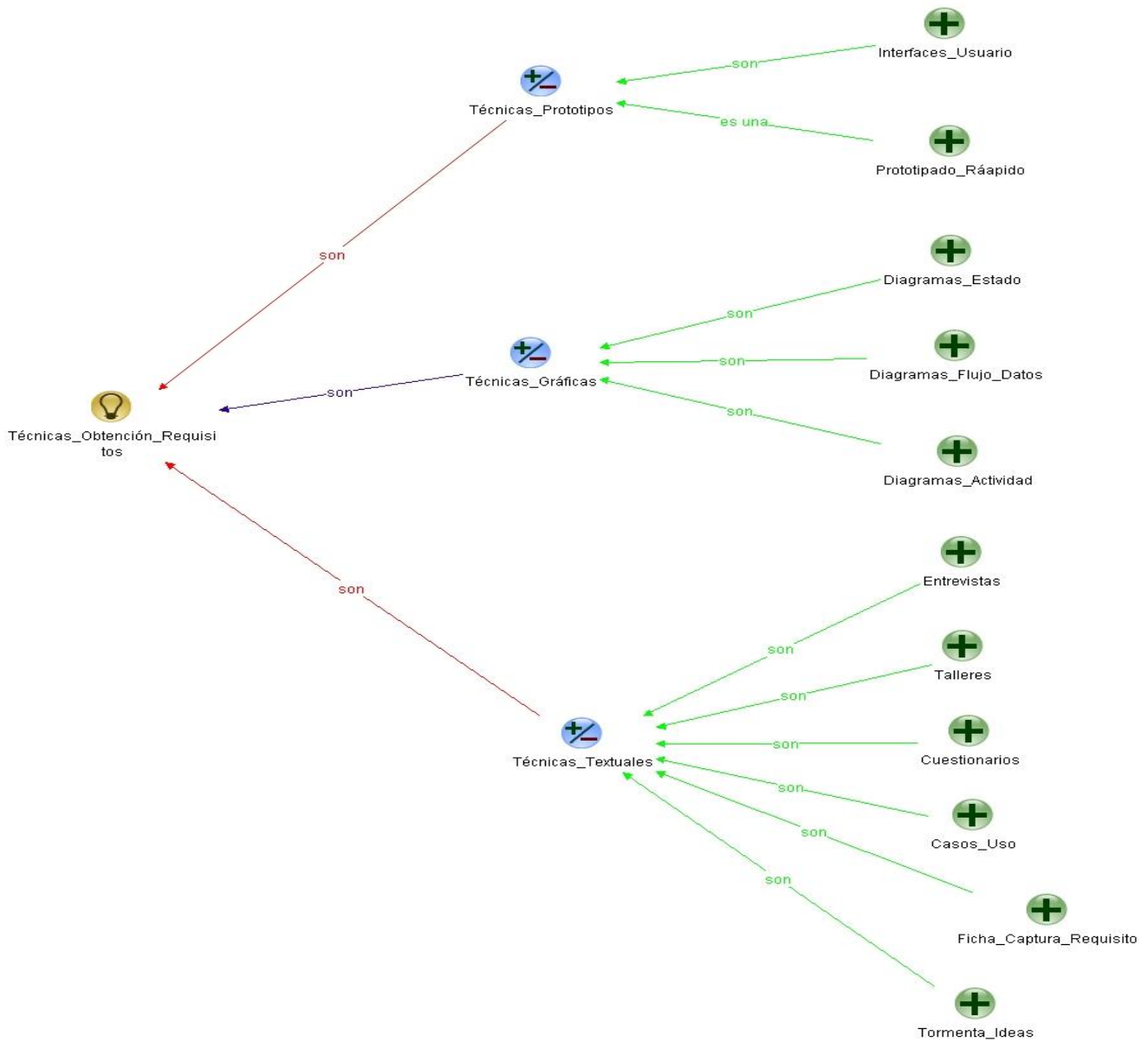


Figura 17: Técnicas de Obtención de Requisitos.

En la Figura 18 se muestran los roles presentes en ambos procesos.

