

UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO  
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS

TRABAJO FINAL DE GRADUACIÓN DE LA LICENCIATURA EN SISTEMAS DE INFORMACIÓN

**ONTOLOGÍA DE SOPORTE AL PROCESO DE INGENIERÍA DE REQUERIMIENTOS**

AUTOR: .....  
SILVIA VIVIANA MANZANEDO      RITA MARÍA ALEJANDRA ABELLEIRA

PROFESOR GUÍA: .....  
ING. MSc. DIANA PALLIOTTO

ASESOR: .....  
ING. MSc. GRACIELA ELISA BARCHINI

---

APROBADO EL DÍA ..... DEL MES DE ..... DEL AÑO 20.....

POR EL TRIBUNAL INTEGRADO POR

.....  
(firma)

.....  
(firma)

.....  
(firma)

.....  
(aclaración)

.....  
(aclaración)

.....  
(aclaración)

*A mis Padres, César y Nora, que son quienes guían mi vida.*

*A mi hermana Alejandra.*

*A mis dos soles, mis sobrinos Facundo y Ana.*

*Silvia Viviana Manzanedo*

*A mis padres, Juan y Alicia*

*A mi hermana Valeria*

*A mis abuelos, José y Zunilda*

*A toda mi familia*

*Rita María Alejandra Abelleira*

## *Agradecimientos*

*A nuestra Directora de Tesis Ing. Diana Palliotto por su presencia, por sus relevantes aportes, sugerencias y comentarios que ayudaron a la concreción de la presente investigación.*

*A nuestra Asesora de Tesis Ing. Graciela Barchini por su constante y desinteresado seguimiento y asistencia.*

S.V.M. y R.M.A.A.

*A mis padres Nora Guillermina Escobar y César Hipólito Manzanedo por su infinito amor, su apoyo incondicional, por la confianza depositada en mí a lo largo de toda mi vida y por enseñarme que no existen límites y que solo depende de mí lograr lo que me proponga.*

*A mi hermana Nora Alejandra Manzanedo por su ánimo y apoyo.*

*A mi amiga Carolina Ger por toda la ayuda brindada a lo largo del desarrollo del presente trabajo y por su gran amistad.*

*Y a todas aquellas personas que me acompañaron y contribuyeron de alguna manera u otra al desarrollo de la presente investigación.*

S.V.M

*Durante estos años de lucha constante para alcanzar uno de mis más grandes anhelos, me acompañaron personas que me brindaron su apoyo incondicional y su amor, por ello, es para mí un verdadero placer expresar mi agradecimiento a todas ellas:*

*A mis padres, por su incansable apoyo*

*A mi hermana, por acompañarme en todo momento*

*A mis abuelos, por su fe en mí*

*A mis tíos, por estar siempre*

R.M.A.A.

*Santiago del Estero, Argentina  
Agosto del 2010*

## CONTENIDO

<b>RESUMEN</b>	vii
<b>INTRODUCCION</b>	viii
<b>CAPITULO I. INTRODUCCIÓN, PLANTEO DEL PROBLEMA, SOLUCIÓN PROPUESTA</b>	1
<b>I.1. Introducción</b>	3
<b>I.2. Planteo del Problema</b>	3
<b>I.2.1. Planteo General del Problema</b>	4
<b>I.3. Alternativa de Solución</b>	4
<b>I.4. Objetivos del Proyecto</b>	5
<b>I.4.1. Objetivos Generales</b>	5
<b>I.4.2. Objetivos Específicos</b>	5
<b>I.5. Antecedentes</b>	6
<b>CAPITULO II. MARCOS REFERENCIALES</b>	
<b>II.1. Introducción</b>	11
<b>II.2. Marco Teórico</b>	11
<b>II.2.1. Ontologías</b>	11
<b>II.2.1.1. Definición</b>	11
<b>II.2.1.2. Estado del Arte</b>	11
<b>II.2.1.3. Principales ventajas de las ontologías</b>	12
<b>II.2.1.4. Objetivos de las ontologías</b>	13
<b>II.2.1.5. Componentes de las ontologías</b>	14
<b>II.2.1.6. Clasificación de las ontologías</b>	15
<b>II.2.1.7. Construcción de ontologías</b>	16
<b>II.2.2. Ingeniería de Requerimientos</b>	17
<b>II.2.2.1. La importancia de los requerimientos</b>	17
<b>II.2.2.2. Clasificación de requerimientos</b>	19
<b>II.2.2.3. Procesos de la IR</b>	21
<b>II.2.2.4. Técnicas utilizadas en la IR</b>	22
<b>II.2.2.5. Herramientas para la IR</b>	31
<b>II.2.3. Metodología DorCu para la Ingeniería de Requerimientos</b>	34
<b>II.2.3.1. Introducción</b>	34
<b>II.2.3.2. Propuesta metodológica para la Ingeniería de Requerimientos</b>	37
<b>II.2.3.2.1. Metodología e Ingeniería de Requerimientos</b>	37
<b>II.2.3.3. Metodología DorCu</b>	38
<b>II.2.3.3.1. Somera explicación de cada etapa</b>	40
<b>II.2.3.4. Gestión de calidad de procesos</b>	46
<b>II.2.3.4.1. Atributos</b>	47
<b>II.2.3.5. Conclusiones</b>	47

<b>II.3. Marco Metodológico</b>	48
<b>II.3.1. Methontology</b>	48
<b>II.3.2. Adquisición de Conocimientos</b>	52
<b>II.3.2.1. Proceso de adquisición de conocimientos</b>	53
<b>II.3.2.2. Extracción de conocimientos</b>	53
<b>II.3.3. Marcos</b>	56
<b>II.3.4. Metodología de trabajo propuesta</b>	58
<b>II.3.5. Entorno de desarrollo. Herramientas para construcción de ontologías</b>	62
<b>II.3.5.1. Protégé</b>	62
<b>II.3.5.2. Protégé OWL</b>	63
<b>II.3.5.3. Lenguaje OWL</b>	63
<b>II.3.5.3.1. Los tipos de OWL</b>	64
<b>II.3.5.4. Razonamiento de una ontología. Herramienta Racer.</b>	65
<b>II.3.6. Conclusión</b>	65
<b>CAPITULO III. ESPECIFICACION DE REQUERIMIENTOS</b>	
<b>III.1. Introducción</b>	69
<b>III.2. Descripción del dominio de la ontología</b>	69
<b>III.3. Definición de los objetivos de la ontología</b>	70
<b>III.3.1. Objetivos de OntoIR</b>	70
<b>III.4. Especificación de requisitos de la ontología</b>	70
<b>III.5. Adquisición de conocimiento</b>	72
<b>III.5.1. Análisis estructural de textos</b>	72
<b>III.6. Conclusión</b>	85
<b>CAPITULO IV. CONCEPTUALIZACION DE OntoIR</b>	
<b>IV.1. Introducción</b>	89
<b>IV.2. Taxonomía de conceptos</b>	89
<b>IV.3. Diagramas de relaciones binarias</b>	95
<b>IV.4. Tabla de clase- subclase- propiedad</b>	97
<b>IV.5. Tabla de propiedades en detalle</b>	100
<b>IV.6. Tabla de reglas</b>	101
<b>IV.7. Tabla de axiomas</b>	106
<b>IV.8. Diagramas de clases</b>	116
<b>IV.9. Esquema conceptual</b>	121
<b>IV.10. Conclusión</b>	122
<b>CAPITULO V. FORMALIZACION DE OntoIR</b>	
<b>V.1. Introducción</b>	129
<b>V.2. Marcos</b>	130
<b>V.3. Ontología basada en marcos</b>	130
<b>V.4. Jerarquía de marcos</b>	132
<b>V.5. Conclusión</b>	137

<b>CAPITULO VI. CONSTRUCCIÓN, PRUEBA Y VALIDACION DE OntoIR</b>	
<b>VI.1. Introducción</b>	141
<b>VI.2. Construcción</b>	141
<b>VI.3. Prueba y Validación</b>	141
<b>VI.3.1. Objetivos formulados</b>	141
<b>VI.4. Verificación de objetivos</b>	142
<b>VI.4.1. Verificación del objetivo a)</b>	142
<b>VI.4.2. Verificación de los objetivos b) y d)</b>	143
<b>VI.4.2.1. Análisis de la ejecución de los casos de prueba</b>	150
<b>VI.4.3. Verificación del objetivo c)</b>	150
<b>VI.5. Conclusión</b>	151
<b>CONCLUSION FINAL</b>	153
<b>BIBLIOGRAFIA</b>	157
<b>ANEXO A. PRUEBA Y VALIDACION</b>	167
<b>ANEXO B. MANUAL DE USUARIO</b>	181
<b>ANEXO C. ENTREVISTAS</b>	191

## RESUMEN

En los últimos años ha ganado reconocimiento la importancia de la Ingeniería de Requerimientos (IR) y los riesgos en que se incurren si ésta se realiza en forma incompleta o incorrecta.

La IR cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas. La importancia de esta fase es esencial puesto que los errores más comunes y costosos de reparar, así como los que más tiempo consumen, se deben a una inadecuada IR.

Reconociendo los problemas que acarrea una IR incorrecta, y teniendo en cuenta los beneficios de usar ontologías, se propone la creación de *OntoIR, Ontología de Soporte al Proceso de Ingeniería de Requerimientos*, la cual asistirá a desarrolladores durante la ejecución del mismo.

Dicha ontología estará basada en la metodología DorCu (Documentación de Requerimientos Centrada en el Usuario) que considera los mejores resultados de distintos enfoques, y se apoya en métodos, técnicas y herramientas ya desarrollados; además tiende a unificar la terminología empleada en el campo de la IR.

Para la construcción de la ontología se utilizará el entorno de desarrollo de Protégé 2000 y el lenguaje OWL.

**Palabras Claves:** Ontologías, Ingeniería de Requerimientos, Procesos, DorCu.

## INTRODUCCION

La Ingeniería de Requerimientos (IR) cumple un papel primordial en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas. Cada una de las actividades involucradas en la IR se considera como crítica dentro del desarrollo de sistemas. Los errores detectados dentro de las primeras etapas del proceso de desarrollo pueden tener amplias repercusiones en fases posteriores. Para que un esfuerzo de desarrollo de software tenga éxito, es esencial comprender perfectamente los requerimientos. [1]

Las ontologías son herramientas conceptuales y técnicas que permiten especificar, estructurar y comunicar el conocimiento de un dominio determinado. Están llegando a ser una herramienta fructífera y están desarrollándose y aplicándose en una variedad de áreas emergentes tales como: modelado de empresas, diagnóstico, toma de decisiones, modelado de procesos y sistemas. [12]

En este proyecto se diseña una ***Ontología de Soporte al Proceso de Ingeniería de Requerimientos***.

En rasgos generales, la función de la Ontología de Soporte es brindar a los desarrolladores las alternativas de solución más probables en base a las características que presenta el proyecto para el cual se deben determinar los requerimientos.

La estructura del presente trabajo se basa, principalmente en la metodología utilizada para el diseño, el desarrollo y la evaluación de ontologías de aplicación.

En el Capítulo I se realiza una descripción de la problemática existente en el dominio de interés y se plantea la alternativa de solución propuesta para la problemática encontrada.

En el Capítulo II se desarrollan los Marcos Teórico y Metodológico. En el primero, se definen conceptos referentes a Ontologías, Ingeniería de Requerimientos, técnicas y herramientas utilizadas para la misma y Metodología DorCu. En el segundo, se define la metodología para la construcción de ontologías Methontology, la metodología para la adquisición de conocimiento (Análisis



Estructural de Textos), el sistema de formalización con Marcos, el lenguaje OWL y la Metodología de trabajo propuesta para la realización del proyecto.

En el Capítulo III se realiza la descripción del dominio, los objetivos y la construcción del Documento de Requisitos de OntoIR. Además, se construyen los glosarios integrados de términos aplicando las técnicas definidas con anterioridad.

En el Capítulo IV se define la jerarquía de los conceptos del dominio de OntoIR quedando formada la taxonomía de los mismos. Se diseñan los Diagramas de Relaciones Binarias de los conceptos que componen la taxonomía. Se construye la Tabla de Clase\_Subclase\_Propiedad que contiene las características de las clases de la ontología, y la Tabla de Propiedades de Detalle en donde se especifica el nombre, conceptos fuente y destino, cardinalidad y relación inversa de cada una de las propiedades definidas. Además, en una Tabla de Reglas, se definen las reglas necesarias para la construcción de OntoIR. A partir de esas reglas, se formulan los axiomas necesarios para obtener la información referente al dominio de la ontología. Se diseñan los Diagramas de Clases correspondientes a los procesos involucrados en la Ontología. Por último, se construye el Esquema Conceptual en el se muestran las clases y subclases principales de la ontología, las relaciones existentes entre ellas y la cardinalidad de dichas relaciones.

En el Capítulo V se realizó la Jerarquía de Marcos, utilizando tres jerarquías correspondientes a cada una de las clases principales de OntoIR.

En el capítulo VI se realiza una descripción de la construcción, la prueba y la validación de OntoIR, se diseñan diferentes casos de prueba para analizar sus resultados, y se elabora un cuestionario dirigido a expertos; esto permite verificar si se cumplen los objetivos definidos al inicio.

# *Capítulo I:*

*Introducción,  
Planteo del problema  
Solución Propuesta*



## **I.1.INTRODUCCION**

En este capítulo se realiza una descripción de la problemática existente durante el Proceso de Ingeniería de Requerimientos (IR) y se plantea la alternativa de solución propuesta para la problemática descripta.

Lo que se pretende conseguir con la solución propuesta es desarrollar una herramienta que sirva de soporte para el desarrollo de software, dicha herramienta asistirá al experto en la fase de IR, para conseguir especificaciones lo más correctas posible, con el fin de desarrollar sistemas más confiables.

## **I.2. PLANTEO DEL PROBLEMA**

La etapa de IR cumple un papel primordial dentro del proceso de desarrollo de software; en ella se reconocen los siguientes problemas al momento de su aplicación:

- No existe una única técnica estandarizada y estructurada que ofrezca un marco de desarrollo que garantice la calidad del resultado del proceso de IR.

Entre las técnicas que se encuentran para las diferentes etapas del proceso de IR se encuentran: [26]

- Para la captura de requisitos: entrevistas, JAD (Joint Application Development / Desarrollo conjunto de aplicaciones), Brainstorming (Tormenta de ideas), Mapas Conceptuales, Casos de uso, Cuestionarios y Listas de verificación, Comparación de Terminología, etc.
  - Para la definición de requisitos: Lenguaje Natural, Glosario y Ontologías, Plantillas o Patrones, Escenarios, Casos de Uso, Lenguajes Formales, etc.
  - Para Validación de requisitos: Revisiones o Seguimientos, Auditorías, Matrices de Trazabilidad, Prototipos, etc.
- Resulta difícil establecer los criterios para seleccionar las técnicas apropiadas a usar en cada etapa del proceso. Entre estos criterios pueden considerarse la facilidad de aprendizaje y de uso, la escalabilidad, el costo y completitud de los resultados y el tiempo requerido para aplicar las técnicas.
  - Cada técnica puede aplicarse en una o más actividades de la IR. En la práctica, la técnica más apropiada para cada actividad dependerá del proyecto que esté desarrollándose. [3]

- Existe una multiplicidad de términos. Diferentes propuestas de trabajo y técnicas que trabajan con iguales conceptos pero la terminología usada para nombrarlos es diferente.
- Existen diferentes propuestas metodológicas que definen el tratamiento de requerimientos de diferentes formas siendo algunas no muy detalladas a la hora de describir el proceso de definición de requisitos, las técnicas a aplicar, los tipos de requerimientos que consideran o los resultados a obtener.  
Entre las propuestas metodológicas están: Oliver and Steiner 1996, CMM nivel Repetitivo, EIA / IS-632, IEEE std. 1220-1994, RUP, entre otras. [3]

### I.2.1. PLANTEO GENERAL DEL PROBLEMA

*En el ámbito de la IR, existen diversas metodologías y técnicas no estandarizadas, algunas de ellas, no explícitas y con terminología no unificada, que dificultan las tareas específicas a realizar por los desarrolladores de software.*

### I.3. ALTERNATIVA DE SOLUCION

La IR es una disciplina para la cual no existe aún un consenso general sobre la definición exacta de los conceptos y la terminología a emplear, así como sobre qué métodos, metodologías y técnicas son más apropiadas para emplear en cada caso. Es un proceso complejo, que debe llevarse a cabo en forma eficiente, para la obtención de un software de calidad.

Puesto que las ontologías son útiles, entre otras cosas, para el modelado de procesos [42], se propone como solución al problema planteado, como se mencionó anteriormente, la construcción de una **ONTOLOGÍA DE SOPORTE AL PROCESO DE INGENIERÍA DE REQUERIMIENTOS**, que sirva de apoyo a los desarrolladores de software. Dicha ontología contendrá información basada en la metodología DorCu, ya que ésta cuenta con los beneficios de considerar los mejores resultados de distintos enfoques y se apoya en diversos métodos, técnicas y herramientas ya desarrollados; además tiende a unificar la terminología empleada en el campo de la IR.

La ontología contendrá las etapas del proceso de IR indicadas en la Metodología DorCu, tales como:

- Elicitación de requerimientos.
- Análisis de requerimientos.
- Especificación de requerimientos.
- Validación y certificación de requerimientos.

Además contendrá información concerniente a las herramientas (por ejemplo: Herramientas CASE), técnicas (puntos de vista, entrevistas, etc.) y demás recursos a emplear en cada una de las etapas, lo que permitirá realizar una selección e integración de las más adecuadas.

La construcción de la ontología estará basada en la metodología Methontology; para la creación de la misma se usará la herramienta Protege 2000, y OWL para la composición de las reglas.

## **I.4. OBJETIVOS DEL PROYECTO**

### **I.4.1. Objetivos Generales**

- Brindar un marco común para llevar a cabo las actividades de IR.
- Facilitar la práctica de la IR en el entorno de los desarrolladores de software.
- Contribuir al mejoramiento de la toma de decisiones de los desarrolladores.
- Estimular y facilitar el reuso de conocimiento.
- Contribuir a la difusión del uso de ontologías en la Ingeniería de Software.

### **I.4.2. Objetivos Específicos**

- Lograr un entendimiento compartido entre los desarrolladores de un producto software en las etapas del Proceso de IR.
- Orientar a los desarrolladores en la elección y el uso de los distintos métodos y técnicas competentes a la IR.
- Ofrecer una herramienta que garantice la calidad del proceso de IR, y que disminuya los riesgos en que se incurren cuando se lleva a cabo de forma inadecuada.
- Asistir a desarrolladores en el análisis y la especificación de los requerimientos durante el proceso de desarrollo del software.

## **I.5. ANTECEDENTES**

La investigación con respecto a los antecedentes que se relacionan con la propuesta de la Ontología para la IR concluye con la revelación de que no existe una propuesta similar a la planteada en el presente trabajo. Se observó que existen diferentes emprendimientos de desarrollos de ontologías, metodologías, etc. que favorecen a los diferentes procesos involucrados en la Ingeniería de Software, pero ninguno plantea el desarrollo de una ontología para el Proceso de IR.

A continuación se detallan los trabajos encontrados que se relacionan con el presente proyecto:

- **Metodología basada en ontología para la detección de errores en el diseño de software** [7]

Esta metodología utiliza una ontología para la especificación y el diseño de software como un modelo de información común para el conocimiento de la especificación y diseño.

- **Ontología para la Ingeniería de Software** [8]

El proyecto SWEBOK tiene como meta construir una ontología para el campo de la Ingeniería de Software, usando el conocimiento disponible en la Guía SWEBOK (así como con conocimiento científico de fuentes tales como IEEE e ISO).

- **Proceso de modelado para Programación Extrema basado en ontología** [9]

La ontología para la programación extrema es una especificación de un modelo formal de los principales conceptos usados en la programación extrema y sus propiedades.

- **Ontología para Fábricas de Software** [10]

Se pretende que esta ontología permita entender las diferentes acepciones del concepto Fábricas de Software (FS), así como las características, innovaciones e

implicaciones de las visiones mas recientes de FS en el ámbito de la Ingeniería de Software.

- **Ontología de la Gestión de proyectos de mantenimiento de software [11]**

Esta ontología incluye los factores que influyen en el mantenimiento de software; además indica un conjunto de aspectos dinámicos y estáticos del dominio, necesarios para gestionar un proyecto de mantenimiento de software.

## *Capítulo II:*

### *Marcos Referenciales*





## **II.1. INTRODUCCION**

La IR es una disciplina que integra métodos, herramientas y procedimientos para el desarrollo del software de computadora.

En este capítulo se exponen los lineamientos teóricos, metodológicos y empíricos sobre las ontologías, como una herramienta de soporte, y la IR como una disciplina primordial dentro del proceso de producción de software.

En el marco teórico se detalla información referente a ontologías, clasificación y sus ventajas; y con respecto a la IR, su importancia, principales etapas, técnicas y herramientas para su aplicación.

En el marco metodológico se describe la metodología que se siguió en el presente trabajo, Metontology, como marco de referencia para la construcción de la ontología, la herramienta Protégé y el lenguaje owl para su implementación.

## **II.2. MARCO TEORICO**

### **II.2.1. ONTOLOGIAS**

#### **II.2.1.1. Definición**

Cuando se trata de definir los conocimientos de un dominio, el uso de ontologías puede ser de utilidad. Ontología, etimológicamente, proviene del griego “ontos” (ser) y “logos” (tratado o estudio); es decir, ontología es el estudio del ser y sus propiedades esenciales.

Una ontología es una especificación formal explícita de una conceptualización compartida. El término “Conceptualización” se refiere a una abstracción, una visión simplificada del mundo que se desea representar, construido a partir de la identificación de los conceptos y relaciones relevantes.

Formal significa que debe ser comprensible para una computadora, por lo que no puede escribirse sólo en lenguaje natural. [43]

#### **II.2.1.2. Estado del Arte**

En la Tesis Doctoral “Modelo para la Creación de Entornos de Aprendizaje basados en técnicas de Gestión del Conocimiento”, presentada por Inés Friss de Kereki Guerrero, se hace referencia a varios autores que dan una definición de ontologías. [43]

Neches y otros, señalan que una ontología define el vocabulario de un área mediante un conjunto de términos básicos y relaciones entre dichos términos, así como las

reglas para combinar los términos y las relaciones para definir extensiones al vocabulario. Para Van Heijst y otros, una ontología es una especificación explícita a nivel de conocimiento de una conceptualización. Es un conjunto de distinciones que son significativas para un agente.

Para Gutiérrez, “Ontología es teoría de los objetos. La ontología de la ciencia, o de una ciencia en particular, es la lista de objetos cuya existencia tiene el científico que postular para poder practicar su disciplina. Una ontología es la colección de entes (objetos) a que se refieren los enunciados de cada disciplina particular. El tema de la ontología es el tema de *lo que hay*.”

Para Rodríguez-Artacho, una ontología es una representación explícita de una conceptualización cognitiva, es decir, la descripción de los componentes de conocimiento relevantes en el ámbito de la modelización.

Toledo y otros, refieren que para la filosofía, ontología es el estudio de la existencia del ser. Citando a Gruber, indican que las definiciones deben ser objetivas y preferiblemente completas, coherentes, extensibles (que se puedan definir nuevos conceptos a partir de los existentes) y tener un mínimo compromiso con la implementación.

Jasper y otros, señalan que una ontología puede tomar una variedad de formas, pero necesariamente deberá incluir un vocabulario de términos y alguna especificación de su significado. Esto incluye definiciones y una indicación de cómo los conceptos están interrelacionados. Las ontologías se usan para mejorar la comunicación entre personas o computadoras.

### **II.2.1.3. Principales ventajas de las ontologías**

Actualmente, la principal ventaja que aportan las ontologías, desde el punto de vista de las bases de datos y de la recuperación de información, tiene que ver con el empleo simultáneo de bases de datos (de muy distinta naturaleza, características y formato) a las que poder interrogar simultáneamente para recuperar la información buscada. Estas bases de datos pueden tener contenido tan dispar como documentación textual poco estructurada (la clásica que forma parte de los buscadores en Internet), documentación fotográfica, documentación geográfica, museográfica, urbanística, espacios naturales, etc.

Otra de las grandes ventajas de las ontologías es la posibilidad de utilizar la información preexistente en dichas bases de datos a través de la propia ontología sin

tener que renunciar a la base de datos original, de manera que pueden convivir y seguir empleándose simultáneamente bases de datos iniciales y ontología; tan sólo se debe tener presente la necesidad de actualizar la ontología con los nuevos datos que vayan añadiéndose a las bases de datos.

Tampoco es necesario introducir de nuevo manualmente -aunque podría hacerse- las bases de datos en la nueva estructura que hemos desarrollado con la ontología. Se puede automatizar el proceso de vuelco de la información mediante el desarrollo de programas que transformen el formato de la base de datos inicial en el formato y lugar adecuado de la nueva estructura ontológica. Lógicamente, cuanto más estructurada se encuentre la información inicial, más sencilla será la transformación correspondiente. [44]

#### **II.2.1.4. Objetivos de las ontologías**

Algunos de los objetivos que se tienen para desarrollar ontologías son los siguientes:

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

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

- *Permitir reutilizar el dominio de conocimiento.*

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

- *Permitir separar conocimiento de dominio del conocimiento operacional.*

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

- *Analizar el conocimiento del dominio.*

Específicamente en lo que se refiere al estudio de los términos y relaciones que lo configuran, ya sea formalmente o no. [45]

### II.2.1.5. Componentes de las ontologías

Los componentes de una ontología varían de acuerdo al dominio de interés y a las necesidades de los desarrolladores. Por lo general, entre los componentes se encuentran los siguientes [46]:

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

### II.2.1.6. Clasificación de las ontologías

Según el grado de generalidad o nivel de dependencia a una tarea o visión en particular, Mizoguchi y colegas presenta la siguiente clasificación de ontologías [47]:

- **Ontologías de Alto Nivel (genéricas):** describen conceptos muy generales como espacio, tiempo, materia, objeto, evento, acción, etc., que son independientes de un problema o dominio en particular. Por lo tanto, parece razonable, al menos en teoría, tener ontologías unificadas de alto nivel para grandes comunidades de usuarios.
- **Ontologías de Dominio y Ontologías de Tarea:** describen, respectivamente, el vocabulario relacionado a un dominio genérico (como medicina o automóviles) o una tarea o actividad genérica (diagnóstico o venta), mediante la especialización de los términos introducidos en la ontología de alto nivel.
- **Ontologías de Aplicación:** describen conceptos que dependen tanto de un dominio como de una tarea en particular, los cuales frecuentemente son especializaciones de ambas ontologías. A menudo, estos conceptos corresponden a los roles desempeñados por entidades del dominio mientras realizan cierta actividad. Contienen conocimiento esencial para modelar una aplicación particular bajo consideración.

Van Heist, propone una clasificación de las ontologías de acuerdo con la cantidad y tipo de la conceptualización. Así se pueden diferenciar los siguientes tipos de ontologías:

- **Terminológicas:** especifican los términos que son usados para representar conocimiento en el universo del discurso.
- **De Información:** especifican la estructura de almacenamiento de bases de datos.
- **De modelado del conocimiento:** especifican conceptualizaciones del conocimiento.

Según el alcance de aplicabilidad de la ontología, se pueden distinguir los siguientes tipos:

- **Ontologías de dominio:** son específicas para un dominio en concreto.
- **Ontologías de tareas:** representan las tareas que son susceptibles de realizar en un dominio en concreto.

- **Ontologías Generales:** representan los datos generales y no específicos de un dominio.

### II.2.1.7. Construcción de ontologías

En la Tesis Doctoral “Modelo para la Creación de Entornos de Aprendizaje basados en técnicas de Gestión del Conocimiento”, presentada por Inés Friss de Kereki Guerrero [43], se hace referencia a diferentes criterios y ventajas en la construcción de ontologías presentados por diferentes autores.

Gruber señala como principios metodológicos para construir ontologías la claridad y objetividad, la completión, la coherencia y el maximizar la ampliación monótona. Este último principio quiere decir que deberían incluirse en la ontología términos nuevos, tanto generales como especializados, de tal manera que no requieran la revisión de las definiciones existentes.

Crear una ontología tiene como ventaja, indica Rodríguez-Artacho que hace explícita la categorización de los elementos y relaciones que intervienen en el modelo de conocimiento, de forma que el modelo se pueda editar y gestionar, y sea posible transmitirlo.

La construcción de una ontología involucra la definición de conjuntos de clases, relaciones, funciones, objetos constantes y axiomas, según Toledo y otros. Hay varias propuestas de lenguajes para describir ontologías, algunos ejemplos son KIF (Knowledge Interchange Format) y Ontolingua, desarrollado por el Departamento de Ciencia de la Computación de la Universidad de Stanford.

Hay varias metodologías para construir ontologías. A partir de la descripción del estándar de IEEE 1074-1995 sobre el proceso de desarrollo de software, se establece que es posible aplicar dicho estándar al desarrollo de ontologías pues son parte de los productos de software. Los criterios utilizados para comparar las metodologías son: herencia de ingeniería del conocimiento, detalle de la metodología, recomendaciones para formalizar el conocimiento, estrategias para construir ontologías (dependiente, semidependiente o independiente de la aplicación), estrategia para identificar conceptos, ciclo de vida recomendado, diferencias con el estándar IEEE 1074-1995, técnicas recomendadas y qué ontologías se han desarrollado utilizándolas. Las metodologías comparadas son, entre otras: Enterprise Ontology de Uschold y otros, la de A. Bernaras, Methontology de Gómez A. y Sensus de Swartout, B. y otros. Las conclusiones de su estudio es que ninguna de las metodologías era complemente madura, aunque la más madura resultó ser Methontology. [43]

## II.2.2. INGENIERÍA DE REQUERIMIENTOS

La consecuencia de la Ingeniería de Sistemas (IS) es la especificación de un sistema o producto basado en computadora. Pero el desafío de la IS, y de los ingenieros del software, es importante: ¿cómo podemos asegurar que hemos especificado un sistema que recoge las necesidades y expectativas del cliente. No hay una respuesta segura a esta difícil pregunta, pero un sólido proceso de IR es la mejor solución de que disponemos actualmente. [13]

Entendiendo la IR como el proceso de descubrimiento y comunicación de las necesidades de los clientes y usuarios, y la gestión de los cambios de dichas necesidades [16], uno de los aspectos más importantes de ella es la comunicación, característica esta que vuelve el proceso complejo por la alta presencia del factor humano que contiene y es la responsable de que la disciplina contenga aspectos sociales y culturales y no sólo de índole técnica. [17]

Además, la IR debe considerarse como un proceso de construcción de una especificación de requisitos en el que se avanza desde unas especificaciones iniciales, que no poseen las características oportunas, hasta especificaciones finales completas, formales y acordadas entre todas las partes. [18]

Se adopta una definición para la IR que se apoya en definiciones de los autores Leite [21], Loucopoulos [22], Dorfman [24] y de la IEEE [23] y en los problemas que aún quedan por resolver: se define a la IR como un proceso centrado en el cliente y usuario y sus necesidades, en donde las etapas de elicitación, de análisis, de especificación, validación y certificación de requerimientos iteran hasta la obtención de documentos isomórficos representativos de las necesidades reales de clientes / usuarios, depuradas en base a procesos cooperativos que se llevan a cabo en distintas comunidades del dominio de información y en donde el isomorfismo de los documentos finales permite salvar la brecha que existe entre el enfoque antropológico de la IR y las etapas siguientes de la Ingeniería de Software, permitiendo que las mismas continúen hacia la prosecución de un software sin fallos. [34]

### II.2.2.1. La importancia de los requerimientos

La IR es una tarea dificultosa. No se trata sólo de escribir lo que el cliente dice que quiere. Un problema fundamental en negocios es que los requerimientos son inherentemente dinámicos. Una práctica inefectiva de requerimientos es un gran

problema. Para proyectos más exitosos son necesarios un desarrollo y una administración más disciplinados. Un requerimiento es un atributo necesario en un sistema, una definición que identifica una capacidad, característica, o factor de calidad de un sistema que tiene valor y utilidad para un cliente o usuario. Los requerimientos son importantes porque ellos proveen las bases para todo el trabajo de desarrollo siguiente.

Identificar los requerimientos reales requiere un proceso de requerimientos interactivo e iterativo, soportado por prácticas efectivas, procesos, mecanismos, métodos, técnicas y herramientas. [20]

Los principales beneficios que se obtienen de la IR son [13]:

- Permite gestionar las necesidades del proyecto en forma estructurada: cada actividad de la IR consiste de una serie de pasos organizados y bien definidos.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: la IR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
- Disminuye los costos y retrasos del proyecto: muchos estudios han demostrado que reparar errores no descubiertos a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la especificación de requerimientos.
- Mejora la calidad del software: la calidad en el software tiene que ver con cumplir un conjunto de requerimientos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc)
- Mejora la comunicación entre equipos: la especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.
- Evita rechazos de usuarios finales: la IR obliga al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema, por lo que se involucra durante todo el desarrollo del proyecto.



### II.2.2.2. Clasificación de requerimientos

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Algunos problemas que surgen durante el proceso de IR son resultado de no hacer una clara separación entre estos diferentes niveles de descripción. Se distinguen utilizando la denominación *requerimientos del usuario* para designar los requerimientos abstractos de alto nivel, y *requerimientos de sistema* para designar la descripción detallada de lo que el sistema debe hacer. Los requerimientos del usuario y del sistema se pueden definir como sigue [3]:

- *Requerimientos del usuario*: son declaraciones, en lenguaje natural y en diagramas, de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar.
- *Requerimientos del sistema*: establecen con detalle las funciones, los servicios y las restricciones operativas del sistema.

A menudo, los requerimientos de sistema se clasifican en funcionales y no funcionales, o como requerimientos de dominio:

- *Requerimientos funcionales*: son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, también pueden declarar explícitamente lo que el sistema no debe hacer.

Los requerimientos funcionales de un sistema describen lo que el sistema debe hacer. Estos dependen del tipo de software que se desarrolle, de los posibles usuarios del software y del enfoque general tomado por la organización al redactar requerimientos.

La especificación de requerimientos funcionales de un sistema debe estar completa y ser consistente. La completitud significa que todos los servicios solicitados por el usuario deben estar definidos. La consistencia significa que los requerimientos no deben tener definiciones contradictorias. Para sistemas grandes y complejos, es prácticamente imposible alcanzar los requerimientos de consistencia y completitud.

*Requerimientos no funcionales*: son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre el proceso de desarrollo y estándares. A menudo se aplican al sistema en su totalidad.

Los requerimientos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento.