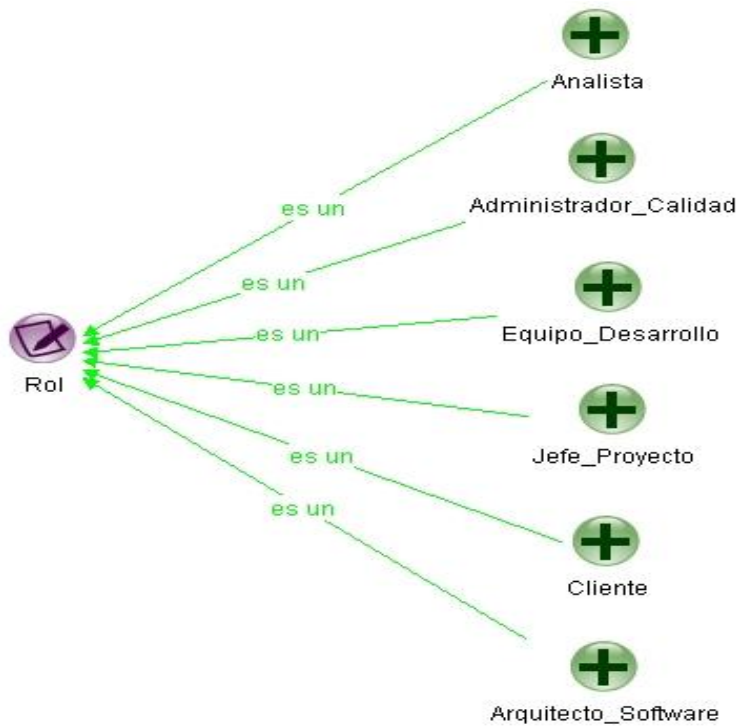


Figura 18: Roles.

El siguiente modelo muestra las actividades que se realizan en el Proceso de Administración de Requisitos (Figura 19).