Estos requerimientos rara vez se asocian con características particulares del sistema. Más bien estos requerimientos especifican o restringen las propiedades emergentes del sistema. A menudo, son más críticos que los requerimientos funcionales particulares.

Los requerimientos no funcionales no sólo se refieren al sistema software a desarrollar. Algunos de estos requerimientos pueden restringir el proceso que se debe utilizar para desarrollar el sistema.

Los requerimientos no funcionales pueden venir de las características requeridas del software (requerimientos del producto), de la organización (requerimientos organizacionales) o de fuentes externas.

Los requerimientos no funcionales pueden ser:

- Requerimientos del producto: especifican el comportamiento del producto
- Requerimientos organizacionales: se derivan de políticas y procedimientos existentes en la organización del cliente y en la del desarrollador.
- Requerimientos externos: este gran apartado incluye todos los requerimientos que se derivan de los factores externos al sistema y de su proceso de desarrollo.

Siempre que sea posible, se deben redactar los requerimientos funcionales de manera cuantitativa para que se puedan probar de modo objetivo.

- *Requerimientos de dominio*: son requerimientos que provienen del dominio de aplicación del sistema y que reflejan las características y restricciones de ese dominio. Pueden ser funcionales o no funcionales.

Se derivan del dominio de aplicación del sistema más que de las necesidades específicas de los usuarios. Normalmente incluyen terminología especializada

del dominio o referencias a conceptos del dominio. Pueden ser requerimientos funcionales nuevos, restringir los existentes o establecer cómo se deben ejecutar cálculos particulares.

### **II.2.2.3. Procesos de la IR**

La meta de la IR es crear y mantener un documento de requerimientos del sistema. El proceso general corresponde a cuatro subprocesos de alto nivel de la IR. Estos tratan de la evaluación de si el sistema es útil para el negocio (estudio de viabilidad); el descubrimiento de requerimientos (obtención y análisis); la transformación de estos requerimientos en formularios estándar (especificación); y la verificación de que los requerimientos realmente definen el sistema que quiere el cliente (validación).

#### **1)-Estudio de viabilidad**

El proceso de IR debería empezar con un estudio de viabilidad. La entrada de éste es un conjunto de requerimientos preliminares, una descripción resumida del sistema y de cómo éste pretende contribuir a los procesos de negocio. Los resultados del estudio de viabilidad deberían ser un informe que recomiende si merece o no la pena seguir con la IR y el proceso de desarrollo del sistema.

#### **2)-Obtención y análisis de requerimientos**

La siguiente etapa del proceso de IR es la obtención y el análisis de requerimientos. En esta actividad, los ingenieros de software trabajan con los clientes y los usuarios finales del sistema para determinar el dominio de la aplicación, qué servicios debe proporcionar el sistema, el rendimiento requerido del sistema, las restricciones de hardware, etc.

#### **3)-Validación de requerimientos**

La validación de requerimientos trata de mostrar que éstos realmente definen el sistema que el cliente desea. Coincide parcialmente con el análisis ya que éste implica encontrar problemas con los requerimientos. La validación de requerimientos es importante debido a que los errores en el documento de requerimientos pueden conducir a importantes costes (al repetir el trabajo) cuando son descubiertos durante el desarrollo o después de que el sistema esté en uso. El coste de arreglar un problema en los requerimientos haciendo un cambio en el sistema es mucho mayor que reparar los errores de diseño o los de codificación. La razón de esto es que un cambio en los requerimientos

normalmente significa que el diseño y la implementación del sistema también deben cambiar y que se debe probar nuevamente.

#### 4)-Gestión de requerimientos

La gestión de requerimientos es el proceso de comprender y controlar los cambios en los requerimientos del sistema. Es necesario mantenerse al tanto de los requerimientos particulares y mantener vínculos entre los requerimientos dependientes, de tal forma que se pueda evaluar el impacto de los cambios en los requerimientos. Hay que establecer un proceso formal para implementar las propuestas de cambios y vincular éstos a los requerimientos del sistema. El proceso de gestión de requerimientos debería empezar en cuanto esté disponible una versión preliminar del documento de requerimientos, pero se debería empezar a planificar cómo gestionar los requerimientos que cambian durante el proceso de obtención de requerimientos. [3]

#### II.2.2.4. Técnicas utilizadas en la IR

Existen varias técnicas para la IR. Cada técnica puede aplicarse en una o más actividades de la IR; en la práctica, la técnica más apropiada para cada actividad dependerá del proyecto que esté desarrollándose.

- **Análisis de los Stakeholders:** Esta técnica es esencial para identificar todos los grupos de stakeholders y averiguar sus intereses. [56]

Durante el desarrollo de esta técnica se trata de responder a las siguientes preguntas: ¿Quiénes son los stakeholder?; ¿Qué metas ven ellos para el sistema?; ¿Por qué desean contribuir?; ¿Qué riesgos y costos ven?; ¿Qué clase de soluciones ven?

- **Observación:** Los usuarios no siempre son conscientes de lo que realmente hacen y cómo lo hacen. Una forma de eliminar esta ceguera es observar lo que realmente está pasando. El analista puede pasar algún tiempo con los usuarios, observando sus tareas diarias. En algunos casos, los analistas pueden utilizar las cámaras de vídeo (con el permiso de los usuarios) para alargar el período de observación. Esto tiene la ventaja de que más tarde puede revisar las cintas con los usuarios y ver lo que realmente se realiza.

La observación mejora enormemente su conocimiento sobre las tareas en curso y algunos problemas de trabajo asociados. [56]

- **Demostración de Tareas:** Es una variante de las entrevistas y la observación. Se les solicita a los usuarios que demuestren cómo realizan una tarea específica. En muchos casos, los usuarios no son capaces de explicar como realizan sus tareas diarias, pero si son capaces de mostrar como realizan tareas específicas. [56]
- **Estudio de Documentación:** Es una forma rápida para obtener información de datos de viejas bases de datos. El analista estudia documentos existentes tales como formularios, archivos, documentación del sistema de computación existente. También se puede realizar impresiones de pantallas del sistema existente. [56]
- **Grupos de Foco:** Se parecen a las sesiones de la técnica de tormenta de ideas, pero son más estructurados. Un grupo de foco inicia con una fase en donde los participantes plantean problemas en la forma actual de hacer las cosas. Luego viene una fase en donde los participantes tratan de imaginar una forma ideal de hacer las cosas. El grupo también intenta explicar por qué esas ideas son buenas. Esto ayuda a formular metas y requerimientos para el nuevo sistema. Varios grupos de stakeholder pueden participar, y al final de la sesión, cada grupo identifica sus temas de alta prioridad. [56]
- **Workshops de Dominio:** Un usuario y desarrollador de workshops ayuda a analizar o diseñar un sistema. En un workshop de dominio el equipo mapea los procesos de negocio. El resultado puede ser una descripción de tareas, diagramas de flujo o diagramas de actividad que describen lo que está en el dominio. Luego, el analista convierte estas descripciones en requerimientos. Es importante que los usuarios expertos participen en los workshops de dominio. Ellos conocen todos los detalles de negocio en su propio dominio. Los administradores pueden participar, aunque raramente conocen los detalles reales de los procedimientos, y no pueden reemplazar a los usuarios expertos. Sin embargo, ellos pueden ser un instrumental para la definición de metas y visiones. [56]
- **Workshops de Diseño:** Los usuarios de workshop de diseño y los desarrolladores cooperan en el diseño de algo, usualmente una interfaz de usuario. [56]

- **Experimentos Piloto:** En muchos casos, el costo de un nuevo sistema puede ser elevado, pero el principal riesgo es si la organización puede adaptarse al sistema y usarlo para mejorar el rendimiento. Los cambios organizacionales mismos a menudo son más costosos que el producto. En esta situación muchos de los riesgos pueden ser eliminados con un experimento piloto. Una pequeña parte de la organización prueba el nuevo sistema sobre unas bases de prueba, pero con datos de producción reales. Al mismo tiempo ellos experimentan con procedimientos de trabajo nuevos. El equipo de proyecto observa los resultados y evalúa los costos y beneficios del nuevo sistema. Esto también ayuda a definir los requerimientos finales y sus prioridades. El problema principal puede ser que el sistema piloto se use por un periodo de tiempo extenso. [56]
- **Puntos de Vista:** Los requerimientos pueden venir de los stakeholder del sistema, del dominio de aplicación y de otros sistemas que interactúan con el sistema a especificar. Estas fuentes de requerimientos se pueden representar como puntos de vista del sistema, donde cada uno representa un subconjunto de requerimientos para el sistema. Cada punto de vista proporciona una perspectiva nueva en el sistema, pero estas no son completamente independientes. [3]
- **Entrevistas:** Esta técnica se emplea para reunir información proveniente de personas o de grupos. Durante la entrevista, el analista conversa con el entrevistado. Por lo general, los entrevistados son usuarios de los sistemas existentes o usuarios en potencia del sistema nuevo. El éxito de esta técnica depende de la habilidad del entrevistador y de su preparación para la misma. [40]
- **Cuestionarios:** Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario. Este cuestionario será cumplimentado por el grupo de personas entrevistadas o simplemente para recoger información en forma independiente de una entrevista. [26]
- **Sistemas Existentes:** Esta técnica consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el sistema a ser construido. Por un lado, se puede analizar las interfaces de usuarios, observando el tipo de información que se maneja y como es manejada; por otro lado, también es útil

analizar las distintas salidas que los sistemas producen (listados, consultas, etc), por que siempre pueden surgir nuevas ideas sobre la base de éstas. [40]

- **Escenarios:** Esta técnica consiste en describir las características del sistema a desarrollar mediante una secuencia de pasos. La representación puede variar dependiendo del autor. Estas representaciones pueden ser casi textuales o ir encaminadas hacia representaciones gráficas en forma de diagramas de flujo. El análisis de escenarios puede ofrecer información importante sobre las necesidades funcionales del sistema. [26]
- **Etnografía:** Es una metodología de investigación de los métodos cualitativos de la antropología cultural y el estudio de las razas y pueblos de una región o sociedad particular. Consiste en la recolección de datos en el terreno y teniendo como informantes a los integrantes de una comunidad dada. Los datos recopilados consisten en la descripción densa y detallada de sus costumbres, creencias, mitos, genealogías, historia, etc. Dicha información se obtiene, sobre todo, por medio de largas entrevistas con miembros de la comunidad o informantes claves de ellas.

Según Sommerville, es una técnica de observación que se puede utilizar para entender los requerimientos sociales y organizacionales. Un analista se sumerge en el entorno laboral donde se utilizará el sistema. Observa el trabajo diario y anota las tareas reales en las que los participantes están involucrados. [57] [3]

- **Joint Application Development/ Desarrollo Conjunto de Aplicaciones:** Esta técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes. El equipo de trabajo se reúne en varias sesiones. En cada una de ellas se establecen los requisitos de alto nivel a trabajar, el ámbito del problema y la documentación. Durante la sesión se discute en grupo sobre estos temas llegándose a una serie de conclusiones que se documentan. En cada sesión se van concretando más las necesidades del sistema. [26]
- **Tormenta de Ideas (Brainstorming):** Es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las

mismas. El número de personas que participa no debe ser muy numeroso, una de ellas debe asumir el rol de moderador. Como técnica de adquisición de requisitos es fácil de usar y de aplicar. Ofrece una visión general de las necesidades del sistema, pero no sirve para ofrecer detalles concretos del mismo, por lo que suele aplicarse en los primeros encuentros. [26]

- **Mapas Conceptuales:** Son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar. Son fáciles de entender por el usuario, más aun si están desarrollados en el lenguaje de éste. Pueden llegar a ser muy ambiguos en casos complejos si no se acompaña de una descripción textual. [26]
- **Comparación de Terminología:** Esta técnica es utilizada en forma complementaria a otras para obtener consenso respecto de la terminología a ser usada en el proyecto de desarrollo. Para ello es necesario identificar el uso de términos diferentes para los mismos conceptos, misma terminología para diferentes conceptos o cuando no hay concordancia exacta ni en el vocabulario ni en los conceptos. [26]
- **Proceso de Análisis Jerárquico (AHP):** Esta técnica tiene por objetivo resolver problemas cuantitativos, para facilitar el pensamiento analítico y las métricas. Consiste en una serie de pasos a saber: encontrar los requerimientos que van a ser priorizados; combinar los requerimientos en las filas y columnas de la matriz de AHP; hacer algunas comparaciones de los requerimientos en la matriz; sumar las columnas; normalizar la suma de las filas y calcular los promedios. Para un volumen grande de requerimientos, esta técnica no resulta la más adecuada. [25]
- **Lenguaje Natural:** Esta técnica consiste en definir los requisitos en lenguaje natural sin usar reglas para ello. Pero, a pesar de que son muchos los trabajos que critican su uso, es cierto que a nivel práctico se sigue utilizando. Resulta ser una técnica muy ambigua para la definición de requisitos. [26]
- **Glosario y Ontologías:** Su uso proviene de la necesidad de establecer un marco de terminología común. En un glosario de términos se recogen y definen



los conceptos más relevantes y críticos para el sistema. En las ontologías no sólo se aparecen los términos, sino también las relaciones entre ellos. [26]

- **Plantilla o Patrones:** Esta técnica tiene por objetivo el describir los requisitos mediante el lenguaje natural pero de una forma estructurada. Una plantilla es una tabla con una serie de campos y una estructura predefinida que el equipo de desarrollo va cumplimentando usando para ello el lenguaje del usuario. Las plantillas eliminan parte de la ambigüedad del lenguaje natural al estructurar la información. Sin embargo, si el nivel de detalle elegido es demasiado detallado, el trabajo de rellenar plantillas y mantenerlas puede ser tedioso. [26]
- **Lenguajes Formales:** Esta técnica se usa para describir los requisitos de un sistema. Resultan ser complejas en su utilización y para ser entendidas por el cliente. El mayor inconveniente es que no favorecen la comunicación entre cliente y analista. Por el contrario, es la representación menos ambigua de los requisitos y la que más se presta a técnicas de verificación automatizadas. [26]
- **Lenguaje Natural Estructurado:** este enfoque depende de la definición de formularios o plantillas estándares para expresar la especificación de requerimientos. [3]
- **Lenguajes de Descripción de Diseño:** este enfoque utiliza un enfoque similar a uno de programación, pero con características más abstractas, para especificar los requerimientos por medio de la definición de un modelo operativo del sistema. Este enfoque no se utiliza ampliamente en la actualidad, aunque puede ser útil para especificaciones de interfaces. [3]
- **Notaciones gráficas:** para definir los requerimientos funcionales del sistema, se utiliza un lenguaje gráfico, complementado con notaciones de texto. [3]
- **Reviews (Revisiones):** Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar la consistencia de la documentación o información faltante. [26]
- **Auditorías:** La revisión de la documentación con esta técnica consiste en el chequeo de los resultados contra una lista predefinida, o definida a comienzos del proceso, es decir que solo una muestra es revisada. [26]

- **Matriz de Trazabilidad:** Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos. [26]
- **Prototipos:** Se basa en obtener un prototipo, a partir de la definición de requisitos, sin tener la totalidad de la funcionalidad del sistema, permiten al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final. [26]
- **Diccionario de Datos:** es una herramienta de modelado importante, sin él el usuario no podrá estar seguro de que entendió los detalles de la aplicación. Es un listado organizado de todos los datos pertinentes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas, componentes de almacenes y cálculos intermedios. [48]
- **Diagrama Entidad Relación (DER):** es una notación gráfica para modelar datos. El DER es un modelo de red que describe con un alto nivel de abstracción la distribución de datos almacenados en un sistema. La importancia de modelar los datos de un sistema podría ser primeramente, porque las estructuras de datos y las relaciones pueden ser tan complejas que se deseará enfatizarlas y examinarlas independientemente del proceso que se llevará a cabo. [48]
- **Diagrama de Transición de Estados (STD):** es una herramienta de modelado que enfatiza el comportamiento dependiente del tiempo del sistema. El comportamiento dependiente del tiempo es un aspecto importante de muchos sistemas. Para algunos sistemas computacionales de empresas este aspecto no es importante, puesto que la secuencia es esencialmente trivial. Así, en muchos sistemas computacionales la función N no puede llevar a cabo su labor hasta que reciba la entrada que requiere y esta entrada se produce como salida de una función N-1, y así sucesivamente. [48]
- **Diagrama de Flujo de Datos (DFD):** es una herramienta gráfica de modelado que permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por conductos y tanques de almacenamiento de datos. Esta

herramienta es mas comúnmente usada por sistemas operacionales en los cuales las funciones del sistema son de gran importancia y son más complejas que los datos que éste maneja. [48]

- **Diagrama de Contexto:** es un caso especial del DFD, en donde una burbuja representa la totalidad del sistema. El diagrama de contexto enfatiza varias características importantes del sistema: las personas, las organizaciones y los sistemas (terminadores), con los que se comunica el sistema; los datos que el sistema recibe del mundo exterior y que deben procesarse de alguna forma; los datos que el sistema produce y que se envían al mundo exterior; los almacenes de datos que el sistema comparte con los terminadores y la frontera entre el sistema y el resto del mundo. [48]
- **Diagrama de Casos de Uso:** Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. Muestran un conjunto de casos de uso y de actores y sus relaciones. [49]
- **Diagrama de Interacción:** consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos; los diagramas de interacción tratan la vista dinámica de un sistema; un término genérico que se aplica a varios tipos de diagramas que enfatizan las interacciones de objetos, incluyendo diagramas de colaboración, diagramas de secuencia y diagramas de actividad. [49]
- **Diagrama de Clases:** muestra un conjunto de clases, interfaces y colaboraciones y las relaciones entre éstos; los diagramas de clases muestran el diseño de un sistema desde un punto de vista estático; es un diagrama que muestra una colección de elementos (estáticos) declarativos. [49]
- **Diagrama de Colaboración:** es un diagrama de interacción que enfatiza la organización estructural de los objetos que envían y reciben mensajes; muestra las interacciones organizadas alrededor de instancias y de los enlaces entre ellas. [49]

- **Diagrama de Componentes:** muestra un conjunto de componentes y sus relaciones; muestran los componentes de un sistema desde un punto de vista estático. [49]
- **Diagrama de Despliegue:** muestra un conjunto de nodos y sus relaciones; un diagrama de despliegue muestra el despliegue de un sistema desde un punto de vista estático. [49]
- **Diagrama de Objetos:** muestra un conjunto de objetos y sus relaciones en un momento determinado; los diagramas de objetos muestran el diseño o los procesos de un sistema desde un punto de vista estático. [49]
- **Diagrama de Secuencia:** es un diagrama de interacción que hace énfasis en la ordenación temporal de los mensajes. [49]
- **Diagrama Nassi\_Shneiderman:** estos diagramas se introdujeron como una técnica estructurada de creación de diagramas de flujo. Son más organizados, más estructurados y más comprensibles que un diagrama de flujo típico, por ello, a veces se los prefiere como herramienta para crear especificaciones de proceso. [48]
- **Diagrama HIPO:** se usa para presentar una visión de alto nivel de las funciones que realiza el sistema, al igual que descomposición de funciones en subfunciones, etc. Puede ser una herramienta de modelado útil porque se parece al diagrama de organización que describe la jerarquía de gerentes, subgerentes, etc. Sin embargo, no muestra los datos utilizados o producidos por el sistema. [48]
- **Diagrama de Flujo:** muestra una lógica secuencial y de tipo procedimiento. No se aconseja su uso; sin embargo, se usan para describir lógica detallada y, si el analista de sistemas se limita a los símbolos de elaboración de diagramas de flujo equivalentes a las construcciones del español estructurado, entonces no tiene nada de malo su uso. [48]
- **Pre/post Condiciones:** son una manera conveniente de describir la función que debe realizar el proceso, sin decir mucho a cerca del algoritmo o procedimiento que se utilizará. Las precondiciones describen todas las cosas que deben darse antes de que el proceso pueda comenzar a ejecutarse. De manera similar las postcondiciones describen lo que debe darse cuando el proceso ha concluido.

Aunque este enfoque es bastante útil y tiene un gran número de ventajas, hay ocasiones en las cuales puede no ser apropiado. La falta de pasos intermedios entre entradas y salidas es deliberada y consciente, pero puede volverse difícil de entender si el lector no visualiza algún tipo de procedimiento que lleve de las entradas a las salidas. Además, si existen relaciones complejas entre entradas y salidas, podría ser más fácil escribir una especificación utilizando lenguaje estructurado. [48]

- **Árbol de Decisión:** Es una herramienta gráfica que permite expresar la lógica de un proceso con claridad y sin ambigüedades. [50]
- **Tabla de Decisión:** es una herramienta para especificar procesos. Son útiles sobre todo cuando un proceso debe producir alguna salida o tomar alguna acción basada en decisiones complejas. Si las decisiones se basan en diversas variables, y si dichas variables pueden tomar diversos valores, entonces se prefiere una tabla de decisiones como herramienta de especificación. [48]

#### II.2.2.5. Herramientas para la IR

**CASE (Ingeniería del Software Asistida por Computadora):** Es el nombre que se le da al software que se utiliza para ayudar a las actividades del proceso del software como la IR, el diseño, el desarrollo de programas y las pruebas. Por lo tanto las herramientas CASE incluyen editores de diseño, diccionarios de datos, compiladores, depuradores, herramientas de construcción de sistemas, etc.

La tecnología CASE proporciona ayuda al proceso de software automatizando algunas de sus actividades, así como proporcionando información acerca del software en desarrollo. [3]

#### Clasificación de CASE

Existen varias formas diferentes de clasificar las herramientas CASE, cada una de las cuales brinda una perspectiva diferente de estas herramientas. A continuación se describen dichas herramientas desde tres de estas perspectivas [3]:

1. **Perspectiva funcional:** las herramientas CASE se clasifican de acuerdo con su función específica.
2. **Perspectiva de proceso:** las herramientas se clasifican de acuerdo con las actividades del proceso que ayudan.

**3. Perspectiva de integración:** las herramientas CASE se clasifican de acuerdo con la forma en que están organizadas en unidades integradas que proporcionan ayuda a una o más actividades del proceso.

La Tabla 2.1 muestra una clasificación de las herramientas de acuerdo a su función. Esta tabla enumera diferentes tipos de herramientas CASE y da ejemplos específicos de cada una.

**Tabla 2.1.** Clasificación funcional de herramientas CASE

<b>Tipos de Herramientas</b>	<b>Ejemplos</b>
Herramientas de planificación	Herramientas PERT, herramientas de estimación, hojas de cálculo.
Herramientas de edición	Editores de texto, editores de diagramas, procesadores de texto.
Herramientas de gestión del cambio	Herramientas de rastreo de requerimientos, sistemas de control de cambios.
Herramientas de gestión de la configuración	Sistema de gestión de versiones, herramientas de construcción de sistemas.
Herramientas de construcción de prototipos	Lenguajes de muy alto nivel, generadores de interfaz de usuario.
Herramientas de apoyo a métodos	Editores de diseño, diccionarios de datos, generadores de código.
Herramientas de procesamiento de lenguaje	Compiladores, intérpretes.
Herramientas de análisis de programas	Generadores de referencias cruzadas, analizadores estáticos, analizadores dinámicos.
Herramientas de pruebas	Generadores de pruebas de datos, comparadores de archivos.
Herramientas de depuración	Sistemas de depuración interactiva.
Herramientas de documentación	Programas de diseño de páginas, editores de imágenes.
Herramientas de reingeniería	Sistemas de referencias cruzadas, sistemas de reestructuración de programas.

La figura 2.1, muestra una clasificación alternativa de las herramientas CASE. Muestra las fases del proceso de desarrollo de software que reciben ayuda por varios tipos de herramientas CASE.

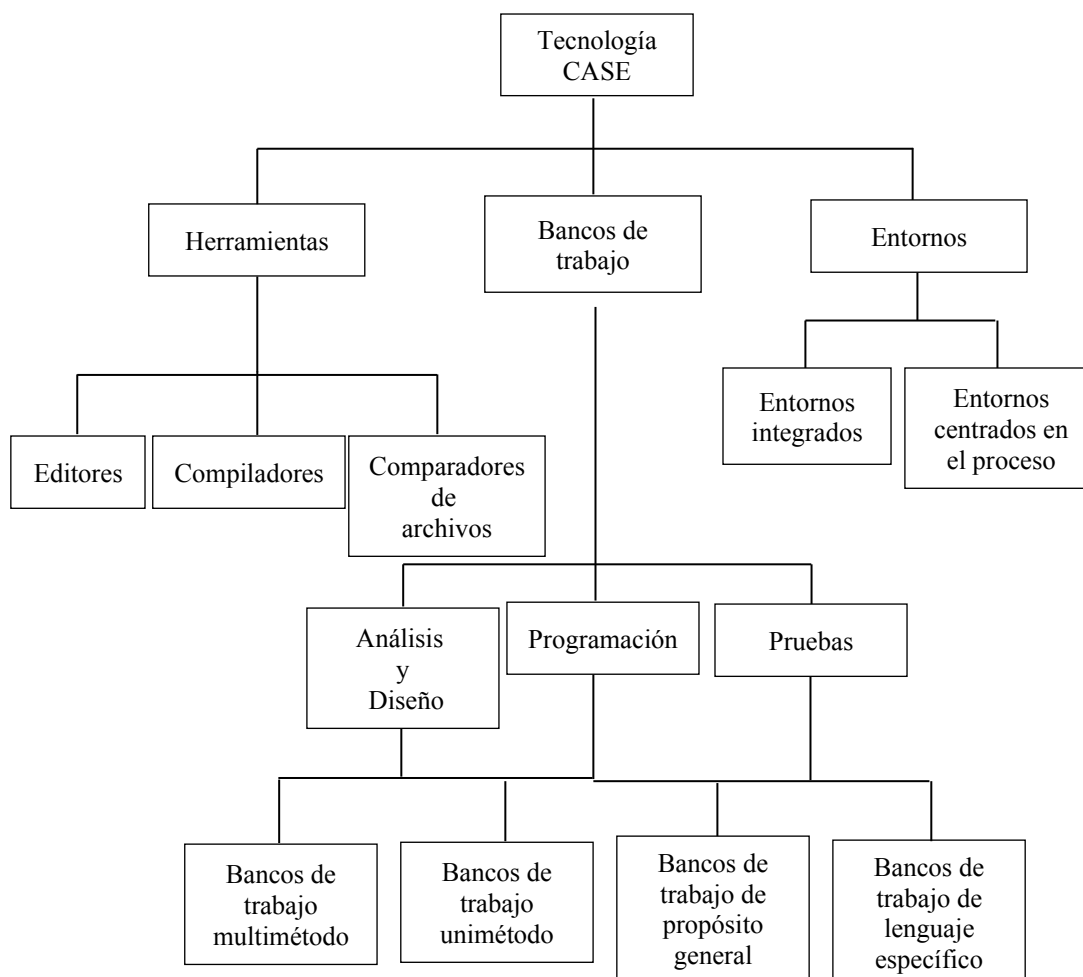
Herramientas de reingeniería			•	•
Herramientas de pruebas			•	•
Herramientas de depuración			•	•
Herramientas de análisis de programas			•	
Herramientas de procesamiento de lenguajes		•	•	
Herramientas de apoyo a métodos	•	•		
Herramientas de construcción de prototipos	•			•
Herramientas de gestión de configuración		•	•	
Herramientas de gestión de cambios	•	•	•	•
Herramientas de documentación	•	•	•	•
Herramientas de edición	•	•	•	•
Herramientas de planificación	•	•	•	•
			Especific.	Diseño
			Implementación	Verificación

**Figura 2.1.** Clasificación basada en las actividades de las herramientas CASE

Otra dimensión de clasificación posible es la amplia ayuda que ofrece la tecnología CASE para el proceso del software. Fuggetta propone que los sistemas CASE se deben clasificar en tres categorías:

1. Las herramientas ayudan a las tareas individuales del proceso como la verificación de la consistencia de un diseño, la compilación de un programa y la comparación de los resultados de las pruebas. Las herramientas pueden ser de propósito general, independientes o agrupadas en bancos de trabajo.
2. Los bancos de trabajo ayudan a las fases o actividades del proceso como la especificación, el diseño, etc. Normalmente consisten en un conjunto de herramientas con algún grado mayor o menor de integración.
3. Los entornos ayudan a todos los procesos del software, o al menos a una parte sustancial de éstos. Normalmente incluyen varios bancos de trabajo agrupados.

La figura 2.2 ilustra esta clasificación y muestra algunos ejemplos de estas clases de ayuda CASE.



**Figura 2.2.** Herramientas, bancos de trabajo y entornos

Cabe destacar que para el desarrollo de la ontología se tiene en cuenta la clasificación basada en las actividades para las cuales brindan soporte las herramientas CASE (Figura 2.1).

## II.2.3. METODOLOGÍA DorCu PARA LA INGENIERÍA DE REQUERIMIENTOS

### II.2.3.1. Introducción

Diferentes autores descomponen al proceso de IR de diversas formas. Es así, por ejemplo, Rzepka, lo considera conformado por tres actividades [34]:

- Elicitar los requerimientos de las diversas fuentes individuales.
- Asegurar que las necesidades de todos los usuarios son consistentes y factibles.



- Validar que los requerimientos que se derivaron son un reflejo exacto de las necesidades del usuario.

Este modelo implica una clasificación secuencial de las actividades, con la elicitación realizada una vez al inicio del proceso. En realidad, sin embargo, el proceso es iterativo, con estas actividades ejecutadas muchas veces, ya que la tarea de definición de requerimientos no puede definirse por medio de una simple progresión a través de, o relación entre, adquisición, expresión, análisis y especificaron. Los requerimientos evolucionan a un paso desigual y tienden a generar requerimientos más extensos a partir de los procesos de definición.

Por tanto, la construcción de la especificación de requerimientos es inevitablemente un proceso iterativo. Así, en cada iteración es necesario considerar si la versión actual de la especificación de requerimientos define el requisito del cliente adecuadamente y, si no lo hace, cómo debe cambiarse o debe extenderse más.

Oberg plantea que, desde el momento en que los requerimientos son necesidades que deben satisfacer los sistemas a ser construidos, y que la satisfacción de determinados conjuntos de requerimientos define el éxito o el fracaso de los proyectos, tiene sentido buscar los requerimientos, escribirlos, organizarlos y seguirlos en el momento en que cambian. Dicho autor considera que la Administración de Requerimientos, haciendo referencia a la IR, es:

- Un enfoque sistemático para elicitar, organizar y documentar los requerimientos del sistema.
- Un proceso que establece y mantiene un acuerdo entre el cliente, el usuario y el equipo del proyecto sobre los requerimientos cambiantes del sistema.

La Rational Software Company, dice Oberg, asume que el término Administración hace una descripción más apropiada de todas las actividades involucradas, y enfatiza la importancia del seguimiento de los cambios para mantener los acuerdos entre la comunidad de usuarios y el equipo del proyecto.

Sin embargo, se considera que este término, en caso de incorporarse al proceso de definición de requerimientos, hace referencia a una actividad de la IR que sirve para controlar y seguir los cambios de los requerimientos.

En el presente trabajo, se sostiene que el término Ingeniería es más apropiado ya que no solo se plantean etapas de definición, sino técnicas, métodos y herramientas involucradas en cada una, lo que lleva a definirla como una disciplina independiente.

Dorfman y Thayer, plantean una definición similar, con la que coinciden en gran medida las autoras de este trabajo, considerando que la IR incluye tareas de elicitación, análisis, especificación, validación y administración de requerimientos de software, siendo la “administración de requerimientos de software” la planificación y control de todas esas actividades relacionadas.

Como puede verse, para sustentarse, la IR sugiere la existencia de un eje troncal de etapas, dejando abierta la posibilidad de que cada uno las refinen cuanto sea necesario. Por tanto, si bien existen diferentes enfoques, estos tienen un común denominador, que puede resumirse en las siguientes etapas fundamentales: Elicitación, Análisis y Especificación, que son las que se adoptan en el presente trabajo:

- *Elicitación*: Es la etapa de mayor interacción con el usuario. Es el momento en el que se recurre, por ejemplo, a la observación, lectura de documentos, entrevistas y relevamientos, entre otras técnicas; la instancia en que equipos multidisciplinarios trabajan conjuntamente con el cliente/usuario, para obtener los requerimientos reales de la mejor manera.
- *Análisis*: La etapa de análisis de requerimientos permite al analista representar el dominio de la información (también conocido como el Universo de Información - UdI) de la aplicación a desarrollar, a través de un uso de un lenguaje más técnico, procurando reducir ambigüedades. Brinda al analista, la representación de la información y las funciones que facilitarán la definición del futuro diseño.
- *Especificación*: No cabe ninguna duda de la importancia de esta etapa y de que la forma de especificar tiene mucho que ver con la calidad de la solución. Los analistas que se han esforzado en trabajar con especificaciones incompletas, inconsistentes o mal establecidas han experimentado la frustración y confusión que invariablemente se produce. Las consecuencias se padecen en la calidad, oportunidad e integridad del software resultante.

Como se puede apreciar, existen grandes diferencias en la terminología usada por los autores de la bibliografía consultada, siendo este un punto importante para destacar, a los efectos del entendimiento de la IR. Es así que hay autores que ven al Análisis de Requerimientos como el proceso completo de definición de requerimientos y no como una etapa metodológica de la IR. De igual manera, la especificación de requerimientos tiene diferentes acepciones: algunos autores se refieren a ella como a

una etapa en la que se describen los requerimientos y otros como a la actividad completa, desde la Elicitación hasta la Especificación propiamente dicha.

De igual manera, se hace referencia a la IR con otros términos tales como “Etapa de Requerimientos” (IEEE - Institute of Electric and Electronic Engineer) y “Administración de requerimientos” (Rational Software Corporation). Pero se debe destacar que, tanto los autores Dorfman y Thayer como Christel, hacen una correcta referencia a IR, diferenciando claramente sus actividades intermedias.

En cuanto a la adopción de las etapas de Elicitación, Análisis y Especificación como eje de la investigación, la decisión fue tomada por considerarse que facilitan el entendimiento de las tareas que en ellas se realizan.

### **II.2.3.2. Propuesta metodológica para la Ingeniería de Requerimientos**

#### **II.2.3.2.1. Metodología e Ingeniería de Requerimientos**

La diferencia entre método y metodología, que establece Checkland, es la que se toma como base para la presente propuesta. Dicho autor afirma que: “la esencia de una metodología – en forma opuesta a lo que ocurre en un método o técnica- es que ofrece un conjunto de pautas o principios que en cualquier instancia específica pueden ser ajustadas tanto a las características de la situación en la cual debe ser aplicada como a las personas que usan el enfoque. Es tal la variedad de situaciones problemáticas humanas que no habrá ningún enfoque para la solución de problemas que pueda ser reducido a una fórmula estándar y manejar aun toda la riqueza de las situaciones en particular”.

Luego de haber realizado el estudio de los materiales a los que se pudo acceder, en el presente trabajo se adopta una definición para IR que se apoya en definiciones de los autores Leite, Loucopoulos, Dorfman y de la IEEE y en los problemas que aun siguen sin resolver.

Por tanto, se define a la IR como *un proceso centrado en el cliente/usuario y sus necesidades, en donde las etapas de Elicitación, de Análisis, de Especificación y de Validación y Certificación de requerimientos iteran hasta la obtención de documentos isomórficos representativos de las necesidades reales de clientes/usuarios, depuradas en base a procesos cooperativos que se llevan a cabo en distintas comunidades del dominio de información y en donde el isomorfismo de los documentos finales permite salvar la brecha que existe entre el enfoque antropológico de la IR y las etapas siguientes de la Ingeniería de Software,*

*permitiendo que las mismas continúen hacia la prosecución de un software sin fallos.*

En cuanto a la no aplicabilidad general de los métodos investigados, unas veces ella es consecuencia de las características particulares del proyecto, otras de problemas derivados de cuestiones económicas que impiden contar con todos los recursos que son necesarios, otras de restricciones de índole tecnológica. Pero esta lista puede seguir ampliándose más y más a medida que aumenta el conocimiento que se tiene respecto a lo que se pretende realizar, por lo que se decide que la metodología propuesta debe ser lo suficientemente flexible y centrada en las necesidades del usuario como para permitir su adecuación según las cuestiones organizacionales, sociales, económicas, tecnológicas y medioambientales.

### **II.2.3.3. Metodología DorCu**

La metodología DorCu (Documentación de Requerimientos Centrada en el Usuario), consta de las siguientes etapas:

- Elicitación de requerimientos
- Análisis de requerimientos
- Especificación de requerimientos
- Validación y certificación de requerimientos

Y los objetivos que se proponen para cada una de ellas son:

- Elicitación de requerimientos

Esta es la etapa en donde se adquiere el conocimiento del trabajo del cliente/usuario, se busca comprender sus necesidades y se detallan las restricciones medioambientales. Como resultado de las acciones realizadas se tiene el conjunto de los requerimientos de todas las partes involucradas.

- Análisis de requerimientos

En esta etapa se estudian los requerimientos extraídos en la etapa previa a los efectos de poder detectar, entre otros, la presencia de áreas no especificadas, requisitos contradictorios y peticiones que aparecen como vagas e irrelevantes. El resultado de haber llevado a cabo las tareas que involucran estos términos puede, en más de una oportunidad, hacer que se deba regresar a la primera etapa, a los efectos de eliminar todas las inconsistencias y falencias que se han detectado. En esta etapa ya se realizan aproximaciones a un lenguaje técnico.

- Especificación de requerimientos

Partiendo de lo elaborado en la etapa anterior tales como funciones, datos, requerimientos no funcionales, objetivos, restricciones de diseño/implementación o costos, e independientemente de la forma en que se realice, esta etapa es un proceso de descripción del requerimiento. Si se presentan dificultades para especificar un requerimiento se debe volver a la etapa anterior que se crea conveniente.

- Validación y certificación de requerimientos

Esta etapa final se nutre de las anteriores y realiza la integración y validación final de lo obtenido en cada una de las etapas anteriores dando, como resultado final, el Documento de Requerimientos. Este documento no es uno solo sino que, como mínimo, existen dos que son isomórficos entre sí: uno destinado al cliente/usuario a los efectos de la certificación de los requisitos y el otro técnico, orientado a nutrir las restantes etapas de la Ingeniería de Software. Y, al igual que en el caso anterior, su resultado puede ser la necesidad de retornar a la Especificación e incluso a la Elicitación, iterando entre etapas y sin perder contacto con el cliente/usuario.

Considerando el tronco metodológico planteado anteriormente, se detallan a continuación sus correspondientes subetapas, destacando en primer lugar que las subetapas de *Elicitación de Requerimientos*, exceptuando la 1.1, son tomadas de la metodología propuesta por Christel debido a que se las considera apropiadas para los fines que se persiguen.

1. Elicitación de Requerimientos

- 1.1 Formar el equipo multidisciplinario.
- 1.2 Buscar hechos.
- 1.3 Recolectar y clasificar requerimientos.
- 1.4 Evaluar y racionalizar.
- 1.5 Dar prioridad.
- 1.6 Integrar y validar.
- 1.7 Documentar la etapa

2. Análisis de Requerimientos
  - 2.1 Reducir ambigüedades en los requerimientos.
  - 2.2 Traducir a lenguaje técnico los requerimientos.
  - 2.3 Plantear un modelo lógico.
  - 2.4 Documentar la etapa.
  
3. Especificación de Requerimientos
  - 3.1 Determinar el tipo de requerimiento.
  - 3.2 Elegir la herramienta de especificación acorde al tipo de requerimiento.
  - 3.3 Especificar de acuerdo a la herramienta seleccionada.
  - 3.4 Documentar la etapa.
  
4. Validación y Certificación de los Requerimientos
  - 4.1 Seleccionar las fuentes de información entre De y Da a los fines de validar el Dp.
  - 4.2 Elegir o diseñar el modelo de documento acorde al grado de detalle requerido y al lector final.
  - 4.3 Elegir la herramienta de documentación que mejor se aplica al modelo seleccionado.
  - 4.4 Documentar respetando los estándares vigentes a la fecha de realización del documento de requerimientos.
  - 4.5 Verificar que el documento de requerimientos del usuario DRu sea isomórfico con el documento técnico DRt.
  - 4.6 Certificar el documento de requerimientos DRu a través del conforme del usuario.

#### **II.2.3.3.1. Somera explicación de las tareas de cada etapa**

##### ***Etapal: Elicitación de requerimientos***

En cuanto al proceso de Elicitación de requerimientos, la propuesta metodológica que se considera apropiada consta de los siguientes pasos:

##### 1.1 Formar el equipo multidisciplinario

Considerando que la formación de la gente de sistemas, tratándose de problemas con alta incidencia del factor humano, no tiene la especialización necesaria como

para diagnosticar el método de Elicitación más apropiado para cada caso en particular, se aconseja que la recolección de requerimientos sea efectuada con el asesoramiento de profesionales especializados. Este asesoramiento puede extenderse incluso a un liderazgo activo de las sesiones de Elicitación por parte de especialistas en ciencias de la comunicación o en ciencias del conocimiento.

### 1.2 Buscar hechos

El primer paso en la Elicitación de requerimientos tiene que ver con el problema a encarar, y quién necesita ser involucrado en esta toma de decisión, tanto como quién se verá afectado por la formulación de los problemas y la eventual solución. Los resultados de esta actividad son: una declaración del contexto del problema, de los objetivos globales, límites e interfaces para el sistema original. Este examen debe efectuarse de manera tal que permita establecer, entre otros, cuál es el rol que desempeñará el sistema a desarrollar, sus objetivos y límites, las restricciones de arquitectura y la existencia o no de sistemas similares dentro de la organización.

### 1.3 Recolectar y clasificar los requerimientos

En esta etapa se obtienen: objetivos, necesidades y requerimientos de clientes y usuarios. Estas necesidades y requerimientos son verificadas comparándolas con los objetivos globales del sistema original expresados durante el hallazgo de hechos. Es importante recolectar tanta información como sea posible. Dependiendo de la manera en cual el sistema se está desarrollando y los grupos que afectará, la etapa de recolección de requerimientos es una combinación de los enfoques composición y descomposición. Es importante en este momento, destacar los términos que son propios del lenguaje del UDI.

Una vez recolectados los requerimientos, se debe proceder a clasificar los mismos en funcionales y no funcionales.

### 1.4 Evaluar y racionalizar

Debe realizarse una valoración del riesgo, para encaminar las inquietudes técnicas, de costos y tiempo. Debe examinarse la coherencia en la información reunida en subetapas previas, para determinar si los requerimientos verdaderos están escondidos o expresados explícitamente. Se realizan abstracciones para

responder preguntas del tipo ¿Por qué usted necesita X?, y si esta pregunta tiene una respuesta concreta, entonces es un requerimiento, sino es un falso requerimiento. Mediante el estudio comparativo de la información de requerimientos se ponen en evidencia las inconsistencias que pueden surgir entre los requerimientos extraídos. Cabe destacar que, tanto en la presente subetapa como en la anterior, se dan instancias de evaluación de factibilidad, negociables entre el cliente/usuario y el analista.

### 1.5 Dar prioridad

En esta etapa, contando ya con requerimientos consistentes, se da un orden de prioridades, de manera tal que las necesidades de alta prioridad pueden ser encaradas primero, lo que permite definir las y reexaminar los posibles cambios de los requerimientos, antes que los requerimientos de baja prioridad (que también pueden cambiar) sean implementados. Durante el desarrollo del sistema, esto permite una disminución de los costos y ahorro de tiempo en procesamiento de los inevitables cambios de los requerimientos.

Los requerimientos deben tener prioridades basándose en las necesidades del usuario, el costo y la dependencia.

### 1.6 Integrar y validar

Esta tarea se lleva a cabo de manera tal que sea posible obtener un conjunto de requerimientos, expresados en el lenguaje del usuario, de los cuales se pueda validar la consistencia con respecto a las metas organizacionales obtenidas en la primera etapa. Las tareas de integración deben ser ejecutadas principalmente por el analista de sistemas, y los resultados del proceso de Elicitación comunicarlos a las otras comunidades involucradas. Esta validación de los requerimientos, realizada por todas las partes afectadas, asegura que se alcanza lo deseado.

### 1.7 Documentar la etapa

Elaborar la lista final de los términos del lenguaje del UdI, y la de sentencias de los requerimientos obtenidos (De).

Como es de esperar, a los efectos de obtener buenos requerimientos, todos estos pasos deben iterar ante la menor inconsistencia detectada, aconsejándose que la iteración se realice recurriendo al cliente/usuario, tantas veces como sea



necesario, para garantizar una correcta depuración del producto final de la etapa de Elicitación.

### ***Etapa 2: Análisis de Requerimientos***

Los pasos a realizar durante esta etapa tienen como objetivo la obtención de buenos requerimientos, para asegurar de esta manera que lo que se derivará a las etapas posteriores será de calidad tal que permita que disminuyan las fallas de los sistemas.

Las subetapas a contemplar durante esta etapa son:

#### ***2.1 Reducir ambigüedades en los requerimientos***

Los requerimientos obtenidos como resultado final de la etapa de Elicitación, deben ser tratados a los efectos de llevarlos a una notación que permita reducir la ambigüedad del lenguaje del usuario. Por consiguiente, en esta subetapa se realizan las tareas que permiten eliminar los términos que tienen más de una acepción, unificando el léxico empleado en el UdI.

#### ***2.2 Traducir a lenguaje técnico los requerimientos***

Los requerimientos, ya con menos ambigüedades, deben ser tratados a los efectos de llevarlos a un lenguaje que se vaya aproximando al lenguaje técnico. Mediante esta traducción se busca aproximar los términos del usuario a los términos del sistema de software.

#### ***2.3 Plantear un modelo lógico***

Partiendo del lenguaje obtenido en la etapa anterior, se lo transforma en una estructura preliminar, es decir en un primer modelo lógico. De esta manera, en la presente subetapa se debe construir un modelo del problema ya sea en términos de diagramas de flujo o cualquier otro tipo de representación que se considere conveniente para el modelado y que permita, además, establecer un vínculo con la Etapa de Especificación.

#### ***2.4 Documentar la etapa***

Elaborar todo tipo de documento que se considere adecuado como soporte para la etapa siguiente. Este documento, dado el caso, puede resumirse a la colección de los modelos lógicos a que se ha arribado (Da).

### ***Etapas 3: Especificación de Requerimientos***

Recién superadas las etapas anteriores debe comenzar a pensarse en la forma de describir los requerimientos. Para ello, se deben seguir las subetapas planteadas a continuación:

#### *3.1 Determinar tipo de requerimiento*

Considerando que existen diferentes tipos de requerimientos, determinar unívocamente a cuál de ellos pertenece el que se está tratando. Esto no significa que deba adoptarse la clasificación por la cual se han decidido las autoras de este estudio, sino que aquí también queda de manifiesto la flexibilidad de la metodología, ya que cada analista de requerimientos puede utilizar la clasificación que considere como la más adecuada.

#### *3.2 Elegir la herramienta de especificación acorde al tipo de requerimiento*

Una vez definido el tipo de requerimiento, seleccionar la herramienta de representación acorde a dicho tipo y al tipo de especificación que se desea realizar. La única restricción al respecto es que la herramienta a seleccionar debe ser de índole formal o, a lo sumo, semiformal, ya que ellas son las únicas que permiten representar a los requerimientos sin ambigüedades.

#### *3.3 Especificar de acuerdo a la herramienta seleccionada*

Representar el requerimiento sobre la base de la elección realizada en la etapa anterior. En caso de existir dificultades para su empleo, volver a la subetapa anterior para realizar una nueva selección o, incluso, a la primera ya que la dificultad de representación puede obedecer al intento de usar una herramienta para un requerimiento cuyo tipo ha sido mal definido, por lo cual se selecciona una inaplicable al caso.

#### *3.4 Documentar la etapa*

Confeccionar el documento representativo de la etapa tomando como base a los modelos formales o semiformales que se han elaborado al realizar la especificación de los requerimientos. Incorporar al mismo toda extensión que se considere de utilidad para la etapa de Validación y Certificación de Requerimientos (Dp).

#### ***Etapa 4: Validación y certificación de requerimientos***

Todo el esfuerzo realizado durante las etapas anteriores puede darse por perdido si no es manifestado en forma correcta, ya que cualquier mala interpretación puede echar por tierra hasta el proceso de IR más consistente, desvirtuando la naturaleza de lo que fue, hasta el momento de su declaración, un buen requerimiento. Tratando de minimizar aún más la posibilidad de error, se proponen las siguientes subetapas para su elaboración:

##### *4.1 Seleccionar las fuentes de información a partir de las cuales validar el documento de especificación*

En esta etapa se propone validar el documento de especificación Dp a partir de los documentos obtenidos de las etapas de Elicitación (De) y análisis (Da), seleccionando como fuente de información aquellos materiales que más aportan, por un lado a la claridad de su descripción y, por el otro, en cuanto a permitir la validación final entre los resultados de todas las etapas anteriores. El documento de especificación (Dp) validado se llamará, en adelante, documento de requerimientos técnico (DRt).

##### *4.2 Elegir o diseñar el modelo de documento acorde al grado de detalle requerido y al lector final*

Si bien muchos autores han propuesto modelos de documentación excelentes, es necesario decidirse por alguno de ellos. Dado el caso de que ninguno de los conocidos satisfaga las necesidades de documentación del analista de requerimientos, se deberá proceder a diseñar aquel que mejor se ajuste a sus necesidades.

##### *4.3 Elegir la herramienta de documentación que mejor se aplica al modelo seleccionado*

Como no todos los modelos pueden ser plasmados con una misma herramienta, se debe seleccionar la que mejor se adecue al problema entre todas las alternativas posibles. Es así que en esta etapa se deberán tener en cuenta, no solo a los procesadores de texto sino también a herramientas de recursos gráficos que permitan la incorporación de diagramas y figuras, si se considera que lo antedicho ayuda a la interpretación que hará el usuario del Documento de Requerimientos.

#### *4.4 Documentar respetando los estándares vigentes a la fecha de realización del documento de requerimientos*

Elaborar el documento de requerimientos orientado al usuario (DRu) a partir del documento de requerimientos técnico (DRt), realizando una traducción a un lenguaje entendible por aquel. Estos documentos deben ser elaborados respetando los estándares que existen a la fecha de su confección. Para ello, el personal de documentación debe estar al tanto de las normas IRAM e ISO y de las dictadas por instituciones como la IEEE. Como el DRu tiene fines de certificación y contractuales, se debe considerar como normas de redacción las disposiciones legales al momento de la confección.

#### *4.5 Validar*

Verificar la correspondencia entre los documentos obtenidos de la etapa anterior, controlando que solo difieran en lo sintáctico y no en lo semántico, es decir que su contenido difiera solamente en el lenguaje utilizado para su definición, alcanzando de esta manera el isomorfismo entre DRt y DRu.

#### *4.6 Certificar*

Proceder a la aprobación del DRu por medio del conforme del cliente, y de esta manera dar por aprobado el Documento de Requerimientos Técnico DRt, el que será utilizado por las restantes etapas de la Ingeniería de Software.

### **II.2.3.4 Gestión de calidad de procesos**

Muchos estudios determinaron que la calidad del proceso de IR influye críticamente a lo largo del ciclo de vida de desarrollo de software. Una calidad baja de la IR genera requerimientos incompletos, inconsistentes, conflictivos, etc. Tales requerimientos causan defectos significativos en el software que pueden ser difíciles de detectar durante la prueba del sistema y es más caro de resolver que los defectos encontrados durante la programación. Por ello es importante implementar un proceso de IR de calidad.

A pesar de que la metodología DorCu fue creada después de estudiar cuidadosamente las diferentes maneras de llevar a cabo el proceso de IR, definidas por distintos autores, consideramos que la misma carece de un control de calidad del proceso, por lo cual se decidió incorporar una *Gestión de Calidad del Proceso*. Para la misma, se definieron atributos que deberán ser considerados a lo largo del proceso de IR para maximizar la calidad del mismo.

**Atributos:**

- Los objetivos de IR son conocidos por los stakeholders y el grupo de IR.
- Los roles de cada actor están bien definidos.
- La IR existe como un procedimiento separado bien definido.
- Los administradores del proyecto cuentan con suficiente experiencia técnica.
- Los integrantes del equipo de desarrollo realizan un uso adecuado de las herramientas disponibles y recomendadas para la IR.

**II.2.3.5. Conclusiones**

La evolución de los estudios encarados por la IR se fue dando paulatinamente. Sin embargo, a partir de la década de 1990, los esfuerzos se concentraron en la búsqueda de técnicas, métodos y herramientas que pudieran ser aplicados durante el proceso de definición de requerimientos para arribar a una etapa de diseño exitosa, dejando de lado la obtención de una metodología capaz de adaptarse a cualquier tipo de sistema y paradigma, brindando un marco de trabajo referencial, independiente del método a aplicar. [34]

De la metodología DorCu se puede destacar que:

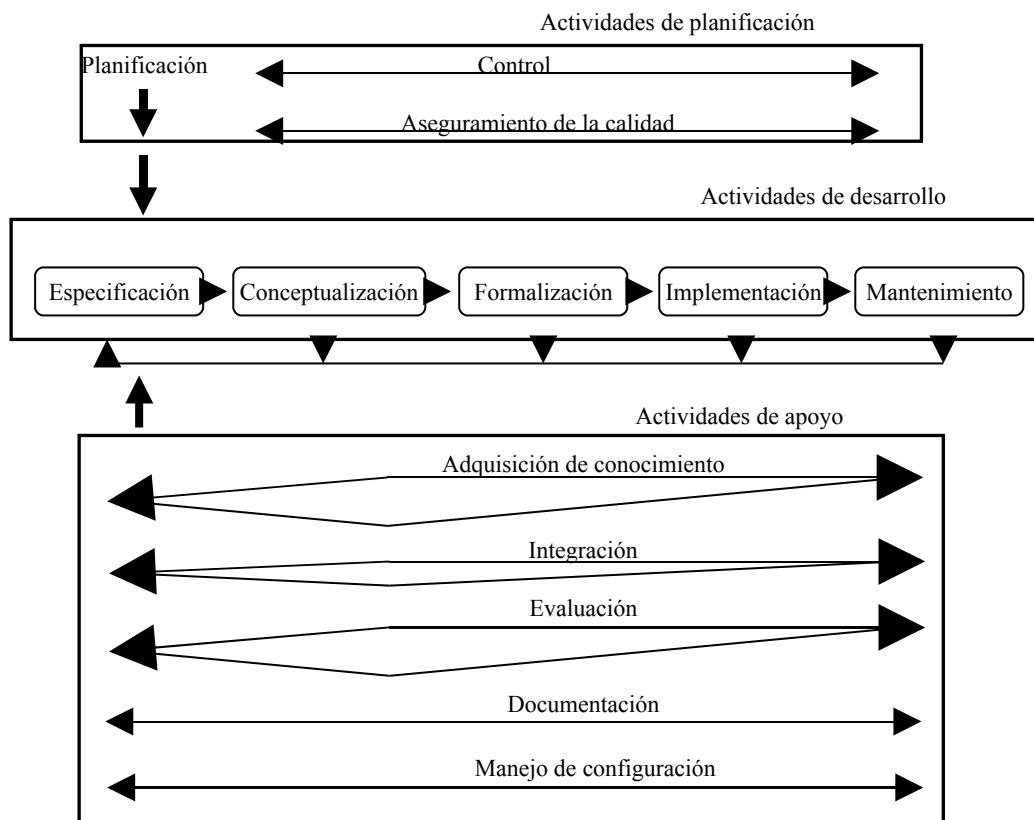
- Contribuye al entendimiento de la IR detallando etapas bien definidas, con lo que disminuyen los problemas existentes tanto en lo que hace a la terminología como a las actividades por ella involucradas.
- Ofrece libertad de acción para realizar la selección e integración de las herramientas a emplear en cada una de las etapas.
- Contempla aspectos metodológicos de documentación tendientes a conseguir un Documento de requerimientos de clara interpretación.
- Por ser una metodología flexible, puede ser aplicada a diferentes paradigmas con el uso de métodos, técnicas y herramientas que se crean convenientes para cada etapa.
- Cubre las falencias en cuanto a la documentación orientada al usuario, al proponer documentos isomórficos (DRt y DRu), destacando además la importancia de la validación y certificación de los documentos de requerimientos.

## II.3. MARCO METODOLOGICO

### II.3.1. Methontology [46]

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

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



**Figura 2.3.** Tareas de la actividad de Conceptualización según Methontology. (Corcho y otros, 2005)

El ciclo de vida de Methontology se muestra en la Figura 2.3. Las actividades de control, aseguramiento de calidad, adquisición de conocimiento, integración,

evaluación documentación y manejo de configuración se realizan simultáneamente con las actividades de desarrollo. La conceptualización debe ser evaluada cuidadosamente para evitar la propagación de errores a las siguientes etapas del ciclo de vida de la ontología.

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

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

Las actividades de desarrollo identificadas para Methontology son las siguientes:

**a) Especificación**

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

**b) Conceptualización**

Consiste en organizar y convertir una percepción informal de un dominio en una especificación semi-formal usando un conjunto de representaciones intermedias (tablas, diagramas) que puedan ser entendidas por los expertos del dominio y los desarrolladores de ontologías. En la Figura 2.4 se muestran los componentes de la ontología (conceptos, atributos, relaciones, constantes, axiomas formales, reglas e instancias) construidos en cada tarea, y se ilustra el orden propuesto para crear tales componentes durante la actividad de conceptualización. Las tareas de conceptualización en Methontology se describen a continuación:

- **Tarea 1: Construir el glosario de términos.** El glosario de términos debe incluir todos los términos relevantes del dominio (conceptos, instancias, atributos, relaciones entre conceptos, etc.), sus descripciones en lenguaje natural, sus sinónimos y acrónimos.
- **Tarea 2: Construir la taxonomía de conceptos.** Cuando el glosario de términos tenga una cantidad importante de elementos, se debe construir una taxonomía que defina la jerarquía entre los conceptos. Se debe evaluar que la taxonomía creada no contenga errores.

- **Tarea 3: Construir un diagrama de relaciones binarias.** El objetivo de este diagrama es establecer las relaciones entre los conceptos de una o más taxonomías de conceptos. Se debe evaluar que el diagrama creado no contenga errores.
- **Tarea 4: Construir el diccionario de conceptos.** El diccionario de conceptos contiene los conceptos del dominio, sus relaciones, instancias, atributos de clases y atributos de instancias. Las relaciones, atributos de instancias y atributos de clases son locales al concepto, lo que significa que sus nombres pueden repetirse en diferentes conceptos.
- **Tarea 5: Definir las relaciones binarias en detalle.** Se crea la tabla de relaciones binarias en la que se describe detalladamente todas las relaciones binarias incluidas en el diccionario de conceptos. Para cada relación binaria se debe especificar: nombre, conceptos fuente y destino, cardinalidad y relación inversa.
- **Tarea 6: Definir los atributos de instancia en detalle.** Se crea la tabla de atributos de instancias en la que se describe detalladamente todos los atributos de instancias incluidos en el diccionario de conceptos. Los atributos de instancias son aquellos atributos que describen las instancias de un concepto, y sus valores pueden ser diferentes para cada instancia del concepto. Para cada atributo de instancia se debe especificar: nombre, concepto al que pertenece, tipo de valor, rango de valores (en el caso de valores numéricos) y cardinalidad.
- **Tarea 7: Definir los atributos de clases en detalle.** Se crea la tabla de atributos de clases en la que se describe detalladamente todos los atributos de clases incluidos en el diccionario de conceptos. Para cada atributo de clase se debe especificar: nombre, concepto donde es definido, tipo de valor, valor y cardinalidad.
- **Tarea 8: Definir las constantes en detalle.** Se crea la tabla de constantes en la que se describe detalladamente cada una de las constantes definidas en el glosario de términos. Para cada constante, se debe especificar: nombre, tipo de valor, valor y unidad de medida (para constantes numéricas).
- **Tarea 9: Definir los axiomas formales.** Se deben identificar los axiomas formales necesarios en la ontología y describirlos con precisión en una tabla. Para cada definición de axioma formal de debe especificar: nombre, descripción, expresión lógica que formalmente lo describe (preferiblemente utilizando lógica



de primer grado), los conceptos, atributos y relaciones binarias a las cuales el axioma hace referencia y las variables utilizadas.

- **Tarea 10: Definir las reglas.** Se deben identificar cuáles reglas son necesarias en la ontología y describirlas en una tabla de reglas. Para cada regla, se debe especificar: nombre, descripción, expresión que formalmente la describe, los conceptos, los atributos y las relaciones a los que hace referencia y las variables usadas en la expresión. Para la especificación de las reglas se sugiere la forma: Si <condiciones> entonces <consecuencias o acciones>.
  - **Tarea 11: Definir las instancias.** Una vez que el modelo conceptual de la ontología ha sido creado, se deben definir las instancias relevantes que aparecen en el diccionario de conceptos en una tabla de instancias. Para cada instancia se debe especificar: nombre, concepto al que pertenece y valores de los atributos.
- c) **Formalización:** Realizar la transformación del modelo conceptual en un modelo formal o semi-computable.
- d) **Implementación:** Realizar la codificación de la ontología utilizando un lenguaje formal (Ontolingua, XOL, OIL, DAML, OWL, entre otros).
- e) **Mantenimiento:** Esta actividad permite la actualización y corrección de la ontología.

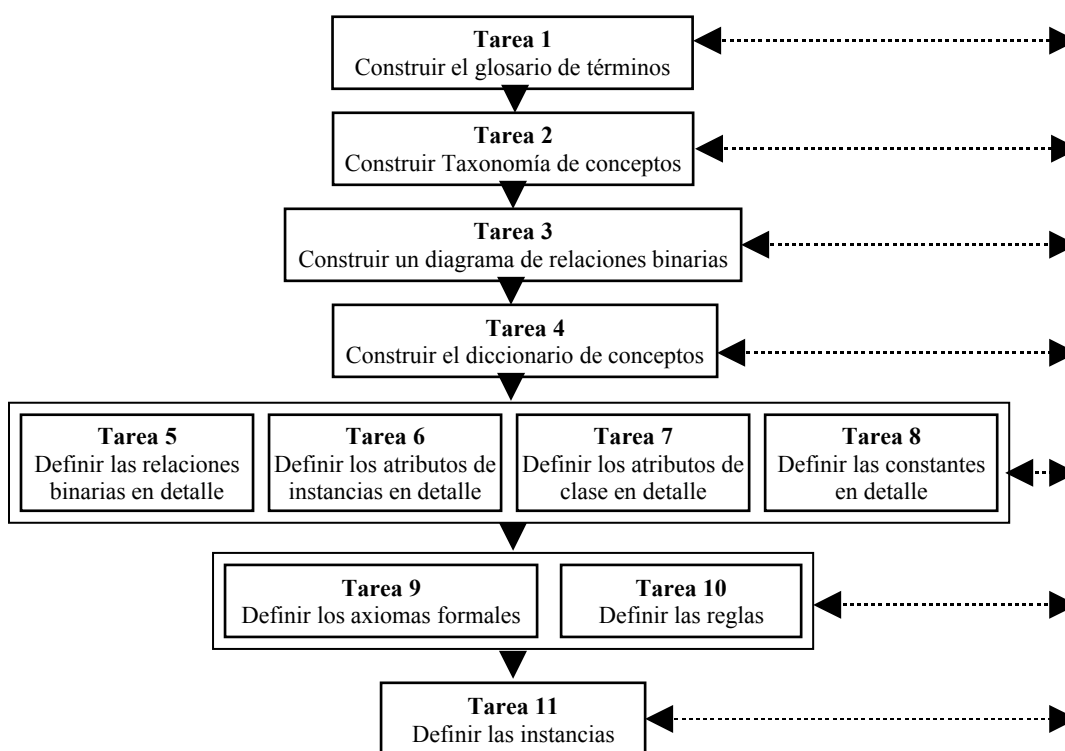


Figura 2.4. Tareas de la actividad de Conceptualización según Methontology.

### II.3.2. ADQUISICION DE CONOCIMIENTOS

La adquisición de conocimientos (AC), es el proceso de recolección de información, a partir de cualquier fuente. Aceptando la palabra información en su sentido más amplio; es decir: datos, noticias y sobre todo conocimientos. [4] [68]

#### **Fuentes de conocimientos**

La información necesaria puede encontrarse en múltiples y variadas fuentes. Entre estas pueden destacarse las siguientes [4] [68]:

- a)- Libros y manuales: los libros son útiles para obtener conocimientos básicos del dominio y las clasificaciones existentes en él. Es decir los libros contienen conocimientos específicos y públicos del dominio y del problema.
- b)- Documentación formal: se trata de aquellos documentos que contienen las políticas y procedimientos, los estándares, las normas y regulaciones, las leyes, etc. de un dominio. De esta fuente puede obtenerse unos conocimientos muy específicos, generalmente de nivel básico: procedimientos estándares de resolución, normas a cumplir al resolver un caso, etc. Los conocimientos proporcionados por esta fuente también son de carácter público.
- c)- Documentación informal: también como fuente de información en forma escrita, aparecen las notas manuscritas, “memos” internas, ayudas de trabajo, etc. que circulan dentro de las organizaciones. Esta documentación proporciona, a menudo, conocimientos heurísticos para resolución de problemas.
- d)- Presentaciones: hace referencia al material utilizado para formación, ya sea impartida o recibida, suele contener conocimientos expuestos de un modo especialmente claro.
- e)- Publicaciones especializadas: las versiones más actualizadas de los conocimientos de un dominio se encuentran expresadas, también de forma escrita en las publicaciones especializadas. En este caso, en forma de revistas, actas de congresos, etc.
- f)- Investigación: otra fuente, también de informaciones actualizadas, son los resultados de las investigaciones que se estén llevando a cabo. En concreto, en forma de: datos empíricos, estudios, informes, resultados estadísticos, etc.
- g)- Visitas: una fuente de conocimientos que suele ser muy útil para clarificar las ideas del ingeniero de conocimiento, son las visitas a los centros de trabajo del experto.

h)- Humanos: junto con los expertos, los directivos y usuarios finales son también fuentes para la extracción de conocimientos.

Finalmente, cuando la fuente de conocimientos se presenta en forma escrita, la adquisición se llamará **Extracción de conocimientos**. Si, por el contrario, los conocimientos se obtienen de seres humanos, el proceso se denominará **Educación de conocimientos**.

### **II.3.2.1. Proceso de adquisición de conocimientos**

El problema de AC consiste en extraer los conocimientos correctos y obtener la cantidad y tipo adecuados de conocimientos en el momento apropiado. Es por esto que el ingeniero de conocimientos debe controlar el proceso de adquisición constantemente, determinado en cada instante que información necesita, en que profundidad, sobre que tema, cual técnica debe emplear para adquirirlo, etc. [68]

### **II.3.2.2. Extracción de conocimientos**

- Estudio de Documentación [68]:

Muy a menudo, parte de los conocimientos relevantes está presente en manuales, libros u otra forma escrita. La documentación debe estar formada por libros básicos o introductorios al dominio y por manuales, catálogos y cualquier otro material escrito que proporcione el experto.

Cuando el ingeniero en conocimiento es capaz de obtener información a partir de la documentación, con o sin ayuda externa, el tiempo total de AC disminuye drásticamente.

Dentro de la documentación son especialmente interesantes los libros de textos introductorios o manuales de formación que definan los términos básicos usados en el dominio. A medida que el ingeniero de conocimiento lee este material de preparación, debe buscar principalmente definiciones de la terminología y conceptos que el experto probablemente usará.

- Análisis Estructural de Textos [68]:

En general, puede decirse que no existen técnicas o herramientas ampliamente aceptadas para llevar a cabo, de modo automático, el análisis de la documentación que necesita realizar el ingeniero de conocimiento. Sin embargo, las escasas técnicas existentes poseen una filosofía subyacente común. Dicha filosofía consiste en la búsqueda, a través de la documentación, de determinados términos.

Las técnicas pueden agruparse en tres grandes familias, dependiendo de los términos que se rastrean:

- a)- Técnicas en las cuales los términos son determinados por el ingeniero de conocimiento “en tiempo de ejecución”. Dicho de otro modo, el ingeniero de conocimiento, ante el análisis de un determinado texto, que constituye una parte de la documentación proporcionada, establece una serie de términos que deben ser buscados automáticamente, o no, en el texto.
- b)- Existe otra familia de técnicas en las que los términos a buscar están preestablecidos por las técnicas y son dependientes del dominio.
- c)- Finalmente, existe un tercer tipo de técnica de análisis de textos donde los términos a buscar están también preestablecidos, pero ahora, son independientes del dominio. En este caso, la técnica establece que los términos a buscar pueden ser: se define como, está relacionado con, es una característica de, etc. Los conocimientos a extraer son del tipo: conceptos, definiciones de terminología y relaciones entre conceptos.