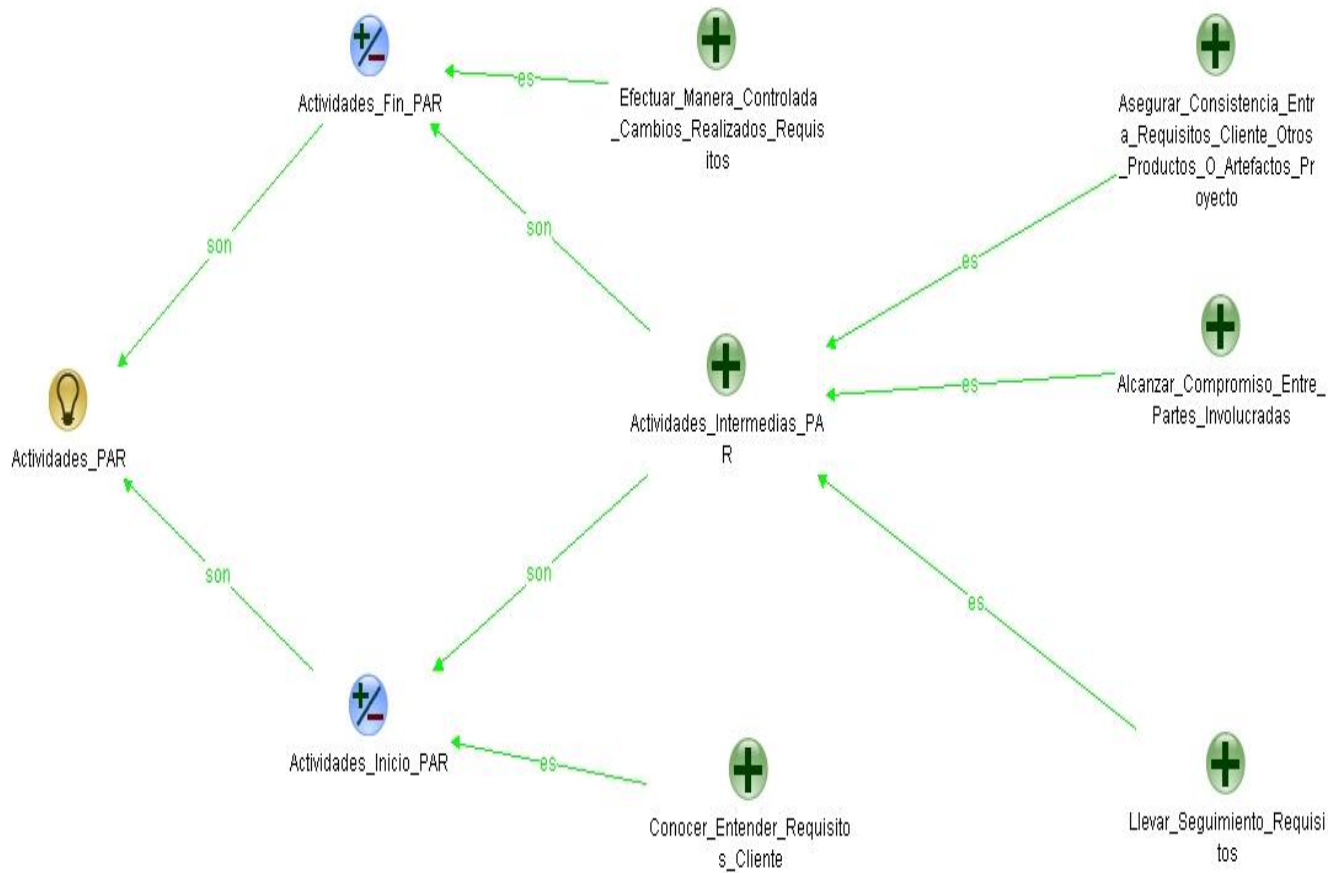


Figura 19: Actividades que se realizan en el Proceso de Administración de Requisitos.

El Proceso Base de Requisitos está compuesto por requisitos, tickets, artefactos, documentos, vista del sistema, entre otros (Figura 20).

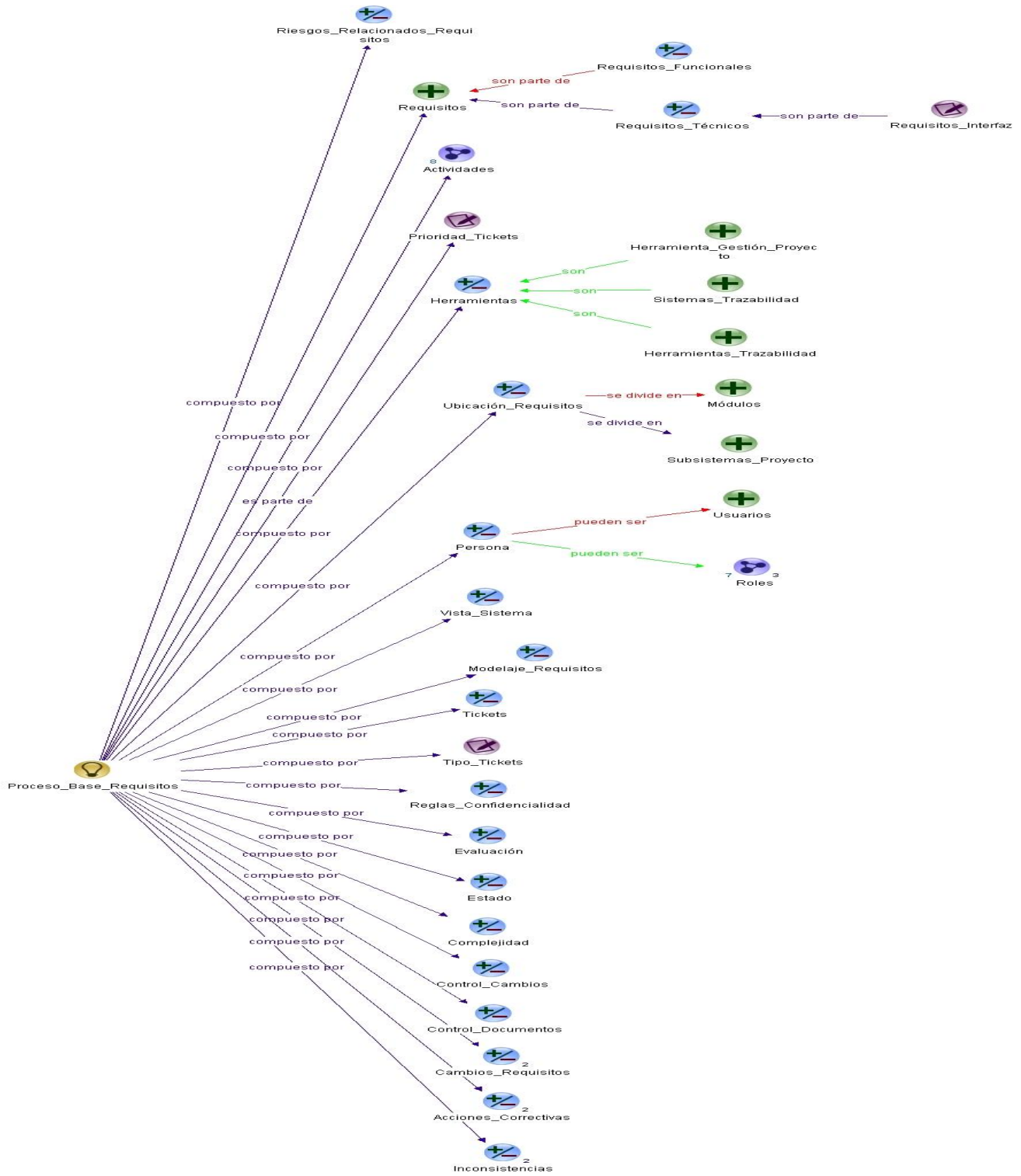


Figura 20: Proceso Base de Requisitos.