Nótese que para las técnicas pertenecientes a la familia primera y segunda los conceptos están ya extraídos “a priori”. Una posibilidad es que los conozca el ingeniero de conocimiento, y por eso decida que éstos son los términos a buscar a través del texto. En la otra posibilidad, el ingeniero de conocimiento desconoce los conceptos del dominio, pero estos son establecidos por la técnica. Quizás, en este último caso, sería más apropiado hablar de clases de conceptos o conceptos generales. Esto significa que estos dos tipos de técnicas no extraen conceptos de la documentación puesto que los términos son ya conocidos antes de aplicar la técnica, sino más bien extraen definiciones o, sobre todo, relaciones entre los conceptos previamente establecidos.

En estos casos la extracción de conocimiento llevada a cabo consiste en:

- Relacionar el término que se estaba rastreando, con todas las palabras de la frase donde el término ha sido encontrado.
- Aceptar la frase como definición del concepto, si el ingeniero de conocimiento así lo estima.

Las técnicas pertenecientes a la tercera familia realizan extracción tanto de conceptos como de relaciones entre los mismos. Estas técnicas suelen denominarse de análisis estructural de textos, dado que extraen conceptos fundamentales del dominio buscando estructuras preestablecidas. La técnica sabe que los conceptos vienen

introducidos en estructuras de tipo definición, o que las relaciones entre conceptos vienen establecidas en estructuras de tipo afirmación relacional. Esto significa que para funcionar, el análisis estructural de textos necesita:

- a)- Tener descritas las estructuras textuales interesantes (definición, afirmación, etc.) y el tipo de conocimiento que aportan (concepto, relación, característica, valor, etc.)

*Estructuras textuales:* la técnica de análisis estructural de textos defiende la existencia de cuatro estructuras fundamentales encargadas de transmitir conocimientos en los textos:

*Definiciones:* introducción de un concepto nuevo en el texto. El concepto puede venir definido en base a distintos criterios. En cada caso, el tipo de conocimientos a extraer es distinto.

*Afirmaciones:* una afirmación es una frase que establece una verdad. Para el objetivo de extraer conocimientos básicos, las afirmaciones que interesan son aquellas que presentan relaciones entre conceptos.

*Leyes:* las leyes de un dominio establecen sus principios básicos, así como las reglas que fijan el funcionamiento de objetivos del dominio. Esta estructura proporciona conocimientos de un nivel más alto que el que se pretende extraer de los textos.

*Procedimientos:* los procedimientos de un dominio establecen los pasos para la resolución de problemas en el dominio. Los conocimientos proporcionados por esta estructura están más allá del objetivo de la extracción de conocimiento a partir de la documentación.

- b)- Tener descrito el modo de detectar las estructuras en el texto.

*Detección:* la técnica de análisis estructural de textos defiende que las estructuras textuales vienen embebidas en patrones independientes del dominio. Se hace necesario, por tanto, definir una lista de patrones de patrones para cada tipo de estructura que se quiera analizar. Ejemplos de patrones son:

- Para las definiciones: A se usa para B; A es un B que C; A está compuesta por B, C,... etc.
- Para las afirmaciones: A es causa de B; A se relaciona con B; A es la finalidad de B, etc.

El trabajo del ingeniero de conocimiento consistirá en:

- 1)-Hojear el texto deteniéndose sólo en las definiciones y afirmaciones de relación. Estas frases serán señaladas.
- 2)-Analizar las frases señaladas para extraer los conocimientos buscados: conceptos relaciones y definiciones de conceptos. [4]

### II.3.3. MARCOS

El sistema de representación basado en marcos es un soporte excelente para la representación de conocimiento del mundo / dominio. [41]

Los *marcos* o *frames* fueron propuestos por Minsky en 1975 como una forma de representación del conocimiento. La idea original se expresaba en los siguientes términos [41]:

"Un *frame* es una estructura de datos para representar una situación estereotipada... Podemos pensar en un Frame como una red de nodos y relaciones.

Los 'niveles superiores' de un Frame son fijos y representan cosas que son siempre ciertas en la situación supuesta. Los niveles inferiores tienen muchos terminales - *slot*' que tienen que llenarse con instancias o datos específicos. Las suposiciones por defecto son asociadas débilmente con sus terminales, de forma tal que ellas puedan ser fácilmente removidas ante nuevos elementos que se ajusten mejor a la situación actual."

Por lo tanto, como resalta Reiter, un *marco* es "una colección de condiciones necesarias y suficientes sobre un concepto estereotipado".

Informalmente, un marco es una estructura de datos compleja que representa una situación estereotipada, por ejemplo hacer una visita a un enfermo o acudir a una fiesta de cumpleaños. Rich plantea que cada marco posee un número de casillas (*slots*) donde se almacena la información respecto a su uso y a lo que se espera que ocurra a continuación, es decir cada *slot* contiene la información sobre un atributo del objeto que se modela, se le puede asociar varios tipos de información, llamados facetas del *slot*, entre las cuales están [41]:

- Value: almacena el valor para el *slot*.
- Procedure: contiene un procedimiento para calcular el valor que debe ser almacenado en la faceta Value.
- Demons: contiene procedimientos que tienen que ser ejecutados cuando cambia el valor almacenado en Value (mantienen la consistencia interna de la información).

- Default: contiene un valor inicial o valores usados comúnmente para la faceta Value.
- Restrictions: contiene un conjunto de expresiones lógicas que tienen que ser verdaderas para el valor almacenado en Value.
- Explanation: almacena documentación sobre el slot.
- Univaluado: Verdadero, si el atributo toma un valor único y falso en caso contrario.

Los marcos organizan los conocimientos en árboles (también llamados jerarquías) o en grafos. Hay dos tipos de marcos: marcos *clase* y marcos *instancia*.

Los marcos clase se utilizan para representar conceptos, clases o situaciones genéricas descritos por un conjunto de propiedades. [43]

Ejemplos son los marcos Persona, Hombre, Mujer. En los dominios de trabajo, existen elementos o instancias de clases. Por ejemplo, Ana, Luis y Pedro son marcos instanciados de la clase Mujer y Hombre. Las relaciones entre conceptos se representan con relaciones entre marcos, siendo las relaciones más comunes: subclase de y superclase de, instancia y representa, disjunto, no disjunto, fraternal, “ad hoc” o a medida (ejemplos: casado-con, divorciado-de). [43]

Los marcos tienen propiedades de clase y de instancia. Las de clase representan atributos o características de un concepto o clase. Toman siempre el mismo valor en todos los elementos o instancias de la clase. Las propiedades de instancia se rellenan en cada instancia con valores concretos, que dependen del elemento de la clase que se esté representando. [43]

Luego de identificadas las propiedades de los marcos clase, se describen las facetas de cada una de esas propiedades. Las facetas pueden ser:

- 1) facetas que definen propiedades de clase, de instancia y relación, por ejemplo tipo ranura, cardinalidad máxima y mínima,
- 2) faceta que define propiedades de clase y relación; por ejemplo la llamada propiedad general y
- 3) facetas que definen propiedades de instancia. Las más comunes son: valores permitidos y valores por omisión. [43]

Se pueden resumir las principales características de los marcos en las siguientes [41]:

- Precisión (*Explicitness*): los objetos, las relaciones entre objetos y sus propiedades se describen de forma precisa; en ausencia de evidencia contraria se usan valores por omisión.
- Activación dinámica de procesos (*Triggering*): es posible adjuntar procedimientos a un marco o alguno de sus componentes de forma que se llamen y ejecuten automáticamente tras la comprobación de cambio de alguna propiedad o valor (p.ej. IF-NEEDED, IF-ADDED).
- Herencia por defecto no-monotónica: los marcos están conceptualmente relacionados, permitiendo que los atributos de los objetos sean heredados de otros objetos predecesores en la jerarquía.
- Modularidad: la base de conocimiento está organizada en componentes claramente diferenciados.

### **II.3.4. METODOLOGIA DE TRABAJO PROPUESTA**

Las fases y etapas que conforman la metodología de trabajo para alcanzar los objetivos son las siguientes:

#### **1. Estudio inicial de la situación actual.**

*Descripción:*

Se realizará una exploración bibliográfica con el fin de obtener información con respecto a la problemática existente en el área considerada. Asimismo, esto se llevará a cabo para conocer las características principales y el vocabulario propio del dominio del problema. Además, se obtendrá información referente a metodologías de construcción de ontologías, herramienta de desarrollo, Protégé y el lenguaje OWL.

*Técnica y herramientas:*

- Recopilación y análisis de textos.

#### **2. Identificación del problema/ Definición de los objetivos del trabajo**

*Descripción:*

A partir de la información obtenida en la etapa anterior, en esta tarea se planteara el problema que se quiere dar solución, además se identificarán los objetivos que se esperan alcanzar una vez concluido el trabajo; los mismos servirán de guía a lo largo del desarrollo de la misma.



*Técnicas y herramientas:*

- Revisión de la información obtenida en la etapa anterior.

### **3. Construcción de la ontología**

En base a la metodología para construcción de ontologías elegida como guía, proponemos las siguientes actividades:

#### **3.1. Especificación de requisitos:**

*Descripción:*

Se identifica el objetivo de la ontología, su dominio y los requisitos que debe satisfacer.

*Técnicas y herramientas:*

- Adquisición de conocimiento.
- Especificación de requisitos.
- Glosarios de términos.

#### **3.2. Conceptualización:**

*Descripción:*

Se estructura el conocimiento de dominio como un modelo significativo al nivel de conocimiento. Se desarrolla una presentación coherente y organizada de manera que satisfaga las especificaciones de requisitos.

**3.2.1. Definir los conceptos y las relaciones binarias:** de los glosarios de términos obtenidos en la etapa anterior se seleccionan aquellos términos independientes que representan los conceptos y se establecen las relaciones existentes entre ellos. Transforma el conocimiento del dominio a una forma más utilizable y entendible.

*Técnicas y herramientas:*

- Taxonomía de conceptos.
- Diagrama de relaciones binarias.
- Relaciones causales (reglas).

**3.2.2. Definir las clases, propiedades de las clases (slots) y sus facetas.** A partir de la taxonomía de conceptos y del diagrama de relaciones binarias se identifican las clases. Se definen las facetas de las propiedades 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.

*Técnicas y herramientas:*

- Tabla clase - subclase - propiedad.
- Relaciones Binarias en detalle.

**3.2.3. Definir la jerarquía de clases.** A partir de 3.2.2 se organiza la jerarquía de clases.

*Técnicas y herramientas:*

- Diagrama de clases.
- Esquema Conceptual.

**3.2.4. Definir axiomas.** Se definen claramente los axiomas que compondrán la Ontología.

*Técnicas y herramientas:*

- Definición de axiomas formales.

### **3.3. Implementación:**

*Descripción:*

La ontología es formalmente representada en un lenguaje computacional (OWL).

*Técnicas y herramientas:*

- Jerarquía de marcos.
- Técnicas de programación.
- Herramientas Protégé, OWL y Racer.

### **3.4. Prueba de la ontología**

*Descripción:*

Se verifica que la ontología arroje los resultados esperados para ver si la misma cumple con el propósito y las funcionalidades establecidas.

Se propondrá el uso de la ontología a un desarrollador, en un caso real, para guiarlo durante el proceso de IR, con el fin de probar si la misma podría optimizar dicho proceso. Por otro lado, se realizará un cuestionario para determinar si al desarrollador le fue útil la ontología o no.

El desarrollador seleccionado deberá acreditar más de tres años de experiencia en el desarrollo de sistemas software. Se considerará antecedentes laborales, sistemas desarrollados, participación en proyectos de desarrollo de los mismos, etc.

Al finalizar la etapa, se confeccionará la documentación referente al proceso de desarrollo de la ontología.

*Técnicas y herramientas:*

- Confección de casos de prueba (con distintas fuentes de información).
- Cuestionarios.
- Plan de documentación.
- Aspectos en base a los cuales se analizará la ontología:
  - Grado de aceptación de la ontología por el experto: considera si todos los componentes, clases, relaciones y axiomas de la ontología son suficientes para ayudar al proceso de IR.
  - Tiempo de respuesta: tiempo en que la ontología tarda en inferir un resultado (Racer).
  - Grado de concordancia entre resultados esperados y obtenidos.

#### **4. Análisis de los Resultados y Confección del informe final**

*Descripción:*

Se analizarán los resultados, con el objeto de llegar a las conclusiones de la investigación. Se confeccionará un documento donde se detallarán los pasos seguidos durante la misma, los resultados obtenidos y las conclusiones arribadas.

*Técnicas y herramientas:*

- Plan de documentación.
- Racer

Se evaluarán los indicadores de medición de los objetivos específicos, para verificar si se alcanzaron los mismos. Los indicadores a tener en cuenta son:

- Vocabulario/ Glosario de términos de la ontología y como fuentes de validación se considerarán la opinión de los expertos (cuestionarios), documentos y estándares.
- Métodos y técnicas ofrecidos por la ontología para cada etapa de la IR; tiempo de ejecución de la ontología para cada caso de prueba. Como fuente de validación se contará con los casos de prueba propuestos por los desarrolladores que ejecutarán la ontología, por otro lado se realizará una encuesta a desarrolladores/expertos en IR.
- Indicadores de calidad de los procesos de IR usados en la construcción de relaciones y axiomas de la ontología.

### **II.3.5. ENTORNO DE DESARROLLO. HERRAMIENTAS PARA CONSTRUCCION DE LA ONTOLOGIA**

#### **II.3.5.1. Protégé**

Protégé es una plataforma libre, de código abierto, que proporciona a una creciente comunidad de usuarios un conjunto de herramientas para construir aplicaciones basadas en conocimiento y modelos de dominio. En este punto, Protégé implementa un gran conjunto de estructuras de modelado de conocimiento y acciones que soportan la creación, visualización y manipulación de ontologías en varios formatos de representación. Protégé puede ser personalizado para proveer un dominio amigable de apoyo a la creación de modelos de conocimientos e ingreso de datos. Además Protégé puede ser extendido por medio de una arquitectura de plug-in y una Interfaz de Programación de Aplicaciones (API) basada en Java para la construcción de aplicaciones y herramientas basadas en conocimiento. [51]

Una ontología describe los conceptos y relaciones que son importantes en un dominio en particular, proporciona un vocabulario para ese dominio así como una especificación computarizada del significado de los términos utilizados en el vocabulario. Las ontologías van desde taxonomías y clasificaciones, esquemas de bases de datos, a una teorías automatizadas completamente. En los últimos años las ontologías han sido adoptadas por comunidades científicas y de negocio como una forma para compartir, reusar y procesar el conocimiento de dominio. Las ontologías son ahora fundamentales para muchas aplicaciones como portales de conocimiento

científico, administración de información e integración de sistemas, comercio electrónico y servicios de la web semántica. [51]

La plataforma de Protégé soporta dos principales formas para modelado de ontologías:

El *editor de marcos de Protégé* permite a usuarios construir y rellenar ontologías que se basan en marcos, de acuerdo con el protocolo de conectividad de base de conocimientos abierto. En este modelo, una ontología consiste de un conjunto de clases organizado en una jerarquía para representar los principales conceptos de un dominio, un conjunto de slots asociados a clases para describir sus propiedades y relaciones, y un conjunto de instancias de esas clases. [51]

El *editor Protégé OWL* permite a los usuarios construir ontologías para la web semántica. Una ontología puede incluir descripciones de clases, propiedades y sus instancias. La semántica formal de OWL especifica como derivar sus consecuencias lógicas. [51]

### **II.3.5.2. Protégé OWL**

El editor Protégé OWL es una extensión del Protégé que soporta el lenguaje de ontologías web (owl). OWL es el más reciente desarrollo en lenguaje de ontología Standard, aprobado por el consorcio de la World Wide Web (W3C) para promocionar la visión de la web semántica. [52]

El editor Protégé OWL permite a usuarios [52]:

- Leer y guardar ontologías OWL y RDF.
- Editar y visualizar clases, propiedades y reglas SWRL.
- Definir las características lógicas de las clases como expresiones owl.
- Ejecutar razonadores como una descripción de clasificadores lógicos.

### **II.3.5.3. Lenguaje OWL**

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 aquellos términos (ontologías). En realidad, OWL es una extensión del lenguaje RDF y emplea las tripletas de RDF, aunque es un lenguaje

con más poder expresivo que éste. 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. OWL surge como una revisión al lenguaje DAML-OIL y es mucho más potente que éste. 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.) [53]

OWL se diseñó para ser utilizado por aplicaciones que necesitan procesar el contenido de la información en lugar de sólo presentar la información a los seres humanos. OWL facilita una mayor interpretación de contenido Web que el soportado por XML, RDF, RDF y esquema (RDF-S) debido a que presenta un mayor vocabulario junto con una semántica formal. OWL tiene tres sublenguajes cada vez más expresivo: OWL Lite, OWL DL y OWL Full. [53]

#### II.3.5.3.1. Los tipos de OWL

El lenguaje owl ofrece tres sublenguajes cada vez más expresivos, diseñados para ser utilizados por comunidades específicas de implementadores y usuarios.

- **OWL Lite** es útil para la creación de jerarquías y restricciones simples. Por ejemplo, solo permite valores de cardinalidad 0 y 1.
- **OWL DL** Proporciona la máxima capacidad de expresión que garantiza computabilidad y decidibilidad (tiempo finito). Por ejemplo una clase puede ser subclase de muchas otras clases, pero no puede ser un caso de otra clase. OWL DL es llamado así debido a su correspondencia con la descripción lógica [Descripción Lógica], un campo de la investigación que ha estudiado un fragmento particular de la lógica de primer orden.
- **OWL Full** ofrece máximo nivel e expresión y la libertad sintáctica de RDF. No ofrece garantías computacionales. Por ejemplo una clase puede ser tratada simultáneamente como una colección de individuos y como un individuo por sí mismo. [54]

Cada uno de estos sublenguajes es una extensión de su predecesor más simple, tanto en lo que puede ser expresado legalmente como en lo que puede ser concluido validamente. [55]

OWL tiene componentes similares a los usados en Protégé. Sin embargo la terminología utilizada para describir estos componentes es diferente. OWL consiste de Individuos, Instancias y Clases, las que corresponde a Instancias, Slots y Clases en Protégé. [52]

Los individuos representan objetos dentro del dominio de interés, por ejemplo, países, personas o mascotas. Las propiedades son relaciones binarias entre los individuos. Las clases, son elementos que contienen un conjunto de individuos. Para que un elemento pertenezca a una clase determinada, debe cumplir con ciertas características que son los requerimientos de membresía de la clase. [52]

Otro componente importante es el OWL Viz, que permite ver la ontología en forma gráfica. [52]

OWL también permite realizar un análisis de consistencias y de la jerarquía de clases. Protégé no cuenta con esta capacidad, sin embargo, existen razonadores como Racer que permiten hacerlo desde la misma interfaz de Protégé. Así, por ejemplo, se puede encontrar que un individuo pertenece a dos clases que son disjuntas entre sí, lo que es una inconsistencia. [52]

#### **II.3.5.4. Razonamiento de una ontología. Herramienta Racer**

Para realizar el razonamiento de una ontología es necesario contar con una herramienta que actúe como razonador para Protégé. Existe una herramienta llamada Racer que se puede conectar con Protégé para el análisis y razonamiento de las ontologías. [72]

Racer se utiliza para tres actividades principales que son:

- Verificación de consistencia: permite controlar que existe consistencia entre las propiedades de objetos y restricciones asignadas a las clases.
- Clasificación taxonómica: encuentra la jerarquía de inferencia de la ontología basada en la jerarquía de aserción, la cual es la clasificación que se obtiene en la construcción de la ontología.
- Computación de tipos inferidos: permite computar los tipos de objetos dentro de la ontología.

### **II.3.7. CONCLUSIÓN**

En este capítulo se realizó la explicación en detalle de todos los temas involucrados en el desarrollo del presente trabajo.

Se especificaron los Marcos Teórico y Metodológico. En el primero, se delimitaron los temas del objeto de estudio; y en el segundo, se volcó información referida a las metodologías de construcción de ontologías, metodología de trabajo a seguir y herramientas utilizadas para el desarrollo de la ontología propuesta.



*Capítulo III:*

*Especificación  
De  
Requisitos*



### III.1. INTRODUCCIÓN

En el presente capítulo se especifica y acota el dominio de la ontología teniendo en cuenta el paso *3.1 Especificación de Requisitos*, del capítulo 2. Se especifican los requerimientos y se establecen los usos que se darán a la ontología.

A partir de esta información se procede a la adquisición de conocimientos (AC). La AC es el proceso de recolección de información, a partir de cualquier fuente (expertos, libros, revistas, informes, etc.), necesaria para construir un sistema basado en conocimiento. [68]

La AC se divide en dos fases: extracción de conocimiento y educación de conocimiento.

La extracción de conocimiento pretende obtener una amplia gama de conocimiento del ámbito donde se desarrolla la tarea. La fuente de conocimiento se presenta en forma escrita. [4]

No existe ningún método completamente automático de AC. Por estas razones, la AC es, en la actualidad, una labor de artesanía hecha a medida en cada caso, de modo que pueda ajustarse a las características de cada situación particular y de las personas concretas (expertos e ingenieros) involucrados en ella. [4]

El Ingeniero de Conocimiento (IC) debe controlar constantemente cuánto conocimiento desea adquirir y con qué profundidad. El esquema que el IC posee del dominio debe estar formado de lo general a lo particular. [68]

### III.2. DESCRIPCIÓN DEL DOMINIO DE LA ONTOLOGÍA

Este trabajo propone el desarrollo de una ontología para la IR, basada en la metodología DoRCu, con la cual se pretende mejorar dicho proceso.

La ontología abarcará las diferentes etapas y subetapas que componen el proceso de IR consideradas por la metodología antes mencionada. Esas etapas son:

- Elicitación de requerimientos.
- Análisis de requerimientos.
- Especificación de requerimientos.
- Validación y certificación de requerimientos.

### **III.3. DEFINICIÓN DE LOS OBJETIVOS DE LA ONTOLOGÍA**

Los objetivos definidos para la ontología se describen a continuación.

#### **III.3.1. Objetivos de OntoIR**

- a)- Lograr que los desarrolladores de un producto software utilicen una terminología común durante el proceso de IR.
- b)- Ofrecer a los desarrolladores la información necesaria acerca de los diferentes métodos y técnicas competentes a la IR, para realizar una elección adecuada entre los mismos.
- c)- Garantizar la calidad del proceso de IR, mediante el uso de la herramienta propuesta, de tal manera que disminuyan los riesgos en que se incurren cuando se lleva a cabo de forma inadecuada.
- d)- Asistir a desarrolladores durante las actividades competentes al proceso de IR.

### **III.4. ESPECIFICACIÓN DE REQUISITOS DE LA ONTOLOGÍA**

En la etapa de especificación de requisitos se describen la meta de la ontología, los usuarios finales, los objetivos y los requisitos funcionales y de información de la misma.

A continuación se muestra la especificación resultante:

#### **1. Meta de la ontología**

Proporcionar una herramienta de calidad que asesore a desarrolladores de productos software, a lo largo del proceso de IR, con el objeto de asegurar la calidad del mismo.

#### **2. Usuarios finales / potenciales**

La ontología esta dirigida para analistas o desarrolladores de software.

#### **3. Requisitos de Información**

<b>RI-01</b>	Información sobre etapas del proceso de IR
Objetivos relacionados	<ul style="list-style-type: none"> <li>_ OBJ-a)</li> <li>_ OBJ-c)</li> <li>_ OBJ-d)</li> </ul>
Descripción	La ontología contendrá información referente a las diferentes etapas que componen el proceso de IR.
Datos específicos	<ul style="list-style-type: none"> <li>_ Etapas que componen el proceso de IR.</li> <li>_ Sub etapas que componen cada etapa del proceso de IR.</li> <li>_ Orden en que se ejecutarán las etapas.</li> <li>_ Documentación producida por cada etapa.</li> </ul>

<b>RI-02</b>	Información sobre técnicas y herramientas útiles al proceso de IR
Objetivos relacionados	<ul style="list-style-type: none"> <li>_ OBJ-b)</li> </ul>
Descripción	La ontología contendrá información referente a las diferentes técnicas y/o herramientas que pueden ser de utilidad en cada etapa del proceso de IR.
Datos específicos	<ul style="list-style-type: none"> <li>_ Nombre de la técnica.</li> <li>_ Etapa para la cual, puede ser de utilidad la técnica.</li> <li>_ Nombre de la herramienta.</li> <li>_ Etapa para la cual puede ser de utilidad la herramienta.</li> </ul>

<b>RI-03</b>	Información sobre normas de redacción y/o estándares vigentes
Objetivos relacionados	<ul style="list-style-type: none"> <li>_ OBJ-c)</li> <li>_ OBJ-d)</li> </ul>
Descripción	La ontología contendrá información referente a los diferentes estándares vigentes, de manera tal que los documentos obtenidos al finalizar cada etapa, se ajusten a ellos para asegurar la calidad de los mismos.
Datos específicos	<ul style="list-style-type: none"> <li>_ Nombre del estándar.</li> <li>_ Momento para ser considerado.</li> </ul>

<b>RI-04</b>	Información sobre actores
Objetivos relacionados	<ul style="list-style-type: none"> <li>_ OBJ-c)</li> </ul>
Descripción	La ontología contendrá información referente a las diferentes personas que pueden estar involucradas en el proceso de IR.
Datos específicos	<ul style="list-style-type: none"> <li>_ Tipo de actor.</li> <li>_ Etapa en la cual puede estar involucrado, o en la cual tenerlo en cuenta.</li> </ul>

#### 4. Requisitos funcionales

El sistema deberá satisfacer cada uno de los requisitos funcionales planteados a continuación:

RF1: Aceptar el ingreso de datos referentes a diferentes etapas.

RF2: Inferir sobre los siguientes temas referentes a la Ingeniería de Requerimientos:

RF2.1: Etapas del proceso de ingeniería de requerimientos y orden de ejecución de las mismas.

RF2.2: Técnicas útiles a aplicar en cada etapa.

RF2.3: Herramientas a utilizar para optimizar la realización de cada etapa.

RF2.4: Documentos a obtener en cada etapa.

RF2.5: Información útil para la confección de cada documento (estándares vigentes, contenido de cada documento).

RF2.6: Participantes de la etapa en cuestión.

### III.5. ADQUISICIÓN DE CONOCIMIENTO

La técnica que se utilizó para realizar la extracción de conocimiento para este proyecto fue el Análisis Estructural de Textos (AET).

Se procedió a estudiar la documentación disponible, cuyo objetivo fue aprender sobre el universo de estudio, lo cual favoreció a la asimilación del conocimiento y permitió establecer una visión profunda del dominio del problema.

#### III.5.1. Análisis Estructural de Textos

Mediante el AET se extraen conceptos fundamentales del dominio buscando estructuras preestablecidas.

La técnica se basa en que los conceptos se encuentran en estructuras de tipo definición, y las relaciones entre conceptos en estructuras de tipo afirmación relacional. Esto significa que para funcionar, el AET necesita [4]:

- Buscar estructuras textuales mediante la detección de patrones: definiciones de conceptos (se usa para, es un, está compuesto por), afirmaciones sobre relaciones (es causa de, se relaciona con, es la finalidad de), principios básicos de funcionamiento, pasos.
- Buscar términos dependientes del dominio.

- Tener descritas las estructuras textuales interesantes (definición, afirmación, etc.) y el tipo de conocimiento que aportan (concepto, relación, características, valor, etc.).
- Tener descrito el modo de detectar las estructuras en el texto.
- Los resultados de la aplicación de la técnica se vuelcan en glosarios de términos (Tabla 3.1 a Tabla 3.7), con los cuales se creará el Documento de Extracción de Conocimiento que contendrá toda la información obtenida referente al dominio. Los glosarios de términos permiten establecer un lenguaje común, eliminando ambigüedades que pudiesen ocurrir al ser interpretados o manipulados determinados términos.

El AET se basó en el estudio de documentación referida a la Metodología DorCu para la Ingeniería de Requerimientos [34]; Libro Ingeniería de Software, 5ª Ed. [13]; Libro Ingeniería de Software, 7a. Ed. [3]; y demás documentación de dominio público sugerida por expertos en el tema.

**Tabla 3.1.** Etapas del proceso de IR.

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>ELICITACIÓN DE REQUERIMIENTOS</b>	Etapa en donde se adquiere el conocimiento del trabajo del cliente/usuario. Se busca comprender sus necesidades y se detallan las restricciones medioambientales. Como resultado de las acciones realizadas se tiene el conjunto de los requerimientos de todas las partes involucradas.	[34]
<b>ANÁLISIS DE REQUERIMIENTOS</b>	Etapa en la que se estudian los requisitos extraídos en la etapa previa a los efectos de poder detectar, entre otros, la presencia de áreas no especificadas, requerimientos contradictorios y peticiones que aparecen como difusas e irrelevantes. En esta etapa ya se realizan aproximaciones a un lenguaje técnico.	[34]
<b>ESPECIFICACIÓN DE REQUERIMIENTOS</b>	Esta etapa es un proceso de descripción de los requerimientos. Una descripción puede ser un documento escrito, un modelo gráfico, un modelo matemático formal, una colección de escenarios de uso, un prototipo o una combinación de los anteriormente citados. Si se presentan dificultades para especificar un requerimiento se debe volver a la etapa anterior que se crea conveniente.	[34] [13]

**Tabla 3.1.** Etapas del proceso de IR (continuación)

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>VALIDACIÓN Y CERTIFICACIÓN DE REQUERIMIENTOS</b>	<p>Esta etapa realiza la integración y la validación final de lo obtenido en cada una de las etapas anteriores logrando, como resultado final, el Documento de Requerimientos. Puede resultar necesario retornar a la especificación e incluso a la elicitación; iterando entre etapas y sin perder contacto con el cliente/usuario.</p> <p>La validación de requerimientos examina las especificaciones para asegurar que todos los requerimientos del sistema ha sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto</p>	[34] [13]

**Tabla 3.2.** Documentos obtenidos durante el Proceso de IR

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>DOCUMENTO DE REQUISITOS DEL SOFTWARE (DRS)</b>	El documento de requerimientos del software (a veces llamado especificación de requerimientos del software o SRS) es la declaración oficial de lo que deben implementar los desarrolladores del sistema. Debe incluir tanto los requerimientos del usuario para el sistema como una especificación detallada de los requisitos del sistema.	[3]
<b>DOCUMENTO DE ELICITACIÓN (DE)</b>	Contiene la lista final de los términos del lenguaje del Universo de Información y la lista de sentencias de los requerimientos obtenidos.	[34]
<b>DOCUMENTO DE ANÁLISIS (DA)</b>	Contiene todo lo que se considere adecuado como soporte para la etapa siguiente. Este documento puede resumirse a los modelos lógicos a los que se ha arribado.	[34]
<b>DOCUMENTO DE ESPECIFICACIÓN (DP)</b>	Se lo confecciona tomando como base a los modelos formales o semiformales que se han elaborado al realizar la especificación de los requerimientos. Se puede incorporar al mismo toda extensión que se considere de utilidad para la etapa de validación y certificación de requerimientos.	[34]
<b>DOCUMENTO DE REQUERIMIENTOS ORIENTADO AL USUARIO (DRU)</b>	Se elabora a partir del documento de requerimientos técnico, realizando una traducción a un lenguaje entendible por el usuario. Este documento se elabora teniendo en cuenta los estándares vigentes a la fecha de su confección (normas IRAM, ISO, estándares de la IEEE).	[34]
<b>DOCUMENTO DE REQUERIMIENTO TÉCNICO (DRT)</b>	Es el documento de especificación validado. La validación de dicho documento se hará a partir de los documentos obtenidos en las etapas de análisis y elicitación.	[34]

**Tabla 3.3.** Técnicas útiles para la IR

TÉRMINO	SIGNIFICADO	FUENTE
<b>ANÁLISIS DE LOS STAKEHOLDERS</b>	<p>Esta técnica es esencial para identificar los grupos de stakeholders y averiguar sus intereses.</p> <p>Durante el desarrollo de esta técnica se trata de responder a las siguientes preguntas:</p> <ul style="list-style-type: none"> <li>-Quiénes son los stakeholders?</li> <li>-Qué metas ven ellos para el sistema?</li> <li>-Por qué desean contribuir?</li> <li>-Qué riesgos y costos ven?</li> <li>-Qué clase de soluciones ven?</li> </ul>	[56]
<b>OBSERVACIÓN</b>	<p>Los usuarios no siempre son conscientes de lo que realmente hacen y cómo lo hacen. Una forma de eliminar esta ceguera es observar lo que realmente está pasando. El analista puede pasar algún tiempo con los usuarios, observando sus tareas diarias. En algunos casos, los analistas pueden utilizar las cámaras de vídeo (con los usuarios el permiso) para alargar el período de observación. Esto tiene la ventaja de que más tarde puede revisar las cintas con los usuarios y ver lo que realmente se realiza.</p> <p>La observación mejora enormemente su conocimiento sobre las tareas en curso y algunos problemas de trabajo asociados.</p>	[56]
<b>DEMOSTRACIÓN DE TAREAS</b>	<p>Es una variante de las entrevistas y la observación. Se les solicita a los usuarios que demuestren cómo realizan una tarea específica. En muchos casos, los usuarios no son capaces de explicar cómo realizan sus tareas diarias, pero sí son capaces de mostrar cómo realizan tareas específicas.</p>	[56]
<b>ESTUDIO DE DOCUMENTACIÓN</b>	<p>El estudio de la documentación es una forma rápida para obtener información de datos de viejas bases de datos. El analista estudia documentos existentes tales como formularios, archivos, documentación del sistema de computación existente. También se puede realizar impresiones de pantallas del sistema existente.</p>	[56]
<b>GRUPOS DE FOCO</b>	<p>Los grupos de foco se parecen a las sesiones de la técnica de tormenta de ideas, pero son más estructurados. Un grupo de foco inicia con una fase en donde los participantes tienen problemas en la forma actual de hacer las cosas. Luego, los participantes tratan de imaginar una forma ideal de hacer las cosas. El grupo trata también de explicar por qué esas ideas son buenas. Esto ayuda a formular metas y requerimientos para el nuevo sistema. Pueden participar varios grupos de stakeholders, y al final de la sesión, cada grupo identifica sus temas de alta prioridad.</p>	[56]
<b>WORKSHOPS DE DOMINIO</b>	<p>En un workshop de dominio el equipo mapea los procesos de negocio. El resultado puede ser una descripción de tareas, diagramas de flujo o diagramas de actividades que describen lo que se realiza en el dominio. Luego, el analista convierte estas descripciones en requerimientos.</p> <p>Es importante que los usuarios expertos participen en los workshops de dominio, puesto que conocen todos los detalles de negocio en su propio dominio. Los administradores pueden participar, raramente conocen los detalles reales de los procedimientos, y no pueden reemplazar a los usuarios expertos. Sin embargo los administradores pueden ser un instrumento para la definición de metas y visiones.</p>	[56]



**Tabla 3.3.** Técnicas útiles para la IR (continuación)

TÉRMINO	SIGNIFICADO	FUENTE
<b>WORKSHOPS DE DISEÑO</b>	<p>Los usuarios y desarrolladores de workshop de diseño cooperan en el diseño, generalmente, de la interfaz de usuario.</p> <p>Esta tipo de workshop es ampliamente usado, pero a menudo el resultado puede ser no conveniente. La razón de ello es debido a que los usuarios se avocan al diseño y a las cuestiones técnicas. Así, se puede perder de vista los objetivos de negocio.</p> <p>Si se usa este tipo de workshop, es fundamental que el equipo de vez en cuando controle la interfaz de usuario con las descripciones de las tareas y los objetivos de negocio. El equipo debe también probar la usabilidad del diseño.</p>	[56]
<b>EXPERIMENTOS PILOTO</b>	<p>En muchos casos el costo de un nuevo sistema puede ser elevado, pero el principal riesgo es si la organización puede adaptarse al sistema y usarlo para mejorar el rendimiento. Los cambios organizacionales mismos a menudo son más costosos que el producto. En esta situación muchos de los riesgos pueden ser eliminados con un experimento piloto. Una pequeña parte de la organización prueba el nuevo sistema sobre unas bases de prueba, pero con datos de producción reales. Al mismo tiempo, experimentan con procedimientos de trabajo nuevos. El equipo de proyecto observa los resultados y evalúa los costos y beneficios del nuevo sistema. Esto también ayuda a definir los requerimientos finales y sus prioridades. El problema principal puede ser que el sistema piloto puede ser usado por un periodo de tiempo extenso.</p>	[56]
<b>PUNTOS DE VISTA</b>	<p>Los requerimientos pueden venir de los stakeholders del sistema, del dominio de aplicación y de otros sistemas que interactúan con el sistema a especificar. Estas fuentes de requerimientos se pueden representar como puntos de vista del sistema, donde cada uno representa un subconjunto de requerimientos para el sistema. Cada punto de vista proporciona una perspectiva nueva en el sistema, pero que no son completamente independientes.</p>	[3]
<b>ENTREVISTAS</b>	<p>Esta técnica se emplea para reunir información proveniente de personas o de grupos. Durante la entrevista, el analista conversa con el encuestado. Por lo común los encuestados son usuarios de los sistemas existentes o usuarios en potencia del sistema propuesto. El éxito de esta técnica depende de la habilidad del entrevistador y de su preparación para la misma.</p>	[40]
<b>CUESTIONARIOS</b>	<p>Consiste en redactar un documento con preguntas cuyas respuestas sean cortas y concretas, o incluso cerradas por unas cuantas opciones en el propio cuestionario. Este cuestionario será cumplimentado por el grupo de personas entrevistadas, o simplemente para recoger información en forma independiente de una entrevista.</p>	[26]
<b>SISTEMAS EXISTENTES</b>	<p>Esta técnica consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el sistema a ser construido. Por un lado podemos analizar las interfaces de usuarios, observando el tipo de información que se maneja y como es manejada, por otro lado también es útil analizar las distintas salidas que los sistemas producen (listados, consultas, etc), porque siempre pueden surgir nuevas ideas sobre la base de estas.</p>	[40]

**Tabla 3.3.** Técnicas útiles para la IR (continuación)

TÉRMINO	SIGNIFICADO	FUENTE
<b>ESCENARIOS</b>	Esta técnica consiste en describir las características del sistema a desarrollar mediante una secuencia de pasos. La representación puede variar dependiendo del autor. Esta representación puede ser casi textual o ir encaminada hacia representaciones gráficas en forma de diagramas de flujo. El análisis de escenarios puede ofrecer información importante sobre las necesidades funcionales del sistema.	[26]
<b>ETNOGRAFÍA</b>	Es una metodología de investigación de los métodos cualitativos de la antropología cultural y el estudio de las razas y pueblos de una región o sociedad particular. Consiste en la recolección de datos en el terreno y teniendo como informantes a los integrantes de una comunidad dada. Los datos recopilados consisten en la descripción densa y detallada de sus costumbres, creencias, mitos, genealogías, historia, etc. Dicha información se obtiene, sobre todo, por medio de largas entrevistas con miembros de la comunidad o informantes claves de ellas.  Según Sommerville, es una técnica de observación que se puede utilizar para entender los requerimientos sociales y organizacionales. Un analista se sumerge por sí solo en el entorno laboral donde se utilizará el sistema. Observa el trabajo diario y anota las tareas reales en las que los participantes están involucrados.	[3]  [57]
<b>JOINT APPLICATION DEVELOPMENT/ DESARROLLO CONJUNTO DE APLICACIONES (JAD)</b>	Esta técnica resulta una alternativa a las entrevistas. Es una práctica de grupo que se desarrolla durante varios días y en la que participan analistas, usuarios, administradores del sistema y clientes. El equipo de trabajo se reúne en varias sesiones. En cada una de ellas se establecen los requisitos de alto nivel a trabajar, el ámbito del problema y la documentación. Durante la sesión se discute en grupo sobre estos temas llegándose a una serie de conclusiones que se documentan. En cada sesión se van concretando más las necesidades del sistema.	[26]
<b>TORMENTA DE IDEAS (BRAINSTORMING)</b>	Es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El número de personas que participa no debe ser muy numeroso, una de ellas debe asumir el rol de moderador. Como técnica de adquisición de requisitos es fácil de usar y de aplicar. Ofrece una visión general de las necesidades del sistema, pero no sirve para ofrecer detalles concretos del mismo, por lo que suele aplicarse en los primeros encuentros.	[26]
<b>MAPAS CONCEPTUALES</b>	Son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos se desarrollan con el usuario y sirven para aclarar los conceptos relacionados con el sistema a desarrollar. Son fáciles de entender por el usuario, más aun si están desarrollados en el lenguaje de éste. Pueden llegar a ser muy ambiguos en casos complejos si no se acompaña de una descripción textual.	[26]
<b>COMPARACIÓN DE TERMINOLOGÍA</b>	Esta técnica se utiliza en forma complementaria a otras para obtener consenso respecto de la terminología a usar en el proyecto de desarrollo. Para ello, es necesario identificar el uso de términos diferentes para los mismos conceptos, misma terminología para diferentes conceptos o cuando no hay concordancia exacta ni en el vocabulario ni en los conceptos.	[26]

**Tabla 3.3.** Técnicas útiles para la IR (continuación)

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>PROCESO DE ANÁLISIS JERÁRQUICO (AHP)</b>	Esta técnica tiene por objetivo resolver problemas cuantitativos, para facilitar el pensamiento analítico y las métricas. Consiste en una serie de pasos a saber: encontrar los requerimientos que van a ser priorizados; combinar los requerimientos en las filas y columnas de la matriz de nxn de AHP; hacer algunas comparaciones de los requerimientos en la matriz; sumar las columnas; normalizar la suma de las filas y calcular los promedios. Para un volumen grande de requerimientos, esta técnica no resulta la más adecuada.	[58]
<b>LENGUAJE NATURAL</b>	Esta técnica consiste en definir los requisitos en lenguaje natural sin usar reglas para ello. Pero, a pesar de que son muchos los trabajos que critican su uso, es cierto que a nivel práctico se sigue utilizando. Resulta ser una técnica muy ambigua para la definición de requisitos	[26]
<b>GLOSARIO Y ONTOLOGÍAS</b>	Su uso proviene de la necesidad de establecer un marco de terminología común. En un glosario de términos se recogen y definen los conceptos más relevantes y críticos para el sistema. En las ontologías no solo se aparecen los términos, sino también las relaciones entre ellos.	[26]
<b>PLANTILLAS O PATRONES</b>	Esta técnica tiene por objetivo el describir los requisitos mediante el lenguaje natural pero de una forma estructurada. Una plantilla es una tabla con una serie de campos y una estructura predefinida que el equipo de desarrollo va cumplimentando usando para ello el lenguaje del usuario. Las plantillas eliminan parte de la ambigüedad del lenguaje natural al estructurar la información. Sin embargo, si el nivel de detalle elegido es demasiado detallado, el trabajo de rellenar plantillas y mantenerlas puede ser tedioso.	[26]
<b>LENGUAJES FORMALES</b>	Esta técnica se usa para describir los requisitos de un sistema. Resultan ser complejas en su utilización y para ser entendidas por el cliente. El mayor inconveniente es que no favorecen la comunicación entre cliente y analista. Por el contrario es la representación menos ambigua de los requisitos y la que más se presta a técnicas de verificación automatizadas.	[26]
<b>LENGUAJE NATURAL ESTRUCTURADO</b>	Este enfoque depende de la definición de formularios o plantillas estándares para expresar la especificación de requerimientos	[3]
<b>LENGUAJES DE DESCRIPCIÓN DE DISEÑO</b>	Este enfoque utiliza un enfoque similar a uno de programación, pero con características más abstractas, para especificar los requerimientos por medio de la definición de un modelo operativo del sistema. Este enfoque no se utiliza ampliamente en la actualidad, aunque puede ser útil para especificaciones de interfaces.	[3]
<b>NOTACIONES GRÁFICAS</b>	Para definir los requerimientos funcionales del sistema, se utiliza un lenguaje gráfico, complementado con notaciones de texto.	[3]
<b>REVIEWS (REVISIONES)</b>	Esta técnica consiste en la lectura y corrección de la documentación completa o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar la consistencia de la documentación o información faltante.	[26]
<b>AUDITORÍAS</b>	La revisión de la documentación con esta técnica consiste en chequeo de los resultados contra una checklist predefinida o definida a comienzos del proceso, es decir que solo una muestra se revisa.	[26]

**Tabla 3.3.** Técnicas útiles para la IR (continuación)

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>MATRIZ DE TRAZABILIDAD</b>	Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.	[26]
<b>PROTOTIPOS</b>	Se basa en obtener un prototipo, a partir de la definición de requisitos, sin tener la totalidad de la funcionalidad del sistema, permiten al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario. Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.	[26]
<b>DICCIONARIO DE DATOS (DD)</b>	Es una herramienta de modelado importante, sin él el usuario no podrá estar seguro de que entendió los detalles de la aplicación. Es un listado organizado de todos los datos pertinentes al sistema, con definiciones precisas y rigurosas para que tanto el usuario como el analista tengan un entendimiento común de todas las entradas, salidas, componentes de almacenes y cálculos intermedios.	[48]
<b>DIAGRAMA ENTIDAD RELACIÓN (DER)</b>	Es una notación gráfica para modelar datos. El DER es un modelo de red que describe con un alto nivel de abstracción la distribución de datos almacenados en un sistema. La importancia de modelar los datos de un sistema podría ser primeramente, porque las estructuras de datos y las relaciones pueden ser tan complejas que se deseará enfatizarlas y examinarlas independientemente del proceso que se llevará a cabo.	[48]
<b>DIAGRAMA DE TRANSICIÓN DE ESTADOS (DTE)</b>	Es una herramienta de modelado que enfatiza el comportamiento dependiente del tiempo del sistema. El comportamiento dependiente del tiempo es un aspecto importante de muchos sistemas. Para algunos sistemas computacionales de empresas este aspecto no es importante, puesto que la secuencia es esencialmente trivial. Así en muchos sistemas computacionales la función N no puede llevar a cabo su labor hasta que reciba la entrada que requiere y esta entrada se produce como salida de una función N-1, y así sucesivamente.	[48]
<b>DIAGRAMA DE FLUJO DE DATOS (DFD)</b>	Es una herramienta gráfica de modelado permite visualizar un sistema como una red de procesos funcionales, conectados entre sí por conductos y tanques de almacenamiento de datos. Esta herramienta es más comúnmente usada por sistemas operacionales en los cuales las funciones del sistema son de gran importancia y son más complejas que los datos que éste maneja.	[48]
<b>DIAGRAMA DE CONTEXTO</b>	Es un caso especial del DFD, en donde una burbuja representa la totalidad del sistema. El diagrama de contexto enfatiza varias características importantes del sistema: las personas, organizaciones y sistemas, (terminadores), con los que se comunica el sistema; los datos que el sistema recibe del mundo exterior y que deben procesarse de alguna forma; los datos que el sistema produce y que se envían al mundo exterior; los almacenes de datos que el sistema comparte con los terminadores y la frontera entre el sistema y el resto del mundo.	[48]
<b>DIAGRAMA DE CASOS DE USO</b>	Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. Muestran un conjunto de casos de uso y de actores y sus relaciones.	[49]

**Tabla 3.3.** Técnicas útiles para la IR (continuación)

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>DIAGRAMA DE INTERACCIÓN</b>	Consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que pueden ser enviados entre ellos; los diagramas de interacción tratan la vista dinámica de un sistema; un término genérico que se aplica a varios tipos de diagramas que enfatizan las interacciones de objetos, incluyendo diagramas de colaboración, diagramas de secuencia y diagramas de actividad.	[49]
<b>DIAGRAMA DE CLASES</b>	Muestra un conjunto de clases, interfaces y colaboraciones y las relaciones entre éstos; los diagramas de clases muestran el diseño de un sistema desde un punto de vista estático; es un diagrama que muestra una colección de elementos (estáticos) declarativos.	[49]
<b>DIAGRAMA DE COLABORACIÓN</b>	Es un diagrama de interacción que enfatiza la organización estructural de los objetos que envían y reciben mensajes; es un diagrama que muestra las interacciones organizadas alrededor de instancias y de los enlaces entre ellas.	[49]
<b>DIAGRAMA DE COMPONENTES</b>	Muestra un conjunto de componentes y sus relaciones; los diagramas de componentes muestran los componentes de un sistema desde un punto de vista estático.	[49]
<b>DIAGRAMA DE DESPLIEGUE</b>	Muestra un conjunto de nodos y sus relaciones; un diagrama de despliegue muestra el despliegue de un sistema desde un punto de vista estático.	[49]
<b>DIAGRAMA DE OBJETOS</b>	Muestra un conjunto de objetos y sus relaciones en un momento determinado; los diagramas de objetos muestran el diseño o los procesos de un sistema desde un punto de vista estático.	[49]
<b>DIAGRAMA DE SECUENCIA</b>	Es un diagrama de interacción que hace énfasis en la ordenación temporal de los mensajes.	[49]
<b>DIAGRAMA DE NASSI_SHNEIDERMAN</b>	Estos diagramas se introdujeron como una técnica estructurada de creación de diagramas de flujo. Son más organizados, más estructurados y más comprensibles que un diagrama de flujo típico, por ello, a veces se los prefiere como herramienta para crear especificaciones de proceso.	[48]
<b>DIAGRAMA HIPO</b>	Se usan para presentar una visión de alto nivel de las funciones que realiza el sistema, al igual que descomposición de funciones en subfunciones, etc. Estos pueden ser herramientas de modelado útiles porque se parecen al diagrama de organización ya familiar que describe la jerarquía de gerentes, subgerentes, etc. Sin embargo, no muestra los datos utilizados o producidos por el sistema.	[48]
<b>DIAGRAMA DE FLUJO</b>	Muestra una lógica secuencial y de tipo procedimiento. No se aconseja su uso, sin embargo, se lo usa solo para describir lógica detallada y si el analista de sistemas se limita a los símbolos de elaboración de diagramas de flujo equivalentes a las construcciones del español estructurado, entonces no tiene nada de malo su uso.	[48]
<b>ÁRBOLES DE DECISIÓN</b>	Es una herramienta gráfica que permite expresar la lógica de un proceso con claridad y sin ambigüedades.	[50]
<b>TABLAS DE DECISIÓN</b>	Es una herramienta para especificar procesos. Son útiles sobre todo cuando un proceso debe producir alguna salida o tomar alguna acción basada en decisiones complejas. Si las decisiones se basan en diversas variables distintas, y si dichas variables pueden tomar diversos valores, entonces se prefiere una tabla de decisiones como herramienta de especificación.	[48]

**Tabla 3.3.** Técnicas útiles para la IR (continuación)

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>PRE / POST CONDICIONES</b>	<p>Son una manera conveniente de describir la función que debe realizar el proceso, sin decir mucho a cerca del algoritmo o procedimiento que se utilizará. Las precondiciones describen todas las cosas que deben darse antes de que el proceso pueda comenzar a ejecutarse. De manera similar las postcondiciones describen lo que debe darse cuando el proceso ha concluido.</p> <p>Aunque este enfoque sea bastante útil y tenga un gran número de ventajas, hay ocasiones en las cuales puede no ser apropiado. La falta de pasos intermedios entre entradas y salidas es deliberada y consciente, pero puede volverse difícil de entender si el lector no visualiza algún tipo de procedimiento que lleve de las entradas a las salidas. Además, si existen relaciones complejas entre entradas y salidas, podría ser más fácil escribir una especificación utilizando lenguaje estructurado.</p>	[48]

**Tabla 3.4.** Tipos de Requerimientos a identificar en el proceso de IR

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>REQUERIMIENTO</b>	Un requerimiento es un atributo necesario en un sistema, un acuerdo que identifica una capacidad, característica o factor de calidad de un sistema para que el tenga valor y utilidad para un cliente o usuario	[20]
<b>REQUERIMIENTOS DEL USUARIO</b>	Son declaraciones, en lenguaje natural y diagramas, de los servicios que se espera que le sistema proporcione y de las restricciones bajo las cuales debe funcionar.	[3]
<b>REQUERIMIENTOS DEL SISTEMA</b>	Establecen con detalle las funciones, servicios y restricciones operativas del sistema. Se clasifican en requerimientos funcionales, no funcionales o requerimientos del dominio.	[3]
<b>REQUERIMIENTOS FUNCIONALES</b>	Son declaraciones de los servicios que debe proporcionar el sistema de la manera en que este debe reaccionar a entradas particulares y de como se debe comportar en situaciones particulares. En algunos casos, también pueden declarar explícitamente lo que el sistema no debe hacer.	[3]
<b>REQUERIMIENTOS NO FUNCIONALES</b>	Son restricciones de los servicios o funciones ofrecidos por el sistema. Incluyen restricciones de tiempo, sobre le proceso de desarrollo y estándares. A menudo se aplican al sistema en su totalidad. Normalmente apenas se aplican a características o servicios individuales del sistema.	[3]
<b>REQUERIMIENTOS DEL DOMINIO</b>	Proviene del dominio de aplicación del sistema y que reflejan las características y restricciones de ese dominio. Pueden ser funcionales o no funcionales.	[3]
<b>REQUERIMIENTOS DEL PRODUCTO</b>	Estos requerimientos especifican el comportamiento del producto. Algunos ejemplos son los requerimientos de rendimiento en la rapidez de ejecución del sistema y cuánta memoria se requiere; los requerimientos de fiabilidad que fijan la tasa de fallos para que el sistema sea aceptable; los requerimientos de portabilidad, y los requerimientos de usabilidad.	[3]

**Tabla 3.4.** Tipos de Requerimientos a identificar en el proceso de IR (continuación)

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>REQUERIMIENTOS ORGANIZACIONALES</b>	Estos requerimientos se derivan de políticas y procedimientos existentes en la organización del cliente y en la del desarrollador. Algunos ejemplos son los estándares en los procesos que deben utilizarse; los requerimientos de implementación, como los lenguajes de programación o el método de diseño a utilizar, y los requerimientos de entrega que especifican cuándo se entregará el producto y su documentación.	[3]
<b>REQUERIMIENTOS EXTERNOS</b>	Este gran apartado incluye todos los requerimientos que se derivan de los factores externos al sistema y de su proceso de desarrollo. Estos pueden incluir los requerimientos de interoperabilidad que definen la manera en que el sistema interactúa con sistemas de otras organizaciones; los requerimientos legislativos que deben seguirse para asegurar que el sistema funcione dentro de la ley, y los requerimientos éticos. Estos últimos son puestos en un sistema para asegurar que será aceptado por sus usuarios y por el público en general.	[3]

**Tabla 3.5.** Herramientas útiles para el proceso de IR

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>INGENIERÍA DEL SOFTWARE ASISTIDA POR COMPUTADORA</b>	Es el nombre que se le da al software que se utiliza para ayudar a las actividades del proceso del software como la Ingeniería de Requerimientos, el diseño, el desarrollo de programas y las pruebas. Por lo tanto las herramientas CASE incluyen editores de diseño, diccionarios de datos, compiladores, depuradores, herramientas de construcción de sistemas, etc.  La tecnología CASE proporciona ayuda al proceso de software automatizando algunas de sus actividades, así como proporcionando información acerca del software en desarrollo.	[3]
<b>HERRAMIENTAS DE PLANIFICACIÓN</b>	Las herramientas que caen dentro de esta categoría se centran en dos áreas principales: el esfuerzo y coste de un proyecto de software; y la planificación del proyecto. Las herramientas de estimación calculan el esfuerzo estimado, la duración del proyecto y el número de gente recomendado.  Las herramientas de planificación de proyectos permiten definir todas las tareas, crear una red de tareas, representar las interdependencias entre tareas y modelizar la cantidad de paralelismo posible dentro del proyecto.	[3] [59]
<b>HERRAMIENTAS DE EDICIÓN</b>	Entre ellas se encuentran: editores de texto, editores de diagramas, procesadores de texto.	[3] [59]
<b>HERRAMIENTAS DE DOCUMENTACIÓN</b>	Las herramientas de producción de documentación y de autoedición se utilizan en casi todos los aspectos de la ingeniería del software. La mayoría de las organizaciones de desarrollo de software emplean mucho tiempo en el desarrollo de documentos y en muchos casos el proceso de documentación es muy ineficiente; por esta razón, estas herramientas constituyen una opción importante para aumentar la productividad.  En la mayoría de los casos las herramientas de documentación suelen estar unidas a otras herramientas CASE.	[3] [59]

**Tabla 3.5.** Herramientas útiles para el proceso de IR (continuación)

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>HERRAMIENTAS DE GESTIÓN DE CAMBIOS</b>	Entre ellas se encuentran las herramientas de rastreo de requerimientos y sistemas de gestión de cambios.	[3] [59]
<b>HERRAMIENTAS DE CONSTRUCCIÓN DE PROTOTIPOS</b>	Las herramientas de creación de prototipos pueden ser herramientas para creación de prototipos en papel, diseñadores de pantalla, herramientas PRO/SIM (de creación de prototipos y de simulación), generadores de código, etc.	[3] [59]
<b>HERRAMIENTAS DE APOYO A MÉTODOS</b>	Entre ellas se encuentran: editores de diseño, diccionario de datos, generadores de código.	[3] [59]

**Tabla 3.6.** Roles del Ingeniero de Requerimientos

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>ADMINISTRADOR</b>	El analista debe manejar el proceso de elicitación de requerimientos y comunicarlo a los stakeholders del sistema. Esta actividad involucra más que una toma de decisiones y priorización de tareas. Los analistas son requeridos frecuentemente para iniciar encuentros con los stakeholders, producir reportes de estado y recordar a los stakeholders sus responsabilidades. En muchos casos el analista es el contacto principal para las preguntas de los stakeholders relacionadas al proyecto, procesos y el sistema.	[60]
<b>ANALISTA</b>	Gran parte de la elicitación involucra analizar no solo el proceso que el sistema debe soportar si no los requerimientos mismos. Los analistas deben traducir e interpretar las necesidades de los stakeholders para hacerlas entendibles para otros stakeholders. Los requerimientos están organizados de acuerdo al significado que cada uno da con respecto al sistema. A menudo el analista es requerido para realizar una introspección durante la elicitación de requerimientos, especialmente cuando los stakeholders no son capaces de expresar sus necesidades claramente o no están familiarizados con la solución.	[60]
<b>FACILITADOR</b>	Cuando la elicitación de requerimientos se realiza mediante entrevistas o sesiones de grupo de trabajo, el analista es requerido para preguntar y registrar respuestas, guiar a los participantes hacia los temas relevantes para obtener información correcta y completa.	[60]
<b>MEDIADOR</b>	Durante la elicitación, los conflictos entre los stakeholders y los requerimientos son inevitables. En muchos casos la priorización de requerimientos por diferentes grupos de stakeholders es fuente de debate y disputa. Cuando estas situaciones ocurren el analista es responsable de encontrar una solución a través de la negociación y el compromiso.	[60]
<b>DESARROLLADOR</b>	El analista es requerido para asumir diferentes roles dentro de la comunidad de desarrolladores durante la elicitación de requerimientos. Esto incluye arquitectos de sistemas, diseñadores, programadores, testadores, aseguradores de la calidad, consultores de implementación y administradores del mantenimiento del sistema.	[60]



**Tabla 3.6.** Roles del Ingeniero de Requerimientos (continuación)

TÉRMINO	SIGNIFICADO	FUENTE
DOCUMENTADOR	A menudo, el analista es responsable de la salida del proceso de elicitación. Comúnmente esto toma la forma de un documento de requerimiento o modelo de sistema detallado. Este rol es importante, puesto que representa el resultado del proceso de elicitación y constituye la base para las etapas subsecuentes del proyecto.	[60]
VALIDADOR	Todos los requerimientos elicitados deben ser validados, verificados y comparados con los objetivos del sistema previamente establecidos. Esto significa que los requerimientos deben describir las necesidades del sistema apropiadamente y que proveerán las funciones necesarias para cubrir los objetivos del sistema.	[60]

**Tabla 3.7.** Términos de competencia del dominio

TÉRMINO	SIGNIFICADO	FUENTE
ESTUDIO DE VIABILIDAD	<p>Todo proceso de ingeniería de requerimientos debería empezar con un estudio de viabilidad. La entrada de este es un conjunto de requerimientos de negocio preliminares, una descripción resumida del sistema y de cómo este pretende contribuir a los procesos del negocio. Los resultados del estudio de viabilidad deberían ser un informe que recomiende si merece o no la pena seguir con la ingeniería de requerimientos y el proceso de desarrollo del sistema.</p> <p>En un estudio de viabilidad, se pueden consultar las fuentes de información, como los jefes de departamentos donde se utilizara el sistema, los ingenieros de software que están familiarizados con el tipo de sistema propuesto, los expertos en tecnología y los usuarios finales del sistema. Una vez que se tiene la información, se redacta el informe del estudio de viabilidad.</p>	[3]
ESTÁNDAR IEEE	IEEE corresponde a las siglas de <i>The Institute of Electrical and Electronics Engineers</i> , el <b>Instituto de Ingenieros Eléctricos y Electrónicos</b> , una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Su trabajo es promover la creatividad, el desarrollo y la integración, compartir y aplicar los avances en las tecnologías de la información, electrónica y ciencias en general para beneficio de la humanidad y de los mismos profesionales.	[61]
NORMAS ISO	<p>Es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica. Su función principal es la de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones a nivel internacional.</p> <p>Es una organización internacional no gubernamental, compuesta por representantes de los organismos de normalización (ON's) nacionales, que produce normas internacionales industriales y comerciales. Dichas normas se conocen como <b>normas ISO</b> y su finalidad es la coordinación de las normas nacionales, en consonancia con el Acta Final de la <a href="#">Organización Mundial del Comercio</a>, con el propósito de facilitar el comercio, facilitar el intercambio de información y contribuir con unos estándares comunes para el desarrollo y transferencia de tecnologías.</p>	[62] [63]

**Tabla 3.7.** Términos de competencia del dominio (continuación)

<b>TÉRMINO</b>	<b>SIGNIFICADO</b>	<b>FUENTE</b>
<b>NORMAS IRAM</b>	Norma técnica del Instituto Argentino de Normalización y Certificación. IRAM es el representante de la Argentina en la International Organization Standardization (ISO), en la Comisión Panamericana de Normas Técnicas (COPANT) y en la Asociación MERCOSUR de normalización (AMN).	[64]
<b>STAKEHOLDER</b>	Es cualquier persona o grupo que se verá afectado por el sistema, directa o indirectamente.	[3]

### III.6.CONCLUSIÓN

En este capítulo se efectuó la especificación del dominio, los objetivos y la especificación de requisitos de la ontología. Además, se realizó la extracción de conocimiento referente al dominio de interés. La finalización de la misma permitió comprender las diferentes etapas involucradas en la IR, las técnicas, las herramientas y demás temas.

Con la información recabada se podrá establecer las reglas y las relaciones pertinentes entre los diferentes términos, útiles para la creación de la ontología OntoIR.

El conocimiento adquirido se refleja en las tablas que componen los glosarios de términos, que servirán de base para los siguientes capítulos.

*Capítulo IV:*  
*Conceptualización*  
*de*  
*OntoIR*



## IV.1. INTRODUCCIÓN

Una ontología es un modelo abstracto del dominio o fenómeno del mundo que representa. En Methontology la actividad de conceptualización convierte y organiza la percepción informal del dominio en una especificación semiformal usando un conjunto de representaciones intermedias basadas en notaciones gráficas y tabulares que pueden ser entendidas por expertos del dominio y desarrolladores de la ontología. [29]

La conceptualización de la ontología mediante gráficos y tablas permite el modelado de los diferentes componentes de la ontología tales como: clases, relaciones, funciones, instancias y axiomas.

La reestructuración del conocimiento adquirido en la etapa anterior (Capítulo III: Especificación de Requerimientos,) se ordena de manera que sea de utilidad para la construcción de la ontología propuesta, OntoIR.

Finalmente, el resultado de la conceptualización es el *modelo conceptual* que consta de: glosarios de términos (fueron obtenidos en el Capítulo III, en la tarea de Adquisición de Conocimientos), una taxonomía de conceptos, diagrama de relaciones binarias, tabla clase \_ subclase \_ propiedades, tabla de propiedades en detalle, tabla de reglas, tabla de axiomas, diagrama de clases y esquema conceptual.

## IV.2. TAXONOMÍA DE CONCEPTOS

La taxonomía define la jerarquía entre los conceptos del dominio. Para su construcción se consideran los conceptos definidos en la tarea III.5. Adquisición de Conocimiento. Las siguientes figuras muestran la taxonomía de conceptos a la que se arribó.

En la Figura 4.1 se muestra la clase principal *Etapas* y las subclases que la componen, correspondientes a las subetapas de cada una de las etapas del proceso de IR.

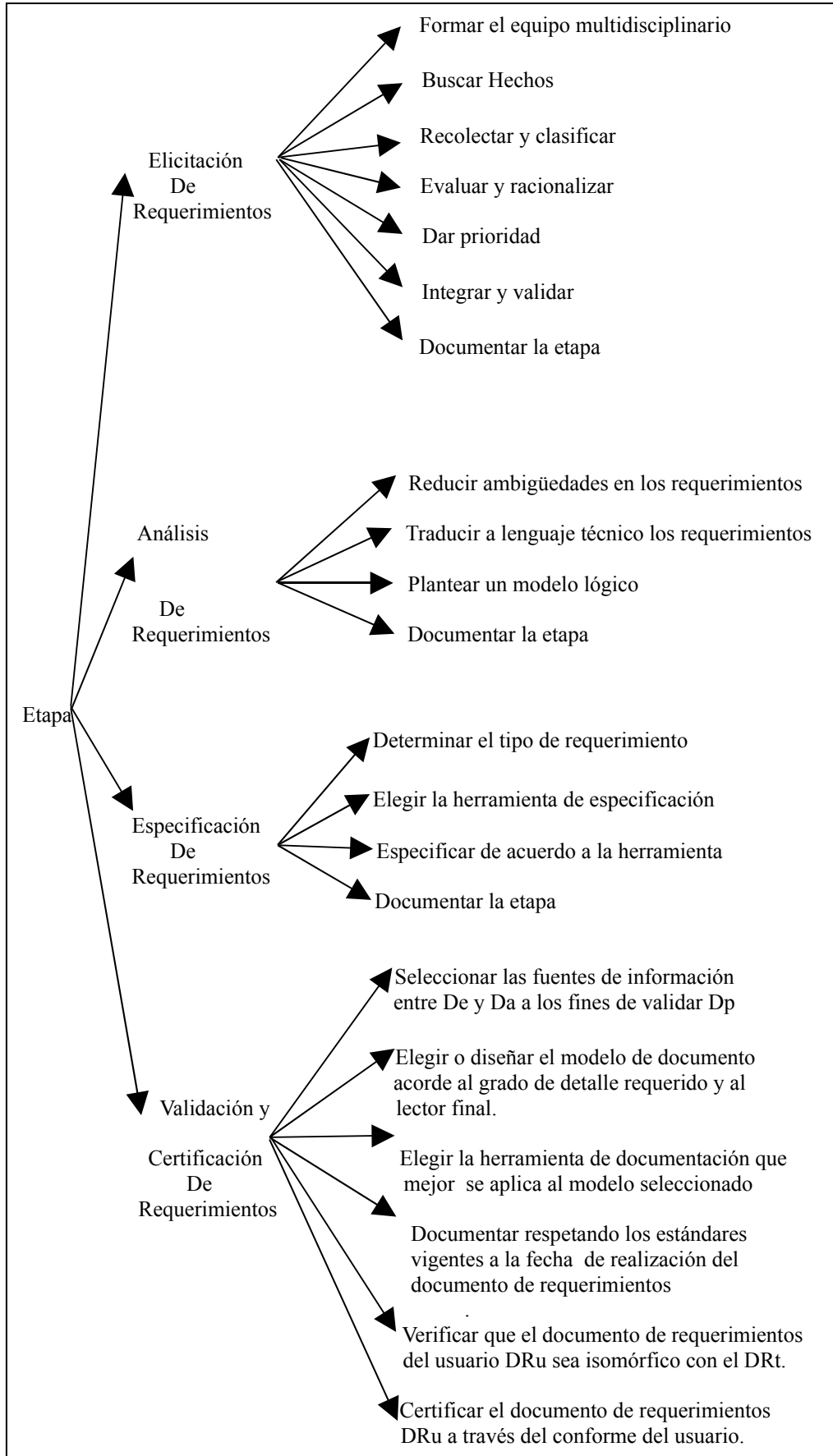
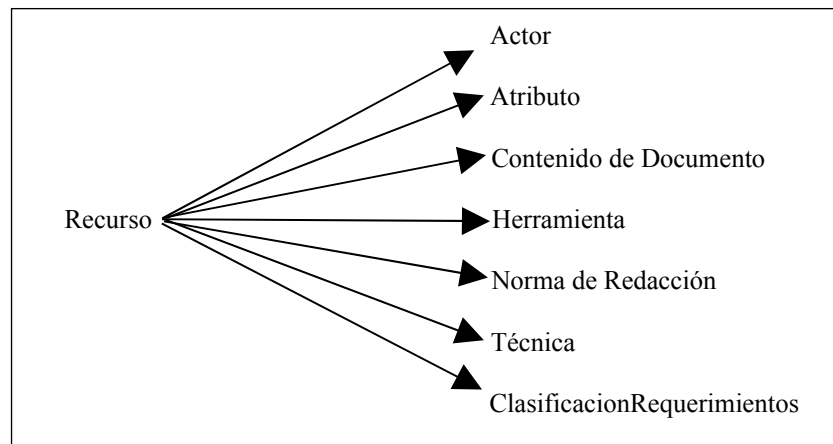


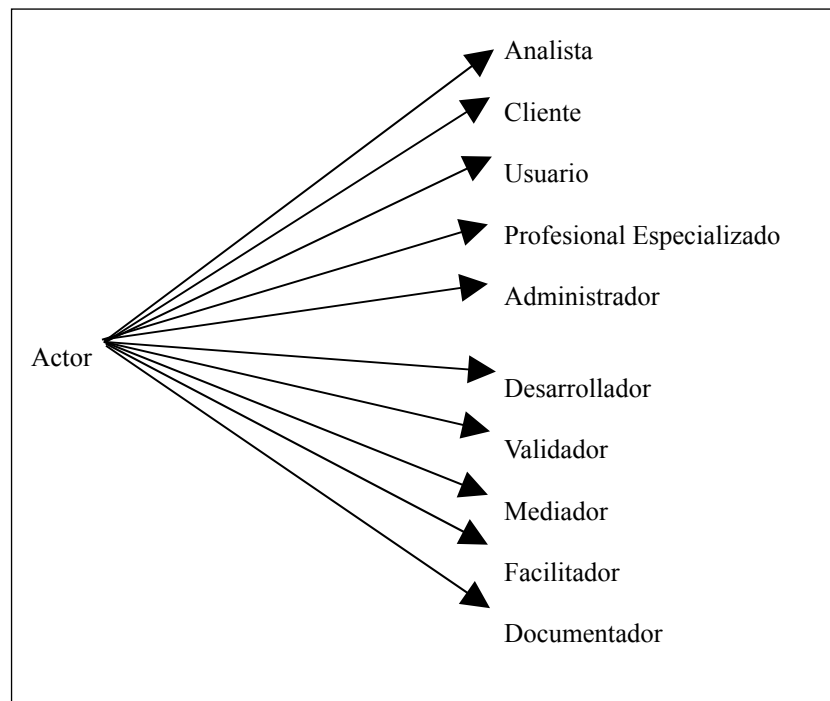
Figura 4.1. Desagregación de la clase Etapa

En la Figura 4.2 se muestra la clase principal *Recurso* y las subclases que la componen, correspondientes a los diferentes recursos que pueden ser de utilidad durante el proceso de IR.