A continuación se muestran las actividades que forman parte del Proceso Base de Requisitos (Figura 21).

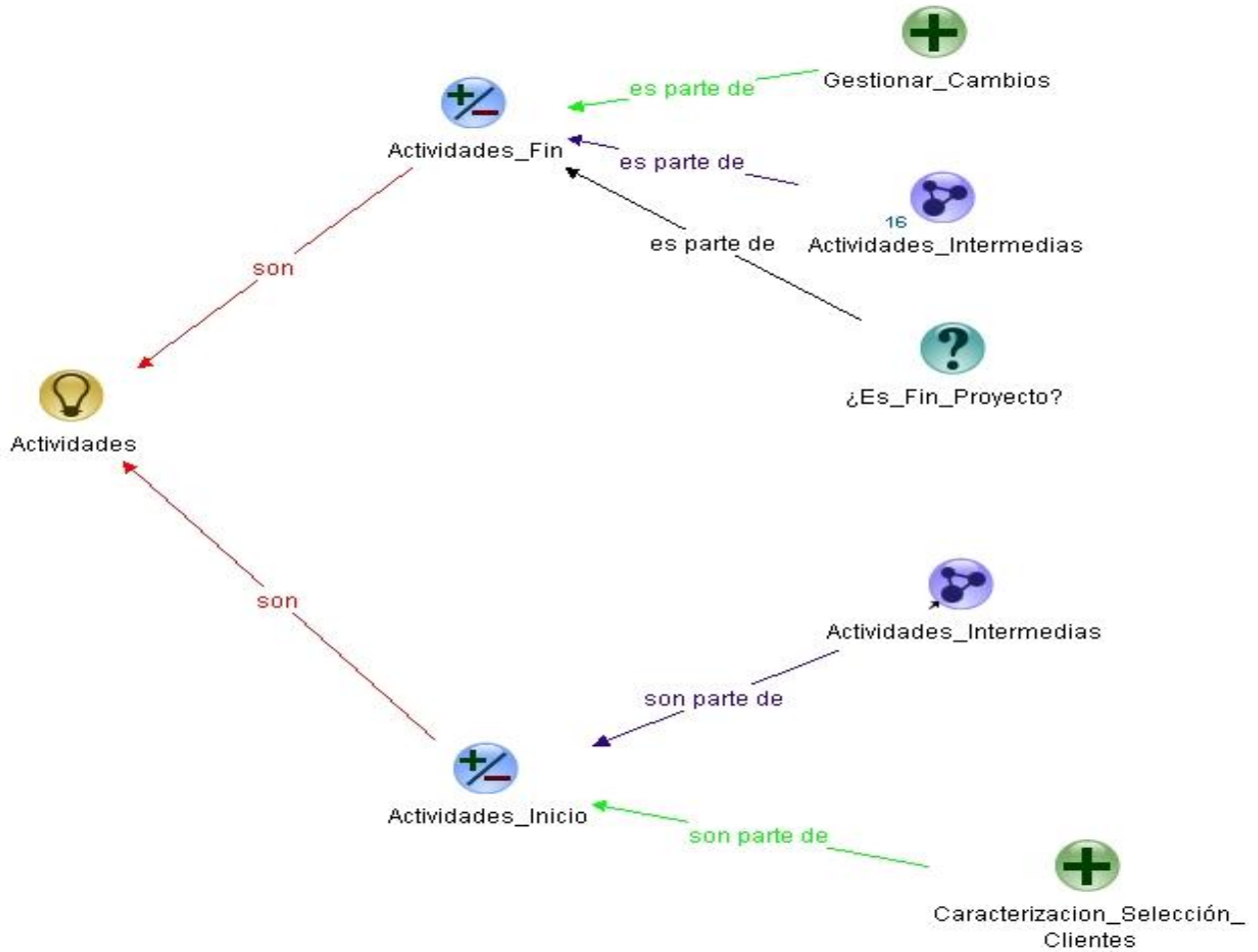


Figura 21: Actividades del Proceso Base de Requisitos.

Se muestran en la Figura 22 las actividades intermedias que forman parte de las actividades de fin y de inicio correspondientes a las actividades en general del Proceso Base de Requisitos.