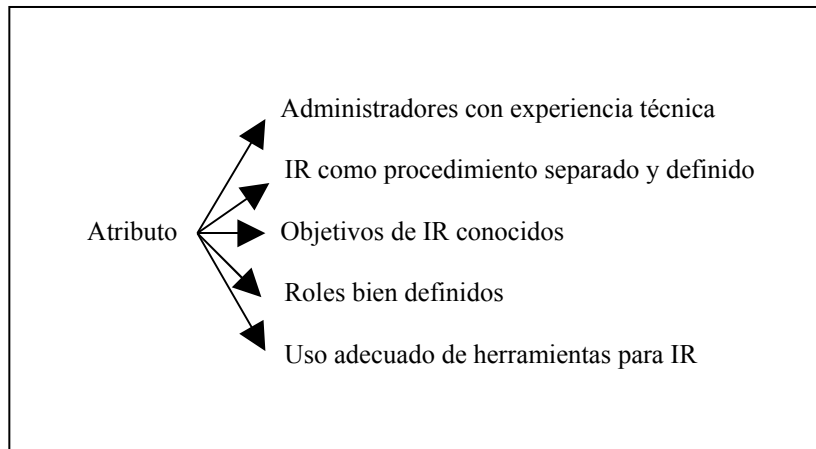
**Figura 4.2.** Desagregación de la clase Recurso

En la Figura 4.3 se muestra la subclase *Actor*, perteneciente a la clase principal *Recurso*, y las subclases que la componen, correspondientes a las diferentes personas que pueden estar involucradas a lo largo del proceso de IR.



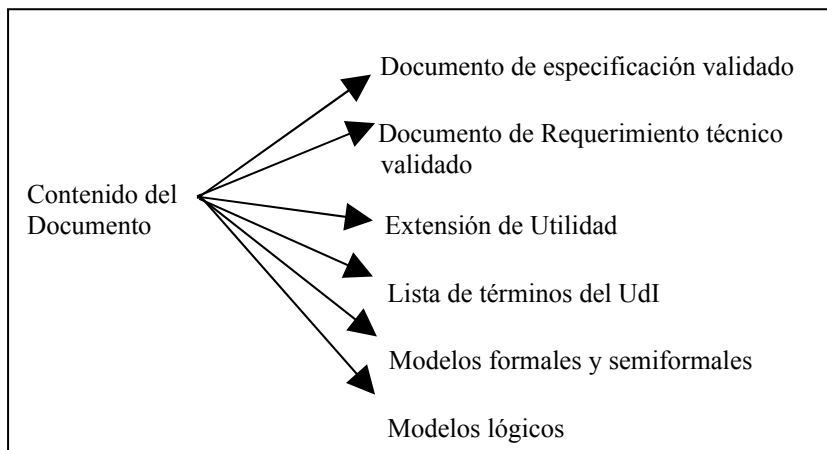
**Figura 4.3.** Desagregación de la clase Actor

En la Figura 4.4 se muestra la subclase *Atributo*, perteneciente a la clase principal *Recurso*, y las subclases que la componen, correspondientes a los diferentes atributos a considerar al momento de efectuar la IR.



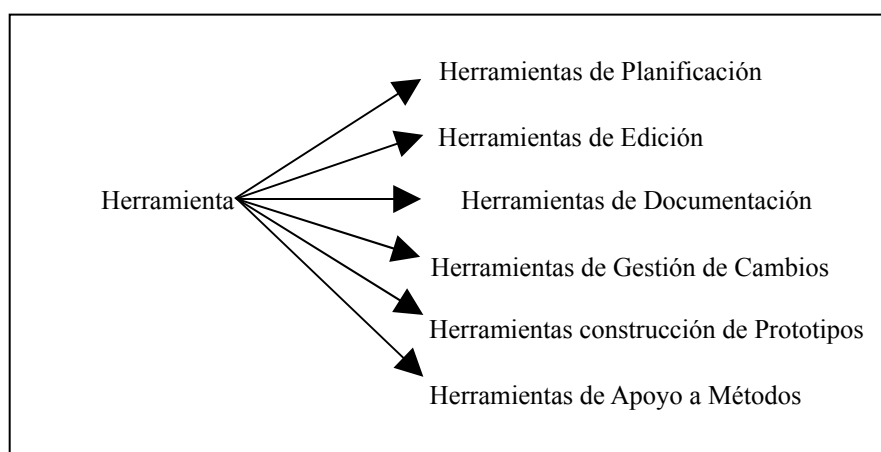
**Figura 4.4.** Desagregación de la clase Atributo

En la Figura 4.5 se muestra la subclase *Contenido del documento*, perteneciente a la clase principal *Recurso*, y las subclases que la componen, correspondientes a los diferentes elementos que constituyen cada uno de los documentos obtenidos a lo largo del proceso de IR.



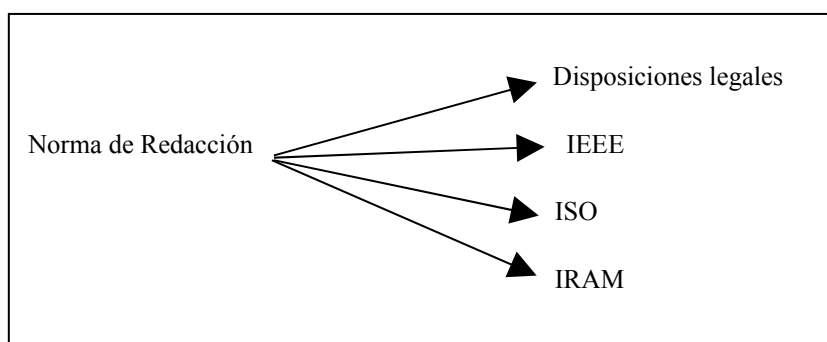
**Figura 4.5.** Desagregación de la clase Contenido del Documento

En la Figura 4.6 se presenta la subclase *Herramienta*, perteneciente a la clase principal *Recurso*, y las subclases que la componen, correspondientes a las diferentes herramientas que pueden ser de utilidad a lo largo del proceso de IR.



**Figura 4.6.** Desagregación de la clase Herramienta

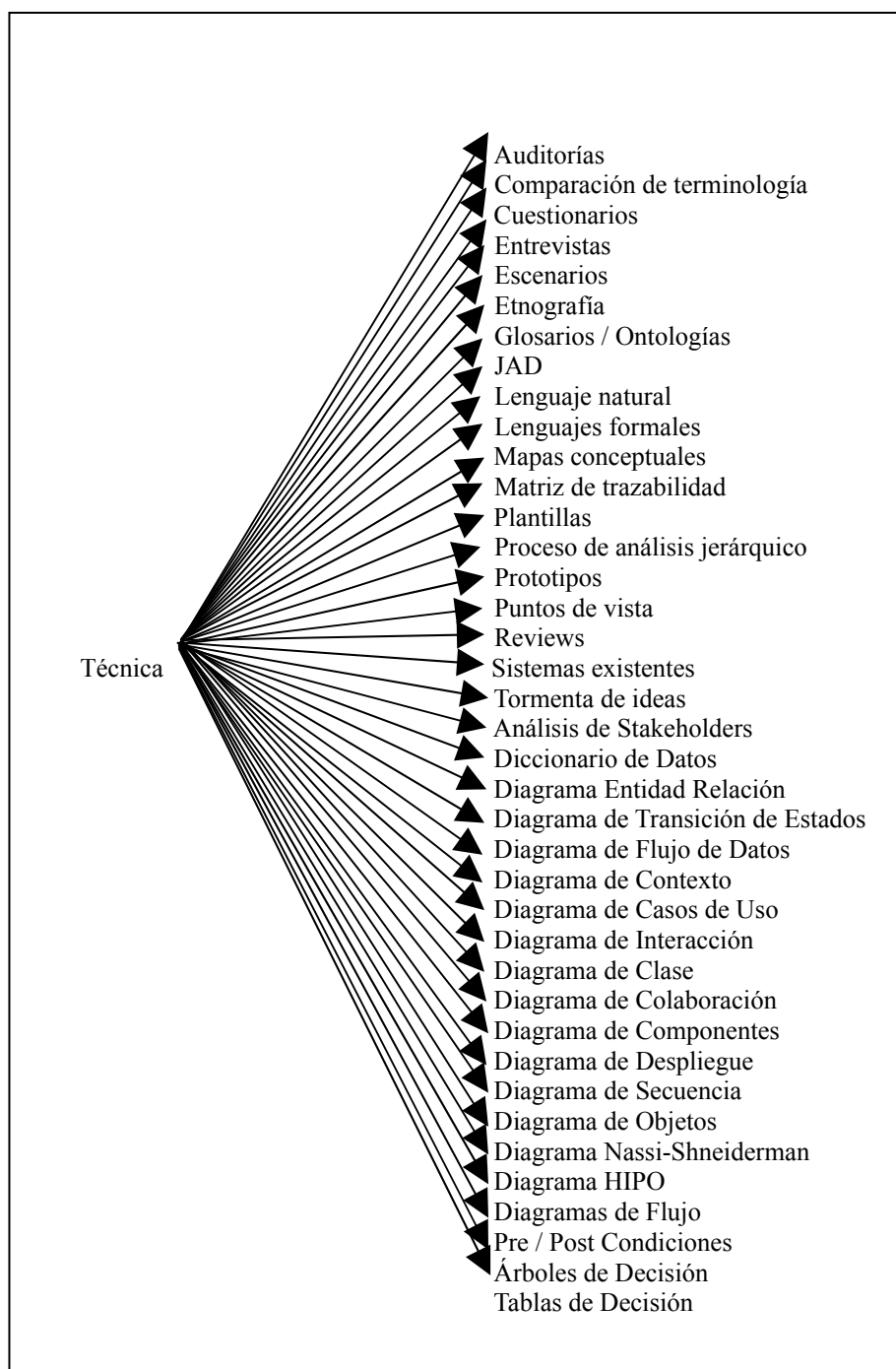
En la Figura 4.7 se muestra la subclase *Norma de Redacción*, perteneciente a la clase principal *Recurso*, y las subclases que la componen, correspondientes a las diferentes normas, estándares o leyes vigentes al momento de confeccionar los diferentes documentos que se obtienen en el proceso de IR.



**Figura 4.7.** Desagregación de la clase Norma de Redacción

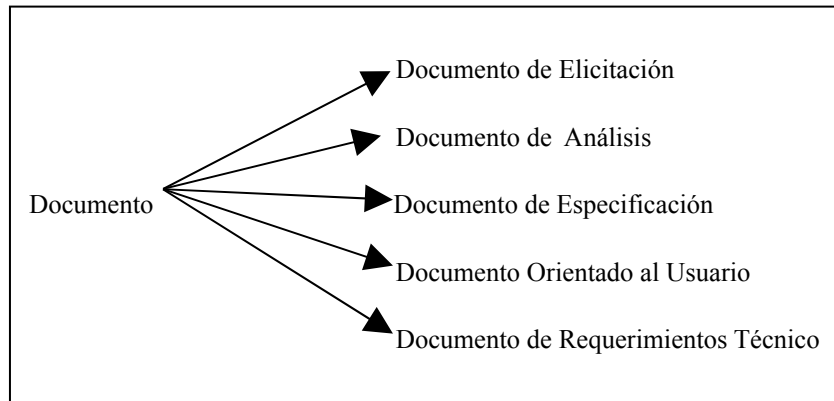
En la Figura 4.8 se muestra la subclase *Técnica*, perteneciente a la clase principal *Recurso*, y las subclases que la componen, correspondientes a las diferentes técnicas que pueden ser aplicadas a lo largo del proceso de IR.





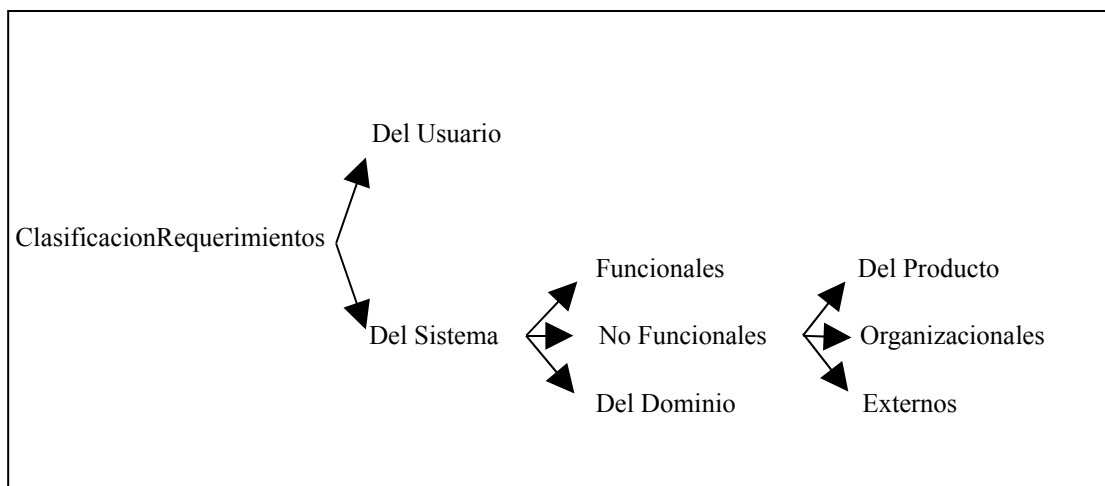
**Figura 4.8.** Desagregación de la clase Técnica

En la Figura 4.9 se muestra la clase *Documento* y las subclases que lo componen, correspondientes a los diferentes documentos que se obtienen a lo largo del proceso de IR.



**Figura 4.9.** Desagregación de la clase Documento

En la Figura 4.10 se muestra la clase *ClasificacionRequerimientos* y las subclases que lo componen, correspondientes a la clasificación de los diferentes tipos de requerimientos propuestos por Sommerville Ian.

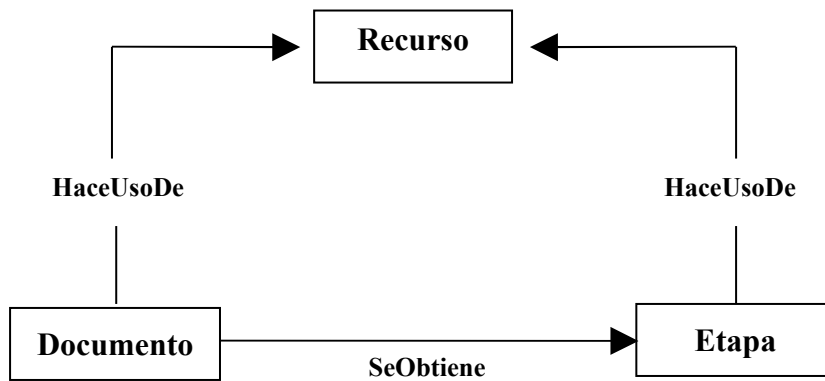


**Figura 4.10.** Desagregación de la clase ClasificacionRequerimientos

### IV.3. DIAGRAMAS DE RELACIONES BINARIAS

Con estos diagramas se establecen los tipos de relaciones entre los conceptos de la taxonomía.

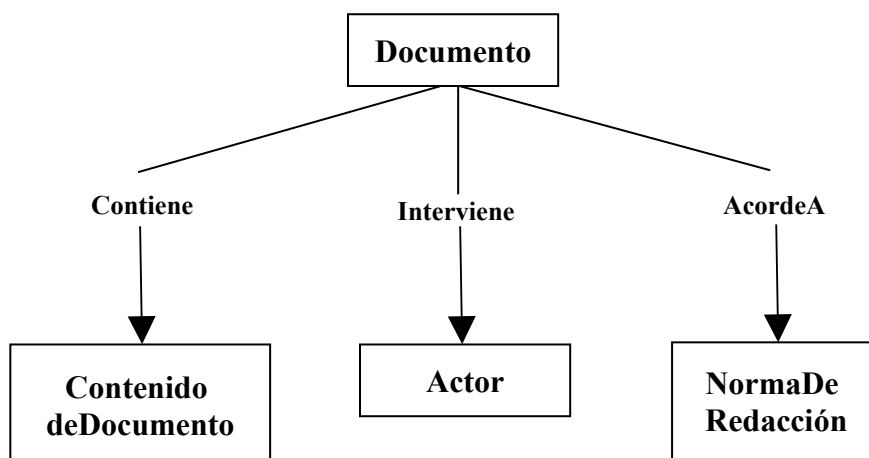
La Figura 4.11 muestra las relaciones binarias existentes entre las clases principales de la ontología. Las clases *Documento* y *Etapa* se relacionan con las subclases de la clase *Recurso* a través de la relación *HaceUsoDe*, y a su vez, la clase *Documento* se relaciona con la clase *Etapa* por medio de la relación *SeObtiene*.



**Figura 4.11.** Diagrama Principal de relaciones binarias

La Figura 4.12 muestra las relaciones binarias existentes entre la clase principal *Documento* y las subclases, de la clase principal *Recurso*, que con ella se relacionan (*Contenido del Documento*, *Actor* y *Norma de Redacción*). La clase *Documento* se relaciona con la clase *ContenidoDeDocumento* a través de la relación *Contiene*; con la clase *Actor* por medio de la relación *Interviene* y con la clase *NormaDeRedaccion* a través de la relación *AcordeA*.

La Figura 4.13 muestra las relaciones binarias existentes entre la clase principal *Etapa* y las subclases, de la clase principal *Recurso*, que con ella se relacionan (*Herramienta ClasificacionRequerimientos*, *Técnica* y *Atributos*). Las subclases de la clase *Etapa* se relacionan con las clases *Herramienta* y *Técnica* a través de la relación *HaceUsoDe*; con la clase *Atributo* por medio de la relación *Considera* y la subclase *DeterminarTipoRequerimiento*, de la clase *Etapa*, se relaciona con *ClasificacionRequerimientos* a través de la relación *SeClasificanEn*.



**Figura 4.12.** Desagregación de relaciones dentro de la clase Documento

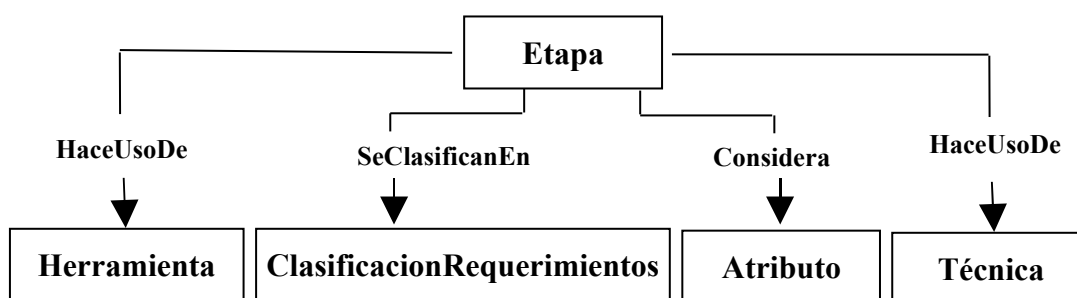


Figura 4.13. Desagregación de relaciones dentro de la clase Etapa

#### IV.4. TABLA CLASE – SUBCLASE – PROPIEDAD

Esta tabla contiene las características de las clases que se utilizaron en el desarrollo de la ontología. Se identifican las clases de más alto nivel; para cada una de ellas se especifica su función, subclases y propiedades que las relacionan.

En el presente trabajo, se han logrado identificar las siguientes clases: *Etapa*, *Recurso* y *Documento*. Para un mayor entendimiento, se detallan las características de las subclases pertenecientes a la clase *Recurso*: *Técnica*, *Herramienta*, *Actor*, *NormadeRedaccion*, *ClasicacionRequerimientos*, *Contenido del Documento* y *Atributo*.

Tabla 4.1. Clase – Subclase – Propiedad de las clases principales

Clases	Descripción	Subclases	Propiedades
<b>Etapa</b>	Contiene información referente a cada una de las etapas del proceso de IR.	<ul style="list-style-type: none"> <li>Elicitación de requerimientos</li> <li>Análisis de requerimientos</li> <li>Especificación de requerimientos</li> <li>Validación y certificación de requerimientos</li> </ul>	PasaA VuelveA SeObtiene HaceUsoDe SeValeDe Considera
<b>Recurso</b>	Contiene información referente a cada uno de los recursos que pueden ser de utilidad para cada etapa, así también como para la confección de los documentos que se obtienen al finalizar cada una de ellas.	<ul style="list-style-type: none"> <li>Técnica</li> <li>Herramienta</li> <li>Actor</li> <li>Normas de Redacción</li> <li>Contenido del Documento</li> <li>Atributo</li> </ul>	HaceUsoDe
<b>Documento</b>	Contiene información referente a los diferentes documentos que se obtienen de ejecutar las diferentes etapas del proceso de IR o de aplicar las diferentes técnicas.	<ul style="list-style-type: none"> <li>Documento de Análisis</li> <li>Documento de Elicitación</li> <li>Documento de Especificación.</li> <li>Documento orientado al usuario</li> <li>Documento de requerimientos técnico</li> </ul>	SeObtiene Interviene AcordeA Contiene

**Tabla 4.2.** Subclases pertenecientes a la clase principal *Recurso*

Subclase	Descripción	Instancias	Propiedad
Técnica	<p>Contiene información referente a las diferentes técnicas que pueden utilizarse para cada una de las etapas del proceso del IR.</p>	<ul style="list-style-type: none"> <li>• Análisis de Stakeholder</li> <li>• Observación</li> <li>• Demostración de tareas</li> <li>• Estudio de documentación</li> <li>• Grupos de foco</li> <li>• Workshops de dominio</li> <li>• Workshop de diseño</li> <li>• Experimentos piloto</li> <li>• Puntos de vista</li> <li>• Entrevistas</li> <li>• Cuestionarios</li> <li>• Sistemas existentes</li> <li>• Escenarios</li> <li>• Etnografía</li> <li>• JAD</li> <li>• Tormenta de ideas</li> <li>• Mapas conceptuales</li> <li>• Comparación de terminología</li> <li>• Proceso de análisis jerárquico</li> <li>• Lenguaje natural</li> <li>• Glosario/Ontologías</li> <li>• Plantillas/Patrones</li> <li>• Lenguajes formales</li> <li>• Lenguaje natural estructurado</li> <li>• Lenguaje de descripción de diseño</li> <li>• Notaciones gráficas</li> <li>• Reviews (revisiones)</li> <li>• Auditorías</li> <li>• Matriz de trazabilidad</li> <li>• Prototipos</li> <li>• Diccionario de datos</li> <li>• Diagrama de entidad relación</li> <li>• Diagrama de transición de estados</li> <li>• Diagrama de flujo de datos</li> <li>• Diagrama de contexto</li> <li>• Diagrama de casos de uso</li> <li>• Diagrama de Interacción</li> <li>• Diagrama de Clases</li> <li>• Diagrama de Colaboración</li> <li>• Diagrama de Componentes</li> <li>• Diagrama de Despliegue</li> <li>• Diagrama de Objetos</li> <li>• Diagrama de Secuencia</li> <li>• Diagrama Nassi_Shneiderman</li> <li>• Diagrama HIPO</li> <li>• Diagrama de Flujo</li> <li>• Pre/Post Condiciones</li> <li>• Árboles de decisión</li> <li>• Tablas de decisión</li> </ul>	HaceUsoDe

**Tabla 4.2.** Subclases pertenecientes a la clase principal *Recurso* (continuación)

Subclase	Descripción	Instancias	Propiedad
<b>Herramienta</b>	Contiene información referente a las diferentes herramientas que pueden ser de utilidad para realizar el proceso de IR.	<ul style="list-style-type: none"> <li>• Herramientas de planificación</li> <li>• Herramientas de edición</li> <li>• Herramientas de documentación</li> <li>• Herramientas de gestión de cambios</li> <li>• Herramientas de construcción de prototipos</li> <li>• Herramientas de apoyo a métodos</li> </ul>	HaceUsoDe
<b>Actor</b>	Contiene información referente a las diferentes personas que intervienen en la confección o validación de los documentos así como a los usuarios finales de los mismos.	<ul style="list-style-type: none"> <li>• Analista</li> <li>• Cliente</li> <li>• Usuario</li> <li>• Profesional especializado</li> <li>• Administrador</li> <li>• Documentador</li> <li>• Desarrollador</li> <li>• Mediador</li> <li>• Facilitador</li> <li>• Validador</li> </ul>	Interviene
<b>Norma de Redacción</b>	Contiene información referente a los diferentes estándares o disposiciones legales vigentes que se deben tener en cuenta en el momento de redactar los diferentes documentos.	<ul style="list-style-type: none"> <li>• Disposiciones Legales</li> <li>• Normas IRAM</li> <li>• Normas ISO</li> <li>• IEEE</li> </ul>	AcordeA
<b>Contenido del Documento</b>	Contiene información referente al contenido de cada documento de acuerdo a la etapa en que se los confecciona.	<ul style="list-style-type: none"> <li>• Documento de especificación validado</li> <li>• Documento de requerimiento técnico validado</li> <li>• Extensión de utilidad</li> <li>• Lista de términos del UDI</li> <li>• Modelos formales/ Semiformales</li> <li>• Modelos lógicos</li> </ul>	Contiene
<b>ClasificaciónRequerimientos</b>	Contiene información referente a la clasificación de tipos requerimientos propuesta	<ul style="list-style-type: none"> <li>• Requerimientos del Usuario</li> <li>• Requerimientos del Sistema</li> <li>• Requerimientos Funcionales</li> <li>• Requerimientos no Funcionales</li> <li>• Requerimientos del Dominio</li> <li>• Requerimientos del Producto</li> <li>• Requerimientos Organizacionales</li> <li>• Requerimientos Externos</li> </ul>	SeClasificaEn
<b>Atributo</b>	Contiene información referente a los atributos que se consideran deben tenerse en cuenta la momento de realizar todo proceso de IR.	<ul style="list-style-type: none"> <li>• Administradores con Experiencia Técnica</li> <li>• IR como procedimiento separado y definido</li> <li>• Objetivos de IR conocidos</li> <li>• Roles bien definidos</li> <li>• Uso adecuado de herramientas</li> </ul>	Considera

#### IV.5. TABLA DE PROPIEDADES EN DETALLE

Para cada relación se especifica el nombre, conceptos fuente y destino, cardinalidad y relación inversa. En la siguiente tabla se muestra la tabla de las relaciones especificadas en la ontología.

**Tabla 4.3.** Propiedades en detalle

Propiedad	Significado	Dominio Clase Destino	Rango Clase Origen	Cardinalidad	Inversa
<b>PasaA</b>	Pasa a la siguiente etapa.	Etapa	Etapa	1:1	VuelveA
<b>SeObtiene</b>	Se usa para especificar que un documento se obtiene de una determinada etapa o de aplicar una determinada técnica.	Etapa	Documento	1:1	Produce
<b>HaceUsoDe</b>	Se usa para especificar que una determinada etapa o que la confección de un documento hace uso de un recurso en particular.	Etapa	Recurso	M:M	EsUsadaPor
<b>AcordeA</b>	Se usa para especificar que un documento se confecciona de acuerdo a un determinado estándar o disposición legal vigente.	Etapa	NormaDeRedaccion	M:M	IndicaEstructura
<b>Contiene</b>	Se usa para especificar el contenido de un determinado documento.	Etapa	ContenidoDel Documento	M:M	FormaParteDe
<b>Interviene</b>	Se usa para indicar las personas que intervienen en la construcción o validación de un determinado documento o los destinatarios del mismo.	Etapa	Actor	M:M	SeInvolucraCon
<b>SeValeDe</b>	Se usa para indicar los documentos en los que se basa para realizar la validación y certificación de requerimientos.	Etapa	Documento	M:1	EsUtilizadoPor
<b>SeClasificaEn</b>	Se usa para indicar la clasificación de los diferentes tipos de requerimientos.	Etapa	ClasificacionRequerimientos	1:1	SeConsideraEn
<b>Considera</b>	Se usa para indicar los atributos a ser considerados durante le proceso de IR.	Etapa	Atributo	M:M	SeAtribuyeA

#### IV.6. TABLA DE REGLAS

Las reglas necesarias en la ontología, se definen en una tabla de reglas. Para cada regla se especifica el nombre, la descripción, la expresión que formalmente la describe y los conceptos a los que hace referencia. Para su especificación se utilizó la forma: *Si <condiciones> entonces <consecuencias o acciones>*. Las reglas que fueron definidas para la ontología, se muestran en la Tabla 4.4. Cabe aclarar, que la definición de las reglas se hizo de acuerdo al orden de las etapas y subetapas definidas por la metodología DorCu para el proceso de IR.

**Tabla 4.4.** Reglas

Reglas	Relaciones causales Suficientes y Necesarias
1	Si se encuentra en la subetapa “ <i>buscar hechos</i> ”, de la etapa de Elicitación de requerimientos, <b>entonces</b> se pueden usar herramientas de edición y planificación, además de las técnicas auditorías, tormenta de ideas, JAD, etnografía, sistemas existentes, cuestionarios, entrevistas o grupos de foco.
2	Si se encuentra en la subetapa “ <i>buscar hechos</i> ”, de la etapa de Elicitación de requerimientos, <b>entonces</b> los actores que pueden intervenir en la misma son: analista, cliente, usuario, administrador y facilitador.
3	Si se encuentra en la subetapas “ <i>dar prioridad</i> ”, de la etapa de Elicitación de requerimientos, <b>entonces</b> se pueden usar herramientas de edición y la técnica Proceso de análisis jerárquico.
4	Si se encuentra en la subetapas “ <i>dar prioridad</i> ”, de la etapa de Elicitación de requerimientos, <b>entonces</b> el actor que interviene en la misma es el analista.
5	Si se encuentra en la subetapas “ <i>Evaluar y racionalizar</i> ”, de la etapa de Elicitación de requerimientos, <b>entonces</b> se pueden usar herramientas de Planificación y Gestión del cambio, además de las técnicas matriz de trazabilidad, plantillas o patrones o reviews.
6	Si se encuentra en la subetapas “ <i>Evaluar y racionalizar</i> ”, de la etapa de Elicitación de requerimientos, <b>entonces</b> los actores que pueden intervenir en la misma son: analista, cliente, usuario, administrador o mediador.
7	Si se encuentra en la subetapa “ <i>Formar equipo multidisciplinario</i> ”, de la etapa de Elicitación de requerimientos, <b>entonces</b> se hace uso de la técnica Análisis de Stakeholder.
8	Si se encuentra en la subetapa “ <i>Formar equipo multidisciplinario</i> ”, de la etapa de Elicitación de requerimientos, <b>entonces</b> los actores que pueden intervenir en la misma son: analista, profesional especializado, administrador o facilitador.



**Tabla 4.4. Reglas (continuación)**

Reglas	Relaciones causales Suficientes y Necesarias
9	Si se encuentra en la subetapa “Integrar y Validar”, de la etapa de Elicitación de requerimientos, <b>entonces</b> se pueden usar herramientas de Edición, de Construcción de prototipos, de Documentación y de Planificación, además de las técnicas de Matriz de trazabilidad, Auditorías, Prototipos o Reviews.
10	Si se encuentra en la subetapa “Integrar y validar”, de la etapa de Elicitación de requerimientos, <b>entonces</b> los actores que pueden intervenir en la misma son: analista, validador, administrador o medidiador.
11	Si se encuentra en la subetapa “Recolectar y clasificar”, de la etapa de Elicitación de requerimientos, <b>entonces</b> se puede hacer uso de herramientas de Edición o de Gestión de cambio.
12	Si se encuentra en la subetapa “Recolectar y clasificar”, de la etapa de Elicitación de requerimientos, <b>entonces</b> los actores que pueden intervenir en la misma son: analista, cliente usuario, administrador o mediador.
13	Si se encuentra en la subetapa “Documentar Elicitación”, de la etapa de Elicitación de requerimientos, <b>entonces</b> los documentos creados se realizan de acuerdo a Disposiciones legales, Estándar IEEE, Normas ISO y normas IRAM.
14	Si se encuentra en la subetapa “Documentar Elicitación”, de la etapa de Elicitación de requerimientos, <b>entonces</b> la documentación obtenida contiene la Lista de términos del Udi más toda extensión que se considere de utilidad.
15	Si se encuentra en la subetapa “Documentar Elicitación”, de la etapa de Elicitación de requerimientos, <b>entonces</b> se hace uso de herramientas de Edición y de Documentación, además de las técnicas Lenguaje natural o Plantillas o Patrones.
16	Si se encuentra en la subetapa “Documentar Elicitación”, de la etapa de Elicitación de requerimientos, <b>entonces</b> los actores que pueden intervenir son: analista o documentador.
17	Si se encuentra en la subetapa “Documentar Elicitación”, de la etapa de Elicitación de requerimientos, <b>entonces</b> se obtiene el Documento de Elicitación.
18	Si se encuentra al final de la subetapa “Documentar Elicitación”, de la etapa de Elicitación de requerimientos, <b>entonces</b> debe pasar a la etapa número dos del proceso de IR: Análisis de Requerimientos. Por otra parte, si existen inconsistencias al finalizar la subetapa “Documentar Elicitación”, entonces se debe iterar la Etapa de Elicitación.
19	Si se encuentra en la subetapas “Construir un modelo lógico”, de la etapa de Análisis de requerimientos, <b>entonces</b> se hace uso de herramientas de Edición o de Apoyo a métodos, además de las técnicas Workshops de diseño, Diagramas de caso de uso, de contexto, de flujo de datos, de entidad relación, de interacción, de secuencia, de colaboración, de despliegue o de componentes.

Tabla 4.4. Reglas (continuación)

Reglas	Relaciones causales Suficientes y Necesarias
20	Si se encuentra en la subetapa “ <i>Construir un modelo lógico</i> ”, de la etapa de Análisis de requerimientos, <b>entonces</b> los actores que pueden intervenir en la misma son el analista o desarrollador.
21	Si se encuentra en la subetapa “ <i>Reducir ambigüedad</i> ”, de la etapa de Análisis de requerimientos, <b>entonces</b> se hace uso de herramientas de Edición o de Apoyo a métodos, además de las técnicas Mapas conceptuales, Comparación de terminología y Glosarios u Ontologías.
22	Si se encuentra en la subetapa “ <i>Reducir ambigüedad</i> ”, de la etapa de Análisis de requerimientos, <b>entonces</b> el actor que interviene en la misma es el analista.
23	Si se encuentra en la subetapa “ <i>Traducir a lenguaje técnico</i> ”, de la etapa de Análisis de requerimientos, <b>entonces</b> se hace uso de herramientas de Edición o de Apoyo a métodos, además de las técnicas Lenguaje natural estructurado, Lenguaje de descripción de diseño, Notaciones gráficas, Diccionario de datos y glosarios u Ontologías.
24	Si se encuentra en la subetapa “ <i>Traducir a lenguaje técnico</i> ”, de la etapa de Análisis de requerimientos, <b>entonces</b> los actores que pueden intervenir en la misma son el analista o el desarrollador.
25	Si se encuentra en la subetapa “ <i>Documentar análisis</i> ”, de la etapa de Análisis de requerimientos, <b>entonces</b> los documentos creados se realizan de acuerdo a Disposiciones legales, Estándar IEEE, Normas ISO y normas IRAM.
26	Si se encuentra en la subetapa “ <i>Documentar análisis</i> ”, de la etapa de Análisis de requerimientos, <b>entonces</b> la documentación obtenida contiene los modelos lógicos creados y toda extensión que se considere de utilidad.
27	Si se encuentra en la subetapa “ <i>Documentar análisis</i> ”, de la etapa de Análisis de requerimientos, <b>entonces</b> se hace uso de herramientas de Edición o Documentación, además de la técnica Lenguaje natural.
28	Si se encuentra en la subetapa “ <i>Documentar análisis</i> ”, de la etapa de Análisis de requerimientos, <b>entonces</b> los actores que intervienen en la misma son el analista o documentador.
29	Si se encuentra en la subetapa “ <i>Documentar análisis</i> ”, de la etapa de Análisis de requerimientos, <b>entonces</b> se obtiene el Documento de análisis.
30	Si se encuentra al final de la subetapa “ <i>Documentar análisis</i> ”, de la etapa de Análisis de requerimientos, <b>entonces</b> se pasa a la etapa tres de dicho proceso: Especificación de requerimientos. Si se detectan áreas no especificadas, requisitos contradictorios y peticiones que aparecen como vagas e irrelevantes al finalizar la etapa número dos del proceso, entonces debe volver a la etapa número Uno del mismo (Elicitación de requerimientos).

**Tabla 4.4. Reglas (continuación)**

Reglas	Relaciones causales Suficientes y Necesarias
31	Si se encuentra en la subetapa “ <i>Determinar tipo de requerimiento</i> ”, de la etapa de Especificación de requerimientos, <b>entonces</b> el actor que interviene en la misma es el analista.
32	Si se encuentra en la subetapa “ <i>Determinar tipo de requerimiento</i> ”, de la etapa de Especificación de requerimientos, <b>entonces</b> la clasificación de los requerimientos se realiza de acuerdo a la clasificación propuesta en ClasificacionRequerimientos.
33	Si se encuentra en la subetapa “ <i>Especificar acorde a herramienta seleccionada</i> ”, de la etapa de Especificación de requerimientos, <b>entonces</b> se usan herramientas de Edición, además de las técnicas Tablas de decisión, Lenguaje natural, Diagramas de flujo de datos, Diagramas de clase, Diccionario de datos, Diagramas de transición de estados, Escenarios, Lenguaje natural estructurado, Pre/Post condiciones, Lenguaje formal, Diagramas de caso de uso, Arboles de decisión, Diagramas de Nassi-Shneiderman, Diagramas HIPO o Diagramas de objetos.
34	Si se encuentra en la subetapa “ <i>Especificar acorde a herramienta seleccionada</i> ”, de la etapa de Especificación de requerimientos, <b>entonces</b> los actores que pueden intervenir en la misma son el analista o el desarrollador.
35	Si se encuentra en la subetapa “ <i>Documentar especificación</i> ”, de la etapa de Especificación de requerimientos, <b>entonces</b> los documentos creados se realizan de acuerdo a Estándar IEEE, Normas ISO y normas IRAM.
36	Si se encuentra en la subetapa “ <i>Documentar especificación</i> ”, de la etapa de Especificación de requerimientos, <b>entonces</b> la documentación obtenida contiene los modelos formales o semiformales creados más toda extensión que se considere de utilidad.
37	Si se encuentra en la subetapa “ <i>Documentar especificación</i> ”, de la etapa de Especificación de requerimientos, <b>entonces</b> se hace uso de Herramientas de Edición o Documentación, además de las técnicas Lenguaje de descripción de diseño, Lenguaje natural, Lenguaje natural estructurado o Notaciones gráficas.
38	Si se encuentra en la subetapa “ <i>Documentar especificación</i> ”, de la etapa de Especificación de requerimientos, <b>entonces</b> los actores que pueden intervenir en la misma son el analista o el documentador.
39	Si se encuentra en la subetapa “ <i>Documentar especificación</i> ”, de la etapa de Especificación de requerimientos, <b>entonces</b> se obtiene el Documento de Especificación.
40	Si se encuentra al final de la subetapa “ <i>Documentar especificación</i> ”, de la etapa de Especificación de requerimientos, <b>entonces</b> debe pasar a la etapa número cuatro de dicho proceso (Validación y Certificación de Requerimientos). Si se presentan dificultades para especificar un requerimiento, durante la etapa de Especificación del proceso de IR, entonces se debe volver una etapa anterior, de dicho proceso, que se crea conveniente (Análisis o Elicitación de Requerimientos).

Tabla 4.4. Reglas (continuación)

Reglas	Relaciones causales Suficientes y Necesarias
41	Si se encuentra en la subetapa “ <i>Seleccionar fuente</i> ”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> para validar el Documento de Especificación se considera el Estándar IEEE.
42	Si se encuentra en la subetapa “ <i>Seleccionar fuente</i> ”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> se usan Herramientas de Edición o Documentación.
43	Si se encuentra en la subetapa “ <i>Seleccionar fuente</i> ”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> los actores que intervienen en la misma son el analista, desarrollador o validador.
44	Si se encuentra en la subetapa “ <i>Seleccionar fuente</i> ”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> se obtiene el Documento de Requerimientos Técnico.
45	Si se encuentra en la subetapa “ <i>Seleccionar fuente</i> ”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> se valida el Documento de Especificación a partir del Documento de Análisis.
46	Si se encuentra en la subetapa “ <i>Seleccionar fuente</i> ”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> se valida el Documento de Especificación a partir del Documento de Elicitación.
47	Si surge algún inconveniente al validar el Documento de Especificación <b>entonces</b> se puede regresar a etapas anteriores del proceso que se crea conveniente (Elicitación, Análisis o especificación de requerimientos).
48	Si se encuentra en la subetapa “ <i>Elegir y diseñar modelo de documento</i> ”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> se tienen en cuenta los modelos definido en el estándar de la IEEE.
49	Si se encuentra en la subetapa “ <i>Elegir y diseñar modelo de documento</i> ”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> los actores que intervienen en la misma son el analista, desarrollador, documentador, administrador, usuario o cliente.
50	Si se encuentra en la subetapa “ <i>Elegir herramienta y documentar</i> ”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> se usan herramientas de Edición o Documentación.
51	Si se encuentra en la subetapa “ <i>Elegir herramienta y documentar</i> ”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> los actores que interviene en la misma son el analista, desarrollador o documentador.
52	Si se encuentra en la subetapa “ <i>Elegir herramienta y documentar</i> ”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> la documentación se realiza de acuerdo a Normas de redacción.

**Tabla 4.4.** Reglas (continuación)

Reglas	Relaciones causales Suficientes y Necesarias
53	Si se encuentra en la subetapa “Elegir herramienta y documentar”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> se obtiene el Documento de requerimientos orientado al usuario.
54	Si se encuentra en la subetapa “Validar”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> la validación se realiza de acuerdo a lo establecido en el estándar de la IEEE.
55	Si se encuentra en la subetapa “Validar”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> los actores que interviene en la misma son el analista, desarrollador o validador.
56	Si se encuentra en la subetapa “Validar”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> se tiene en cuenta el Documento de Requerimientos técnico.
57	Si se encuentra en la subetapas “Validar”, de la etapa de Validación y Certificación de requerimientos, <b>entonces</b> se tiene en cuenta el Documento de requerimientos orientado al usuario.
58	Si se encuentra al inicio del proceso de IR, <b>entonces</b> se pueden considerar los Atributos de Calidad para maximizar la calidad del proceso de IR.

#### IV.7. TABLA DE AXIOMAS

Para obtener información referente al Proceso de IR (por ejemplo; secuencias de etapas, técnicas adecuada para cada una de ellas, herramientas que pueden ser de utilidad, estructuras de documentos, etc.), se realiza la formulación de axiomas.

Los axiomas se estructuran de la siguiente manera [47]:

**Op.Log.** <propiedad> **Cond.Op.Log.** C<sub>1</sub>, C<sub>2</sub>, ... C<sub>n</sub>

Donde:

**Op. Log.** es el operador lógico que limita la relación. Puede ser:

$\forall$  <propiedad> **only** C<sub>1</sub>, C<sub>2</sub>, ..., C<sub>n</sub>

$\exists$  <propiedad> **some** C<sub>1</sub>, C<sub>2</sub>, ..., C<sub>n</sub>

$\leq$  <propiedad> **max** <cardinal> C<sub>1</sub>, C<sub>2</sub>,... C<sub>n</sub>

$\geq$  <propiedad> **min** <cardinal> C<sub>1</sub>, C<sub>2</sub>,... C<sub>n</sub>

$\ni$  <propiedad> **has** C<sub>1</sub>, C<sub>2</sub>, ..., C<sub>n</sub>

$=$  <propiedad> **exactly** <cardinal> C<sub>1</sub>, C<sub>2</sub>,... C<sub>n</sub>

**Propiedad:** es la propiedad que vincula dos clases o subclases.

**Cardinal:** es un valor numérico.

$C_i$  representa las condiciones.

En la siguiente tabla se muestran los axiomas que se crearon a partir del conocimiento adquirido de las diferentes fuentes de información consultadas.

**Tabla 4.5. Axiomas**

N° Axioma			Fuente de Adquisición
1	<b>Formulación</b>	Para la subetapa “Buscar hechos“, de la etapa Elicitación de Requerimientos, existen diversas herramientas y técnicas que pueden ser utilizadas para lograr el objetivo de la misma.	[3] [26]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HEdición <b>or</b> HPlanificación <b>or</b> Auditorias <b>or</b> TormentaDeIdeas <b>or</b> JAD <b>or</b> Etnografía <b>or</b> SistemasExistentes <b>or</b> Cuestionarios <b>or</b> Entrevistas <b>or</b> GruposDeFoco)	[59]
2	<b>Formulación</b>	En la ejecución de la subetapa “Buscar hechos”, de la etapa Elicitación de Requerimientos, pueden intervenir los siguientes actores: Analista, Cliente, Usuario, Administrador o Facilitador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista <b>or</b> Cliente <b>or</b> Usuario <b>or</b> Administrador <b>or</b> Facilitador)	
3	<b>Formulación</b>	Para la subetapa “Dar prioridad”, de la etapa de Elicitación de Requerimientos existen diversas herramientas y técnicas que pueden ser usadas para lograr el objetivo de la misma.	[3] [26] [59]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HEdición <b>or</b> ProcesoDeAnálisisJerarquico)	
4	<b>Formulación</b>	En la ejecución de la subetapa “Dar prioridad”, de la etapa Elicitación de Requerimientos, interviene el Analista.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> Analista	
5	<b>Formulación</b>	Para la subetapa “Evaluar y racionalizar”, de la etapa de Elicitación de Requerimientos existen diversas herramientas y técnicas que pueden ser usadas para lograr el objetivo de la misma.	[3] [26] [59]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HPlanificación <b>or</b> HGestiónDeCambio <b>or</b> MatrizDeTrazabilidad <b>or</b> PlantillasPatrones <b>or</b> ComentariosReviews)	

**Tabla 4.5.** Tabla de Axiomas (continuación)

N° Axioma			Fuente de Adquisición
6	<b>Formulación</b>	En la ejecución de la subetapa “Evaluar y racionalizar”, de la etapa Elicitación de Requerimientos, pueden intervenir los siguientes actores: Analista, Cliente, Usuario, Administrador o Mediador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista <b>or</b> Cliente <b>or</b> Usuario <b>or</b> Administrador <b>or</b> Mediador)	
7	<b>Formulación</b>	Para la subetapa “Formar equipo multidisciplinario”, de la etapa de Elicitación de Requerimientos se hace uso de la técnica Análisis de Stakeholder para lograr el objetivo de la misma.	[3] [56]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> AnalisisStakeholder	
8	<b>Formulación</b>	En la ejecución de la subetapa “Formar equipo multidisciplinario”, de la etapa Elicitación de Requerimientos, pueden intervenir los siguientes actores: Analista, Profesional especializado, Administrador o Facilitador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista <b>or</b> ProfesionalEspecializado <b>or</b> Administrador <b>or</b> Facilitador)	
9	<b>Formulación</b>	Para la subetapa “Integrar y validar”, de la etapa de Elicitación de Requerimientos existen diversas herramientas y técnicas que pueden ser usadas para lograr el objetivo de la misma.	[3] [26] [59]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HEdición <b>or</b> HConstrucciónDePrototipo <b>or</b> HDocumentación <b>or</b> HPlanificación <b>or</b> MatrizDeTrazabilidad <b>or</b> Auditorias <b>or</b> Prototipos <b>or</b> ComentariosReviews)	
10	<b>Formulación</b>	En la ejecución de la subetapa “Integrar y validar”, de la etapa Elicitación de Requerimientos, pueden intervenir los siguientes actores: Analista, Validador, Administrador o Mediador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista <b>or</b> validador <b>or</b> Administrador <b>or</b> Mediador)	
11	<b>Formulación</b>	Para la subetapa “Recolectar y clasificar”, de la etapa de Elicitación de Requerimientos existen diversas herramientas que pueden ser usadas para lograr el objetivo de la misma.	[3] [59]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HEdición <b>or</b> HGestiónDeCambio)	

Tabla 4.5. Tabla de Axiomas (continuación)

N° Axioma			Fuente de Adquisición
12	<b>Formulación</b>	En la ejecución de la subetapa “Recolectar y clasificar”, de la etapa Elicitación de Requerimientos, pueden intervenir los siguientes actores: Analista, Cliente, Usuario, Administrador o Mediador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista <b>or</b> Cliente <b>or</b> Usuario <b>or</b> Administrador <b>or</b> Mediador)	
13	<b>Formulación</b>	En la ejecución de la subetapa “Documentar Elicitación”, de la etapa Elicitación de Requerimientos, se puede tener en cuenta diferente documentación disponible para llevarla a cabo.	[34] [61] [62] [63] [64]
	<b>Axioma</b>	$\exists$ AcordeA <b>some</b> (DisposicionesLegales <b>or</b> Estandar_IEEE <b>or</b> NormasIRAM <b>or</b> NormasISO)	
14	<b>Formulación</b>	En la ejecución de la subetapa “Documentar Elicitación”, de la etapa Elicitación de Requerimientos, el documento resultante contendrá el listado de los términos del universo de información y toda extensión que se considere de utilidad para su claridad.	[34]
	<b>Axioma</b>	$\exists$ Contiene <b>some</b> (ExtensionDeUtilidad <b>or</b> ListaTerminosDeUdI)	
15	<b>Formulación</b>	Para la subetapa “Documentar Elicitación”, de la etapa de Elicitación de Requerimientos existen diversas herramientas y técnicas que pueden ser usadas para lograr el objetivo de la misma.	[3] [26] [59]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HDocumentación <b>or</b> HEdición <b>or</b> LenguajeNatural <b>or</b> PlantillasPatrones)	
16	<b>Formulación</b>	En la ejecución de la subetapa “Documentar Elicitación”, de la etapa Elicitación de Requerimientos, pueden intervenir los siguientes actores: Analista o Documentador.	[60]
	<b>Axioma</b>	Interviene <b>some</b> (Analista <b>or</b> Documentador)	
17	<b>Formulación</b>	Al finalizar la subetapa “Documentar Elicitación”, de la etapa Elicitación de Requerimientos, se obtendrá como resultado el Documento de Elicitación (De).	[34]
	<b>Axioma</b>	$\exists$ SeObtiene <b>some</b> DocumentoDeElicitacion	
18	<b>Formulación</b>	Al finalizar la subetapas “Documentar Elicitación”, de la etapa de Elicitación de Requerimientos, se deberá seguir con la etapa de Análisis de Requerimientos. A los efectos de obtener buenos requerimientos, todos los pasos de la primera etapa deben iterar ante la menor inconsistencia detectada tantas veces como sea necesario para garantizar una correcta depuración del producto final de la etapa de Elicitación.	[34]
	<b>Axioma</b>	U (PasaA <b>some</b> Etapa2_AnalisisDeRequerimientos) <b>or</b> (VuelveA <b>some</b> Etapa1_ElicitacioDeRequerimientos)	



**Tabla 4.5.** Tabla de Axiomas (continuación)

N° Axioma			Fuente de Adquisición
19	<b>Formulación</b>	Para la subetapa “Construir modelo lógico”, de la etapa de Análisis de Requerimientos existen diversas herramientas y técnicas que pueden ser usadas para lograr el objetivo de la misma.	[3]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HEdición <b>or</b> HApoyoAMétodos <b>or</b> DiagDeCasosDeUso <b>or</b> WorkshopsDeDiseño <b>or</b> DiagDeContexto <b>or</b> DiagFlujoDeDatos <b>or</b> DiagEntidadRelacion <b>or</b> DiagDeInteraccion <b>or</b> DiagDeSecuencia <b>or</b> DiagDeColaboracion <b>or</b> DiagDeDespliegue <b>or</b> DiagDeComponentes)	[48] [49] [59]
20	<b>Formulación</b>	En la ejecución de la subetapa “Construir modelo lógico”, de la etapa Análisis de Requerimientos, pueden intervenir los siguientes actores: Analista o Desarrollador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista <b>or</b> Desarrollador)	
21	<b>Formulación</b>	Para la subetapa “Reducir ambigüedad”, de la etapa de Análisis de Requerimientos existen diversas herramientas y técnicas que pueden ser usadas para lograr el objetivo de la misma.	[3] [26]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HEdición <b>or</b> HApoyoAMétodos <b>or</b> MapasConceptuales <b>or</b> ComparacionDeTerminologia <b>or</b> GlosarioYOntologias)	[59]
22	<b>Formulación</b>	En la ejecución de la subetapa “Reducir ambigüedad”, de la etapa Análisis de Requerimientos, interviene el Analista.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> Analista	
23	<b>Formulación</b>	Para la subetapa “Traducir a lenguaje técnico”, de la etapa de Análisis de Requerimientos existen diversas herramientas y técnicas que pueden ser usadas para lograr el objetivo de la misma.	[3] [26]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HEdición <b>or</b> HApoyoAMétodos <b>or</b> LenguajeNaturalEstructurado <b>or</b> LenguajeDescripcionDiseño <b>or</b> NotacionesGraficas <b>or</b> DiccionarioDeDatos <b>or</b> GlosarioYOntologias)	[48] [59]
24	<b>Formulación</b>	En la ejecución de la subetapa “Traducir a lenguaje técnico”, de la etapa Análisis de Requerimientos, pueden intervenir el Analista o Desarrollador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista <b>or</b> Desarrollador)	

Tabla 4.5. Tabla de Axiomas (continuación)

N° Axioma			Fuente de Adquisición
25	<b>Formulación</b>	En la ejecución de la subetapa “Documentar Análisis”, de la etapa Análisis de Requerimientos, se puede tener en cuenta diferente documentación disponible para llevarla a cabo.	[34] [61] [62] [63] [64]
	<b>Axioma</b>	$\exists$ AcordeA <b>some</b> (DisposicionesLegales or Estandar_IEEE or NormasIRAM or NormasISO)	
26	<b>Formulación</b>	En la ejecución de la subetapa “Documentar Análisis”, de la etapa Análisis de Requerimientos, el documento resultante contendrá los modelos lógicos creados y toda extensión que se considere de utilidad para su claridad.	[34]
	<b>Axioma</b>	$\exists$ Contiene <b>some</b> (ModelosLogicos or ExtensionDeUtilidad)	
27	<b>Formulación</b>	Para la subetapa “Documentar Análisis”, de la etapa de Análisis de Requerimientos existen diversas herramientas y técnicas que pueden ser usadas para lograr el objetivo de la misma.	[3] [26] [59]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HEdición or HDocumentación or LenguajeNatural)	
28	<b>Formulación</b>	En la ejecución de la subetapa “Documentar Análisis”, de la etapa Análisis de Requerimientos, pueden intervenir el Analista o Documentador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista or Documentador)	
29	<b>Formulación</b>	Al finalizar la subetapa “Documentar Análisis”, de la etapa Análisis de Requerimientos, se obtendrá como resultado el Documento de Análisis (Da).	[34]
	<b>Axioma</b>	$\exists$ SeObtiene <b>some</b> DocumentoDeAnalisis	
30	<b>Formulación</b>	La segunda etapa del proceso de IR (Análisis de Requerimientos) debe ser seguida por la etapa numero Tres de dicho proceso (Especificación de Requerimientos). En la etapa Dos del proceso se estudian los requerimientos extraídos en la etapa anterior a los efectos de poder detectar, entre otros, la presencia de áreas no especificadas, requisitos contradictorios y peticiones que aparecen como vagas e irrelevantes. El resultado de haber llevado a cabo las tareas que involucran estos términos puede, en mas de una oportunidad hacer que se deba regresar a la primera etapa, a los efectos de eliminar todas las inconsistencias y falencias que se han detectado.	[34]
	<b>Axioma</b>	U (PasaA <b>some</b> Etapa3_EspecificacionDeRequerimientos) or (VuelveA <b>some</b> Etapa1_ElicitacioDeRequerimientos)	

**Tabla 4.5.** Tabla de Axiomas (continuación)

N° Axioma			Fuente de Adquisición
31	<b>Formulación</b>	En la ejecución de la subetapa “Determinar tipo de requerimiento”, de la etapa Especificación de Requerimientos, interviene el Analista.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> Analista	
32	<b>Formulación</b>	Para la ejecución de la subetapa “Determinar tipo de requerimiento”, de la etapa de Especificación de requerimientos se puede tener en cuenta la clasificación de requerimientos detallada en ClasificacionReq.	[3]
	<b>Axioma</b>	$\exists$ SeClasificaEn <b>some</b> ClasificacionReq	
33	<b>Formulación</b>	Para la subetapa “Especificar acorde a herramienta seleccionada”, de la etapa de Especificación de Requerimientos existen diversas herramientas y técnicas que pueden ser usadas para lograr su objetivo.	[3] [26] [48] [49] [50] [59]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HEdición <b>or</b> TablasDeDecision <b>or</b> LenguajeNatural <b>or</b> DiagFlujoDeDatos <b>or</b> DiagDeClases <b>or</b> DiccionarioDeDatos <b>or</b> Escenarios <b>or</b> DiagTransicionDeEstados <b>or</b> LenguajeNaturalEstructurado <b>or</b> PrePostCondiciones <b>or</b> LenguajeFormal <b>or</b> DiagDeCasosDeUso <b>or</b> ArbolesDeDecision <b>or</b> DiagNassi-Shneiderman <b>or</b> DiagHIPO <b>or</b> DiagDeObjetos)	
34	<b>Formulación</b>	En la ejecución de la subetapa “Especificar acorde a herramienta seleccionada”, de la etapa Especificación de Requerimientos, pueden intervenir el Analista o Desarrollador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista <b>or</b> Desarrollador)	
35	<b>Formulación</b>	En la ejecución de la subetapa “Documentar Especificación”, de la etapa Especificación de Requerimientos, se puede tener en cuenta diferente documentación disponible para llevarla a cabo.	[34] [61] [62] [63] [64]
	<b>Axioma</b>	$\exists$ AcordeA <b>some</b> (Estandar_IEEE <b>or</b> NormasIRAM <b>or</b> NormasISO)	
36	<b>Formulación</b>	En la ejecución de la subetapa “Documentar Especificación”, de la etapa Especificación de Requerimientos, el documento resultante contendrá los modelos formales o semiformales creados y toda extensión que se considere de utilidad para su claridad.	[34]
	<b>Axioma</b>	$\exists$ Contiene <b>some</b> (ModelosFormalesSemiformales <b>or</b> ExtensionDeUtilidad)	
37	<b>Formulación</b>	Para la subetapa “Documentar Especificación”, de la etapa de Especificación de Requerimientos existen diversas herramientas y técnicas que pueden ser usadas para lograr el objetivo de la misma.	[3] [26] [59]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HDocumentación <b>or</b> HEdición <b>or</b> LenguajeDescripcionDiseño <b>or</b> LenguajeNatural <b>or</b> LenguajeNaturalEstructurado <b>or</b> NotacionesGraficas)	

**Tabla 4.5.** Tabla de Axiomas (continuación)

N° Axioma			Fuente de Adquisición
38	<b>Formulación</b>	En la ejecución de la subetapa “Documentar Especificación”, de la etapa Especificación de Requerimientos, pueden intervenir el Analista o Documentador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista <b>or</b> Documentador)	
39	<b>Formulación</b>	Al finalizar la subetapa “Documentar Especificación”, de la etapa Especificación de Requerimientos, se obtendrá como resultado el Documento de Especificación (Dp).	[34]
	<b>Axioma</b>	$\exists$ SeObtiene <b>some</b> DocumentoDeEspecificacion	
40	<b>Formulación</b>	La tercera etapa del proceso de IR (Especificación de Requerimientos) debe ser seguida por la etapa numero cuatro de dicho proceso (Validación y Certificación de Requerimientos). Partiendo de lo elaborado en la etapa anterior, (Análisis de Requerimientos), tales como funciones, datos, requerimientos no funcionales, objetivos, restricciones de diseño/implementación o costos, e independientemente de la forma en que se realice, la etapa de Especificación es un proceso de descripción del requerimiento. Si se presentan dificultades para especificar un requerimiento se debe volver a la etapa anterior que se crea conveniente.	[34]
	<b>Axioma</b>	(PasaA <b>some</b> Etapa4_ValidacionCertificacionDeRequerimientos) <b>or</b> (VuelveA <b>some</b> (Etapa2_AnalisisDeRequerimientos <b>or</b> Etapa1_ElicitacioDeRequerimientos))	
41	<b>Formulación</b>	Para la ejecución de la subetapa “Elegir o diseñar el modelo del documento”, de la etapa de Certificación y Validación de requerimientos, se tiene en cuenta los modelos detallados en el estándar de la IEEE.	[34] [61]
	<b>Axioma</b>	$\exists$ AcordeA <b>some</b> Estandar_IEEE	
42	<b>Formulación</b>	En la ejecución de la subetapa “Elegir o diseñar el modelo de documento”, de la etapa de Validación y Certificación de Requerimientos, pueden intervenir el Analista, Desarrollador, Administrador, Documentador, Usuario o Cliente.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista <b>or</b> Desarrollador <b>or</b> Documentador <b>or</b> Administrador <b>or</b> Usuario <b>or</b> Cliente)	
43	<b>Formulación</b>	En la ejecución de la subetapa “Elegir herramienta y documentar”, de la etapa Validación y certificación de requerimientos, se puede tener en cuenta diferente documentación disponible para llevarla a cabo.	[34] [61] [62] [63] [64]
	<b>Axioma</b>	$\exists$ AcordeA <b>some</b> (DisposicionesLegales <b>or</b> Estandar_IEEE <b>or</b> NormasIRAM <b>or</b> NormasISO)	

**Tabla 4.5.** Tabla de Axiomas (continuación)

N° Axioma			Fuente de Adquisición
44	<b>Formulación</b>	Para la subetapa “Elegir herramienta y documentar”, de la etapa de Validación y certificación de Requerimientos existen diversas herramientas que pueden ser usadas para lograr el objetivo de la misma.	[3] [59]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HDocumentación <b>or</b> HEdición)	
45	<b>Formulación</b>	En la ejecución de la subetapa “Elegir herramienta y documentar”, de la etapa Validación y Certificación de Requerimientos, pueden intervenir el Analista, Desarrollador o Documentador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Documentador <b>or</b> Analista <b>or</b> Desarrollador)	
46	<b>Formulación</b>	Al finalizar la subetapa “Elegir herramienta y documentar”, de la etapa Validación y certificación de Requerimientos, se obtendrá como resultado el Documento de Requerimientos Orientado al Usuario (DRu).	[34]
	<b>Axioma</b>	$\exists$ SeObtiene <b>some</b> DocumentoDeReqOrientadoUsuario	
47	<b>Formulación</b>	En la ejecución de la subetapa “Seleccionar fuentes de información”, de la etapa Validación y Certificación de Requerimientos, para realizar la validación del documento de especificación se tomará en cuenta lo establecido, en cuanto a validación, por el estándar IEEE.	[34] [61]
	<b>Axioma</b>	$\exists$ AcordeA <b>some</b> Estandar_IEEE	
48	<b>Formulación</b>	Para la subetapa “Seleccionar fuentes de información”, de la etapa de Validación y certificación de Requerimientos existen diversas herramientas que pueden ser usadas para lograr el objetivo de la misma.	[3] [59]
	<b>Axioma</b>	$\exists$ HaceUsoDe <b>some</b> (HDocumentación <b>or</b> HEdición)	
49	<b>Formulación</b>	En la ejecución de la subetapa “Seleccionar fuentes de información”, de la etapa Validación y Certificación de Requerimientos, pueden intervenir el Analista, Desarrollador o Validador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista <b>or</b> Desarrollador <b>or</b> Validador)	
50	<b>Formulación</b>	Al finalizar la subetapa “Seleccionar fuentes de información”, de la etapa Validación y certificación de Requerimientos, se obtendrá como resultado el Documento de Requerimientos Técnico (DRt).	[34]
	<b>Axioma</b>	$\exists$ SeObtiene <b>some</b> DocumentoDeReqTecnico	

Tabla 4.5. Tabla de Axiomas (continuación)

N° Axioma			Fuente de Adquisición
51	<b>Formulación</b>	En la ejecución de la subetapa “Seleccionar fuentes de información”, de la etapa de Validación y certificación de requerimientos, se propone validar el Documento de Especificación a partir del Documento de Análisis.	[34]
	<b>Axioma</b>	$\exists$ SeValeDe <b>some</b> DocumentoDeAnalisis	
52	<b>Formulación</b>	En la ejecución de la subetapa “Seleccionar fuentes de información”, de la etapa de Validación y certificación de requerimientos, se propone validar el Documento de Especificación a partir del Documento de Elicitación.	[34]
	<b>Axioma</b>	$\exists$ SeValeDe <b>some</b> DocumentoDeElicitacion	
53	<b>Formulación</b>	La etapa Cuatro del Proceso de IR se nutre de las anteriores y realiza la validación e integración final de lo obtenido en cada una de las etapas anteriores dando, como resultado final, el Documento de Requerimientos. El resultado de la misma puede ser la necesidad de retornar a las etapas anteriores iterando entre etapas.	[34]
	<b>Axioma</b>	$\exists$ VuelveA <b>some</b> (Etapa1_ElicitacioDeRequerimientos <b>or</b> Etapa2_AnalisisDeRequerimientos <b>or</b> Etapa3_EspecificacionDeRequerimientos)	
54	<b>Formulación</b>	En la ejecución de la subetapa “Validar”, de la etapa Validación y Certificación de Requerimientos, para realizar la validación de los documentos de Requerimiento Técnico y El Orientado al Usuario, se tomará en cuenta lo establecido, en cuanto a validación, por el estándar IEEE.	[34] [61]
	<b>Axioma</b>	$\exists$ AcordeA <b>some</b> Estandar_IEEE	
55	<b>Formulación</b>	En la ejecución de la subetapa “Validar”, de la etapa Validación y Certificación de Requerimientos, pueden intervenir el Analista, Desarrollador o Validador.	[60]
	<b>Axioma</b>	$\exists$ Interviene <b>some</b> (Analista <b>or</b> Desarrollador <b>or</b> Validador)	
56	<b>Formulación</b>	En la ejecución de la subetapa “Validar”, de la etapa de Validación y certificación de requerimientos, se propone validar la correspondencia entre DRt y DRu	[34]
	<b>Axioma</b>	$\exists$ SeValeDe <b>some</b> DocumentoDeReqOrientadoUsuario	
57	<b>Formulación</b>	En la ejecución de la subetapa “Validar”, de la etapa de Validación y certificación de requerimientos, se propone validar la correspondencia entre DRt y DRu	[34]
	<b>Axioma</b>	$\exists$ SeValeDe <b>some</b> DocumentoDeReqTecnico	

**Tabla 4.5.** Tabla de Axiomas (continuación)

N° Axioma			Fuente de Adquisición
58	Formulación	Al momento de llevar a cabo el procedimiento de IR es necesario considerar atributos de calidad para maximizar la calidad del mismo.	[69] [70]
	Axioma	$\exists$ Considera <b>some</b> Atributo	

#### IV.8. DIAGRAMAS DE CLASES

El *diagrama de Clases* captura la estructura lógica del sistema - las clases y cosas que constituyen el modelo. Es un modelo estático, que describe lo que existe y qué atributos y comportamiento tiene, más que cómo se hace algo. Los diagramas de Clases son los más útiles para ilustrar las relaciones entre las clases e interfaces. [65]

A partir de la Tabla 4.1 de Clase-Subclase-Propiedad se construyeron los diagramas de clases correspondientes a los procesos involucrados en la Ontología OntoIR.