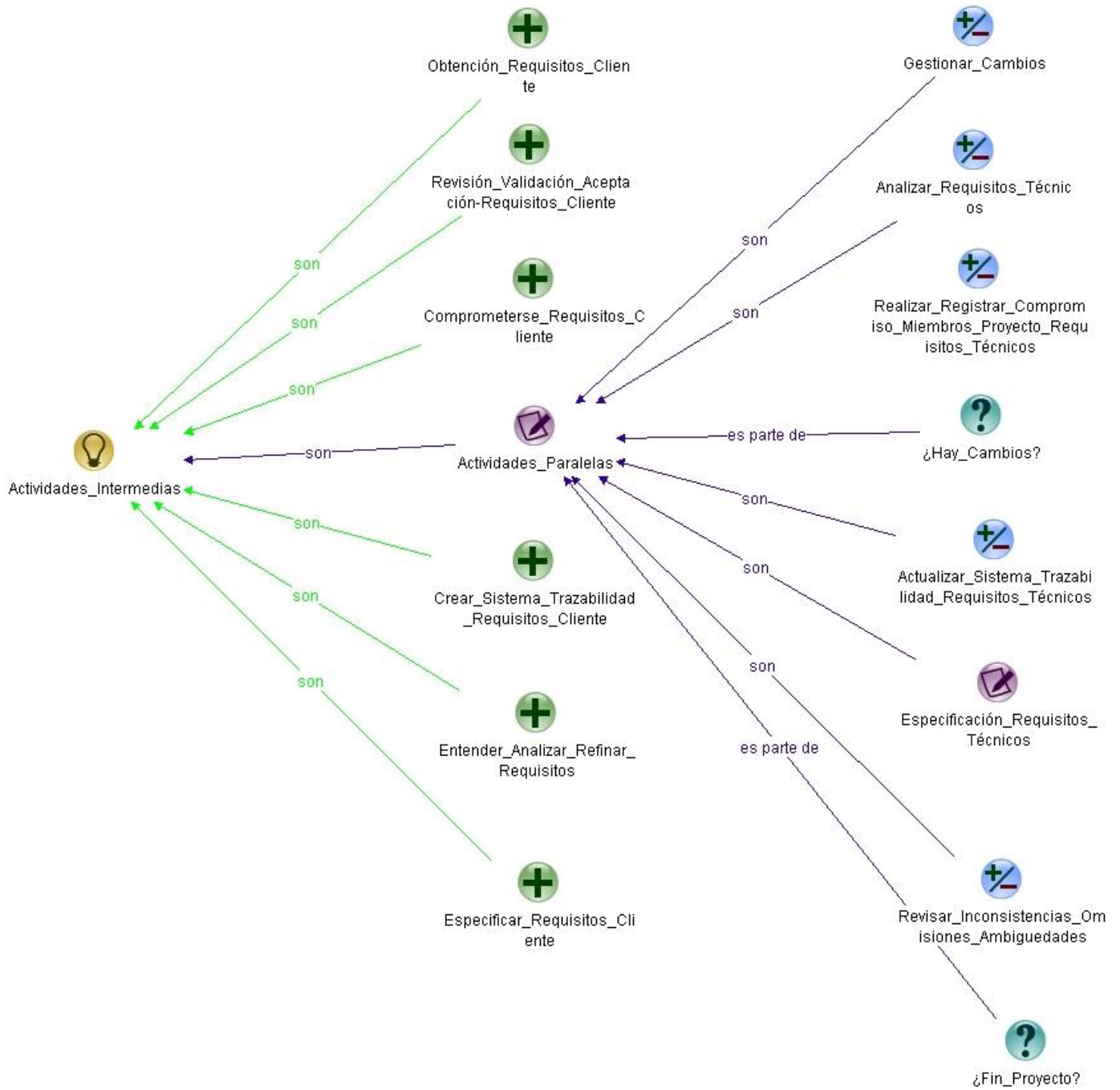


Figura 22: Actividades intermedias del Proceso Base de Requisitos.

Glosario de términos

Taxonomía: es la ciencia que se encarga de la categorización o clasificación de cosas basado en un sistema predeterminado.

Web: World Wide Web o simplemente Web, es un sistema de documentos enlazados unos con otros, accesibles a través de internet.

Herramienta: Aplicación empleada para la construcción (de ahí su nombre) de otros programas o aplicaciones.

Mapa conceptual: es una herramienta que permite organizar y representar, de forma gráfica y a través de un esquema, el conocimiento.

Estándares: Modelo o patrón.

Compleción: Acción y efecto de completar.

Viabilidad: Posibilidad de llevarse a cabo un plan o proyecto.