La tarea “T1: Establecer Proceso”, es la que permite obtener información acerca de etapas, Herramientas, Técnicas, Autores, Documentos del Proceso de IR en general.

La tarea “T2: Obtener Etapa”, tiene como entrada la Metodología DorCu, a partir de la cual se obtienen las etapas pertenecientes al Proceso de IR.

La tarea “T3: Obtener Herramienta”, tiene como entrada cada una de las etapas del Proceso de IR para las cuales se determina las herramientas adecuadas a utilizar en cada una de ellas.

La tarea “T4: Obtener Técnica”, tiene como entrada cada una de las etapas del Proceso de IR para las cuales se determina las técnicas adecuadas a utilizar en cada una de ellas.

La tarea “T5: Obtener Actores”, tiene como entrada cada una de las etapas del Proceso de IR para las cuales se determinan los actores que intervienen en la ejecución de cada una de ellas.

La tarea “T6: Obtener Documento”, tiene como entrada cada una de las etapas del Proceso de IR y técnicas involucradas en cada una de ellas, para las cuales se determinan los Documentos obtenidos.

La tarea “T7: Determinar Técnica”, establece las técnicas adecuadas para cada etapa, de las cuales se obtiene información con la tarea T8 Obtener Información de Etapa Requerida.

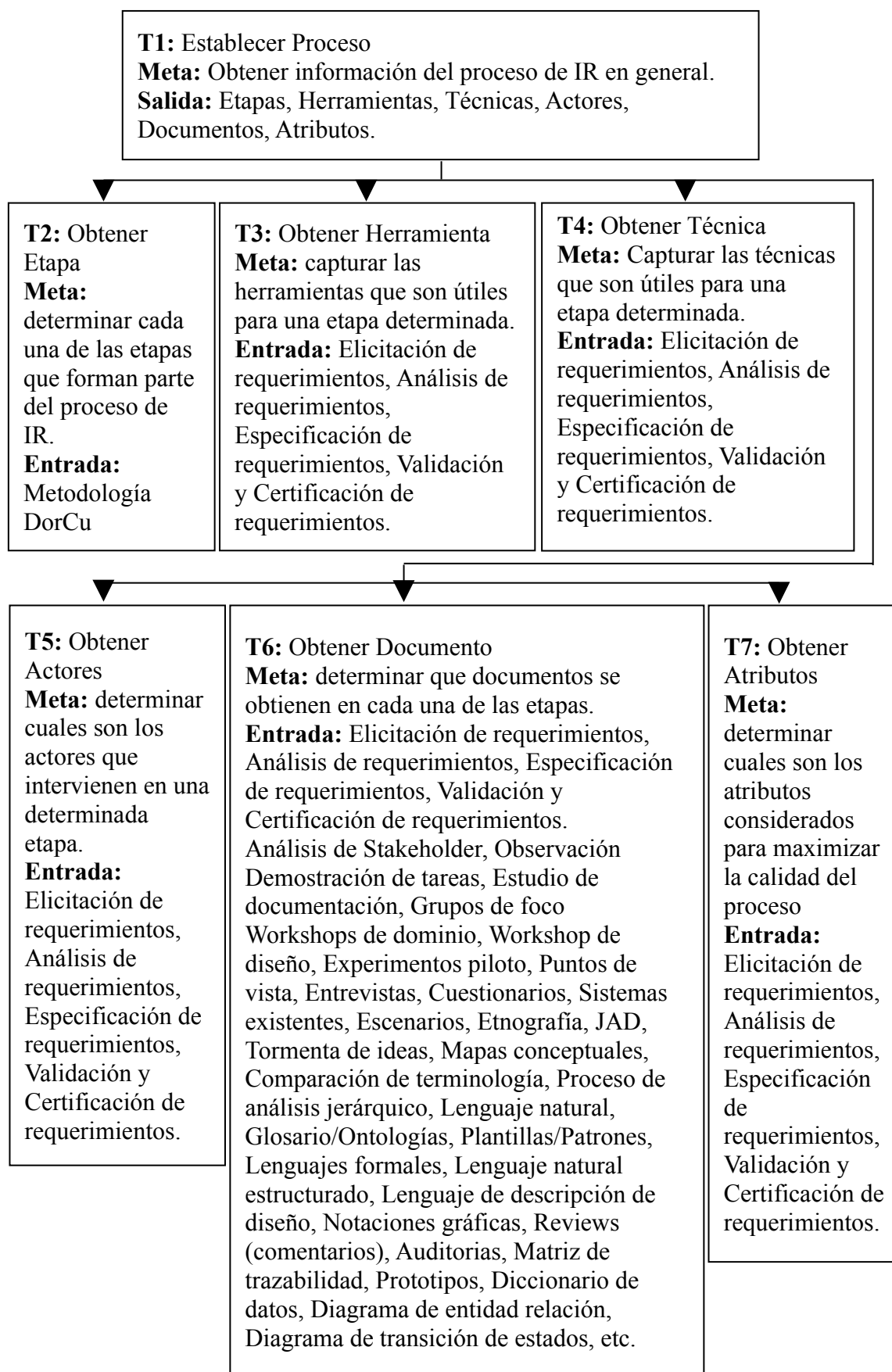
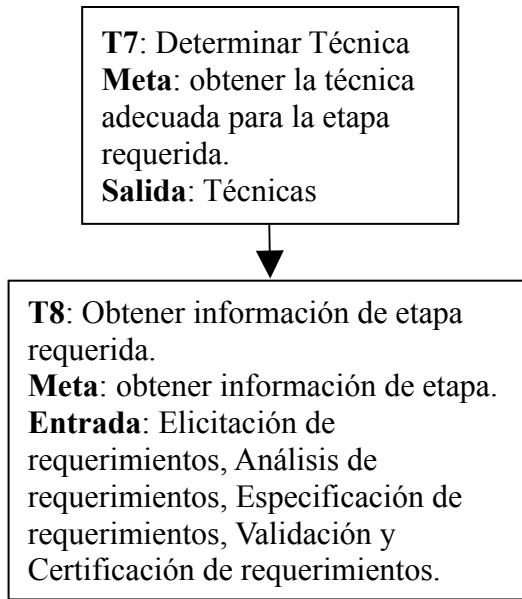


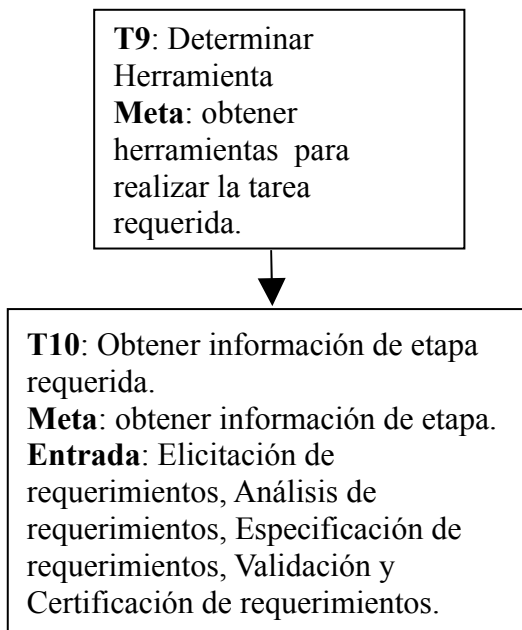
Figura 4.14. Diagrama de clases correspondiente a Establecer Proceso





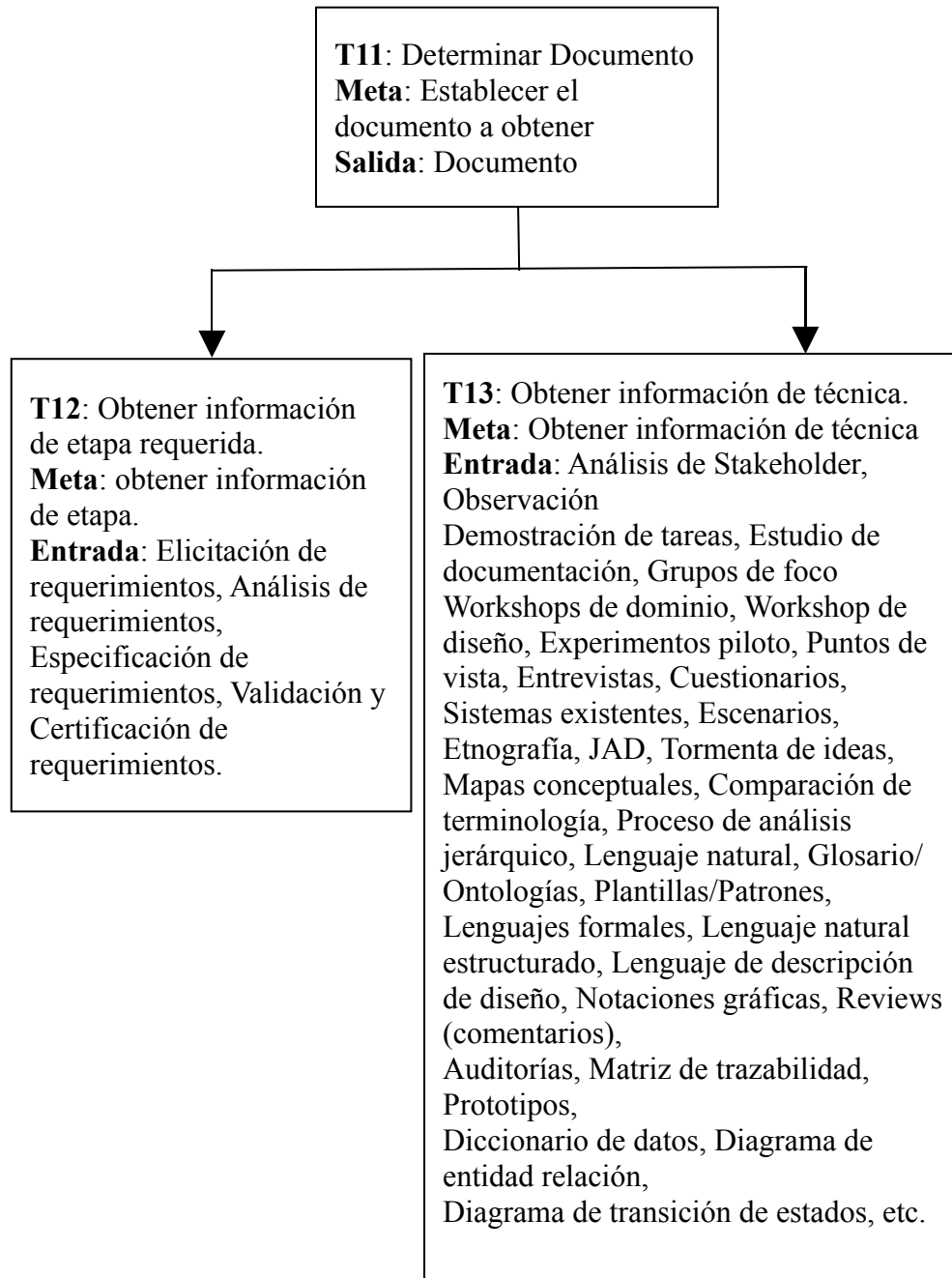
**Figura 4.15.** Diagrama de clases correspondiente a la tarea Determinar Técnica

La tarea “T9: Determinar Herramienta”, establece las herramientas adecuadas para cada etapa, de las cuales se obtiene información con la tarea “T10: Obtener Información de Etapa Requerida”.



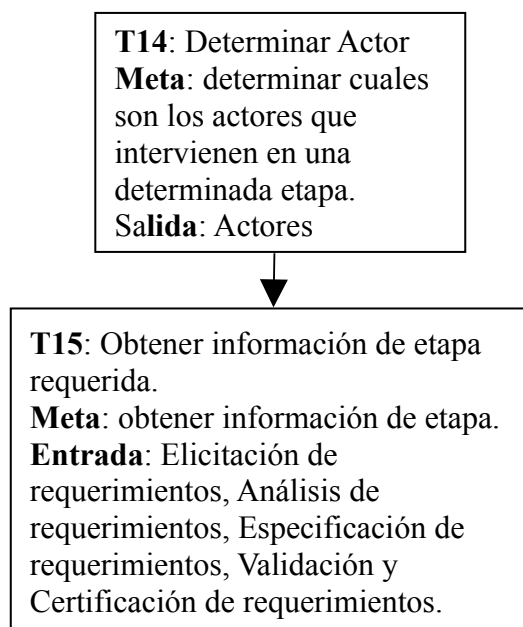
**Figura 4.16.** Diagrama de clases correspondiente a la tarea Determinar Herramienta

La tarea “T11: Determinar Documento”, establece el documento a obtener en cada etapa y técnica utilizada, de las cuales se obtiene información de las tareas “T12: Obtener Información de Etapa Requerida”, y “T13: Obtener Información de Técnica”, respectivamente.



**Figura 4.17.** Diagrama de clase correspondiente a la tarea Determinar Documento

La tarea “T14: Determinar Actor”, establece cuales son los actores que intervienen en cada una de las etapas, de las cuales se obtiene información con la tarea “T15: Obtener Información de Etapa Requerida”.



**Figura 4.18.** Diagrama de clase correspondiente a la tarea Determinar Actor

#### IV.9. ESQUEMA CONCEPTUAL

En el esquema conceptual quedan plasmadas las clases y subclases principales de la ontología, las relaciones existentes entre ellas y la cardinalidad de dichas relaciones.

En la figura 4. , se muestra lo siguiente:

- En una o muchas **etapas**, pueden intervenir uno o muchos **actores** en su realización.
- En una o muchas **etapas**, se pueden utilizar una o muchas **herramientas** para su realización.
- En una o muchas **etapas**, se pueden utilizar una o muchas **técnicas** para su concreción.
- Una **etapa**, produce un **documento** de salida.
- Uno o muchos **documentos** pueden estar formados por varios **contenidos de documentos**.
- Uno o muchos **documentos** están redactados de acuerdo a una o varias **normas de redacción**.

- En una o muchas **etapas** se pueden considerar uno o muchos **atributos** de calidad del proceso.

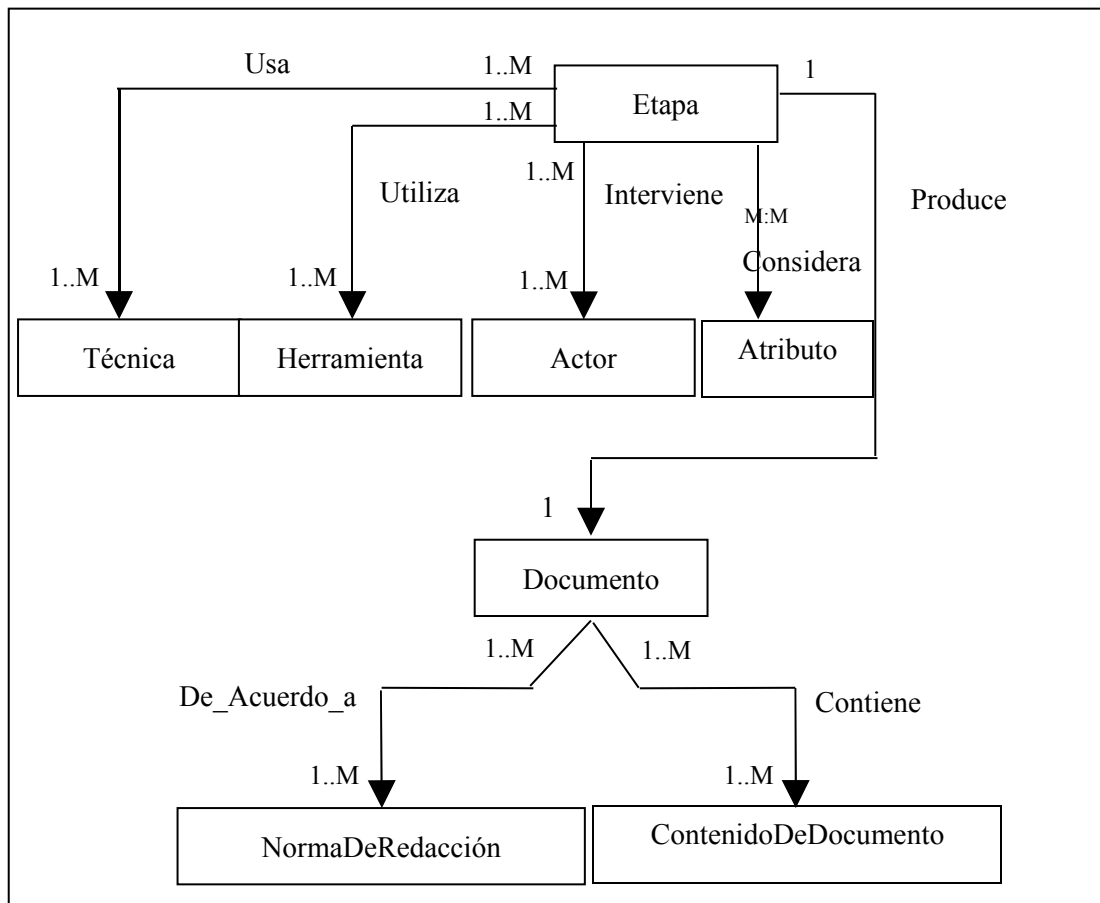


Figura 4.19. Esquema Conceptual de OntoIR

## V.10. CONCLUSIÓN

En este capítulo se diseñó la taxonomía de conceptos. Para ello se tomó como base los glosarios de términos obtenidos en el Capítulo 3: Especificación de Requisitos, en la etapa Adquisición de Conocimientos. Dicha taxonomía muestra la jerarquía de los conceptos del dominio, Figuras 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9 y 4.10.

En base a la taxonomía de conceptos definida se realizaron los diagramas de relaciones binarias. Los mismos muestran las relaciones existentes entre dichos conceptos. Las Figuras 4.11, 4.12 y 4.13 muestran los diagramas obtenidos.

A continuación se confeccionaron tablas de clases-subclases-propiedad. Las mismas muestran una descripción de las clases identificadas, subclases de las mismas y propiedades que las relacionan entre sí. Las Tablas creadas son: 4.1 y 4.2.

Para proporcionar información más específica de cada propiedad identificada se

construyó la Tabla de Propiedades en Detalle (Tabla 4.3).

Luego se confeccionó la Tabla de Reglas (Tabla 4.4), la misma muestra las relaciones necesarias y suficientes para inferir información referente al Proceso de IR.

A partir de la Tabla de Reglas se realizó la formulación de los axiomas de la ontología para arribar a la información referente al dominio en cuestión, obteniéndose así la Tabla de Axiomas (Tabla 4.5).

Se construyeron los Diagramas de clases para cada uno de los procesos involucrados en la ontología. Las siguientes Figuras muestran los diagramas arribados: 4.14, 4.15, 4.16, 4.17 y 4.18.

Por último, se construyó el Esquema conceptual de la ontología, Figura 4.19, que permite observar las clases y subclases de la misma, las relaciones existentes entre ellas y cardinalidad de dichas relaciones.

*Capítulo V:*  
*Formalización*  
*de*  
*OntoIR*



## V.1. INTRODUCCIÓN

En esta etapa se pretende realizar una representación formal, es decir, crear modelos formales que brinden una representación semi-computable de los conocimientos y conductas del experto que puedan ser utilizadas por una computadora.

Existen distintos tipos de representación que permiten formalizar los conocimientos conceptualizados; estas representaciones reciben el nombre de formalismos de representación. Aquí, se tuvo en cuenta que el formalismo de Marcos que es una de las técnicas más empleadas cuando el conocimiento del dominio está organizado en base a conceptos [66]. A través de formalismos de marcos se representaron los conceptos y sus atributos determinados en la fase conceptualización. La selección de los formalismos de representación para el presente trabajo se funda en su adecuación, claridad y eficiencia.

Los marcos permiten [67]:

- Formalizar los conocimientos estáticos de OntoIR de una manera sencilla y comprensible.
- Organizar los conceptos y sus atributos correspondientes en estructuras taxonómicas; esto facilita la representación de las clasificaciones de los principales conceptos que forman OntoIR.
- Con los marcos clases se representan los conceptos y situaciones genéricas de OntoIR.
- Con los marcos instanciados se representan las situaciones concretas de OntoIR.
- Expresar de una manera completa y sencilla las relaciones y las propiedades que caracterizan a cada concepto.
- Compartir propiedades y evitar conocimientos redundantes, a través de la herencia de propiedades.
- Almacenar gran cantidad de información asociada con las propiedades de cada concepto.

## V.2. MARCOS

La tabla 4.1 (Clases-Subclases-Propiedad de las clases principales) y la tabla 4.2 (Subclases pertenecientes a la clase principal *Recursos*), del Capítulo IV, se formalizan como el marco de una clase. En la siguiente tabla se muestra la correspondencia entre Clase, Subclase y Propiedad con el formalismo de marcos.

**Tabla 5.1.** Formalización de la Tabla Clase-Subclase-Propiedad

<b>Tabla clase-subclase- propiedad</b>	<b>Marco</b>
Clase	Marco clase
Subclase	Ranuras del marco clase
Propiedad	Relación entre los marcos

## V.3. ONTOLOGIA BASADA EN MARCOS

Se utilizan tres jerarquías de marcos unidas mediante relaciones “ad-hoc” para la representación de nuestra ontología. Estas jerarquías son:

1. Jerarquía de Etapas: establece el conjunto de etapas pertenecientes al proceso de IR. Cada instancia de esta jerarquía puede usar uno o más recursos para su desarrollo, a su vez cada etapa produce un documento de salida.
2. Jerarquía de Recursos: establece el conjunto de recursos de los cuales se prescinden para llevar a cabo cada una de las etapas a lo largo del proceso de IR. Un recurso puede ser útil para diferentes etapas.
  - Etapas, utilizando la relación “Hace\_uso\_de” para especificar la etapa a la cual es útil
3. Jerarquía de Documentos: establece el conjunto de documentos que se obtienen a lo largo del proceso de IR.
  - Etapas, utilizando la relación “Se\_obtiene” para especificar la etapa desde la cual se obtiene.

A continuación, en la Figura 5.1 se presenta el Esquema de Marcos de la Ontología OntoIR. |



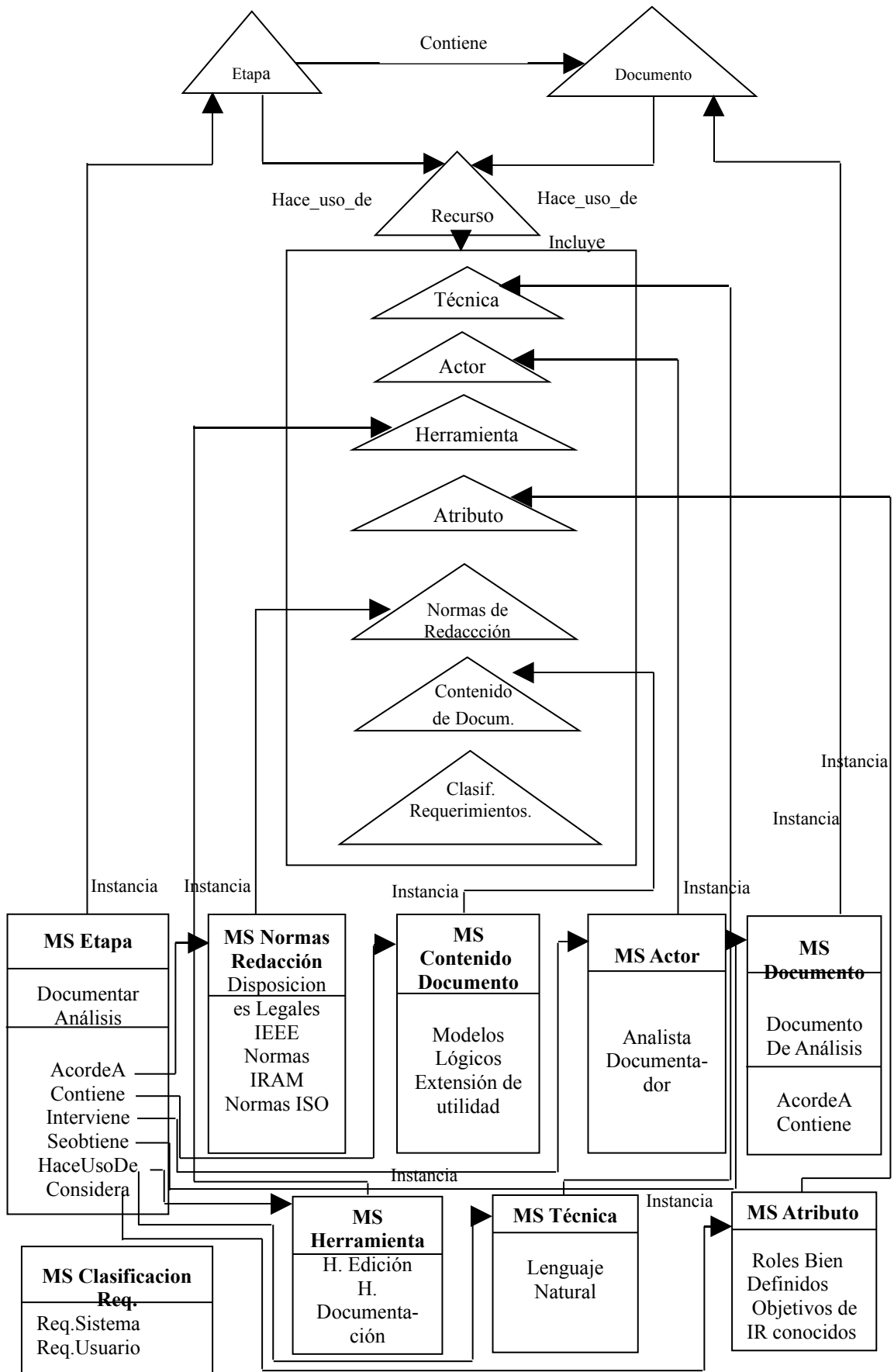
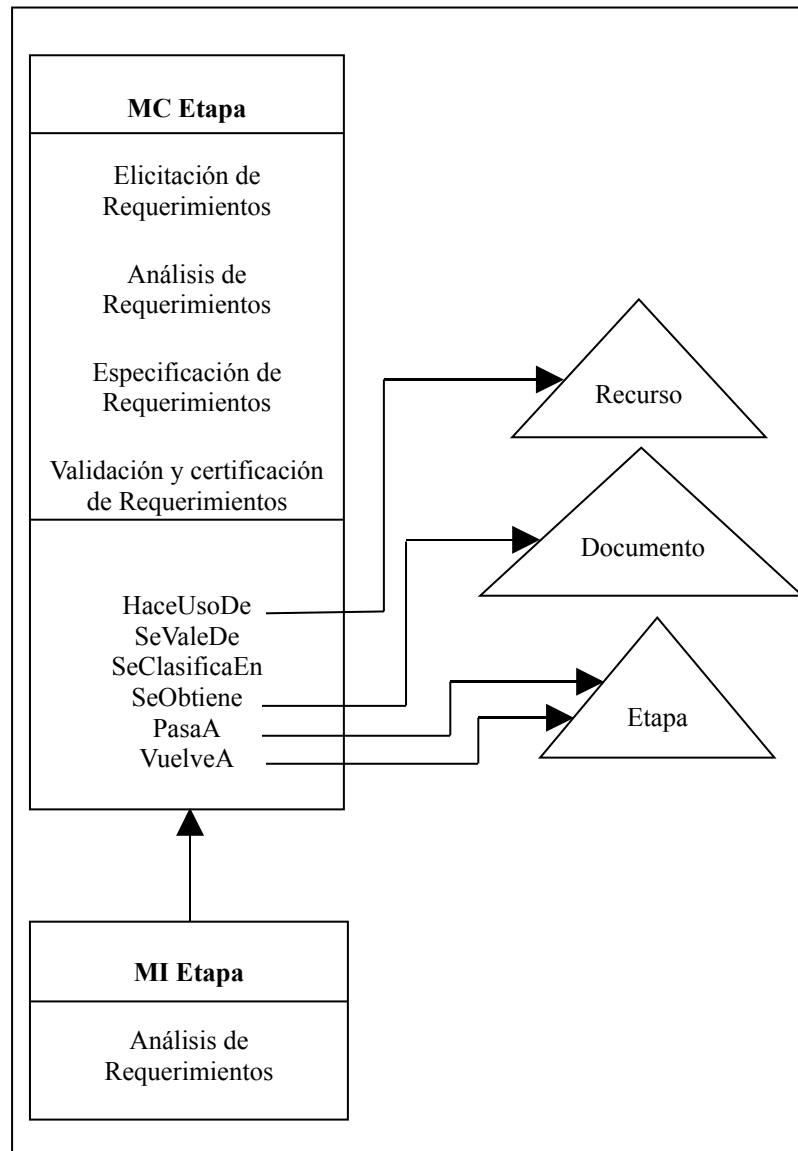


Figura 5.1. Esquema de marcos de OntoIR

## V.4. JERARQUÍA DE MARCOS

A continuación se describen las jerarquías de marcos correspondientes a los marcos clases de la Figura: “Esquema de Marcos de la Ontología”.



**Figura 5.2.** Jerarquía del marco clase Etapa

La Figura 5.2 muestra la Jerarquía del marco *Etapa*, una instancia del mismo y las relaciones existentes con los demás marcos: *Recurso*, *Documento* y *Etapa*. Cabe aclarar, que las relaciones que no están especificadas en la figura, tales como: *SeValeDe* y *SeClasificaEn*, se las nombra nada más porque las mismas relacionan la clase *Etapa* con subclases pertenecientes a la clase *Recurso*.

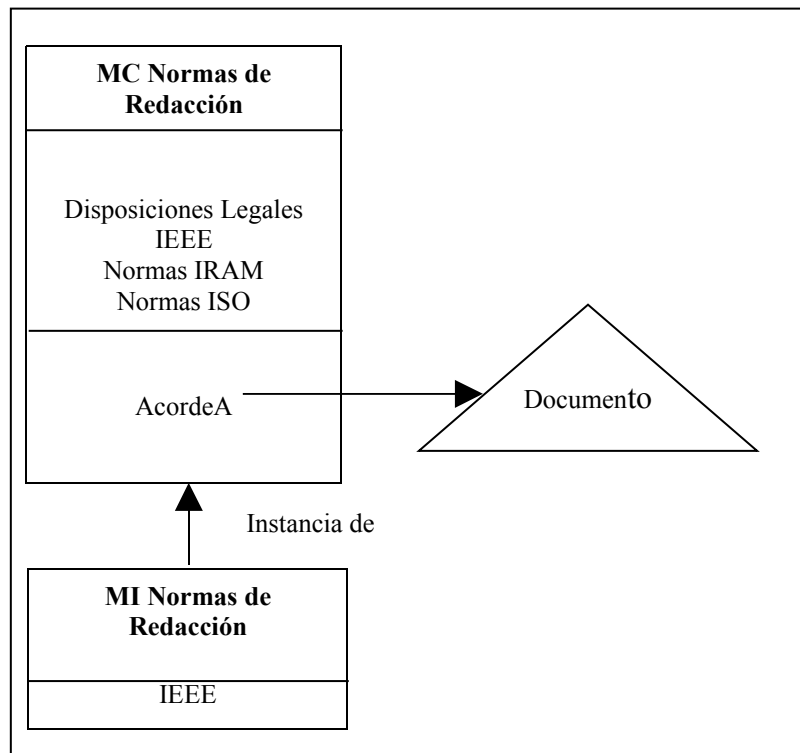


Figura 5.3. Jerarquía del marco clase Normas de Redacción

La Figura 5.3 muestra la jerarquía del marco *Normas de Redacción*, una instancia del mismo y la relación existente con el marco clase *Documento*.

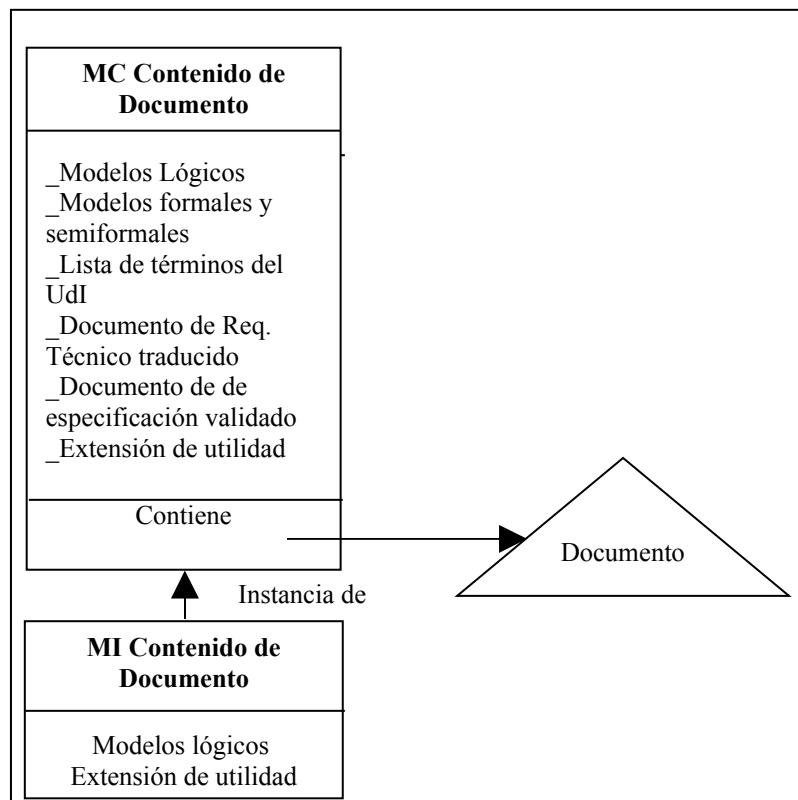
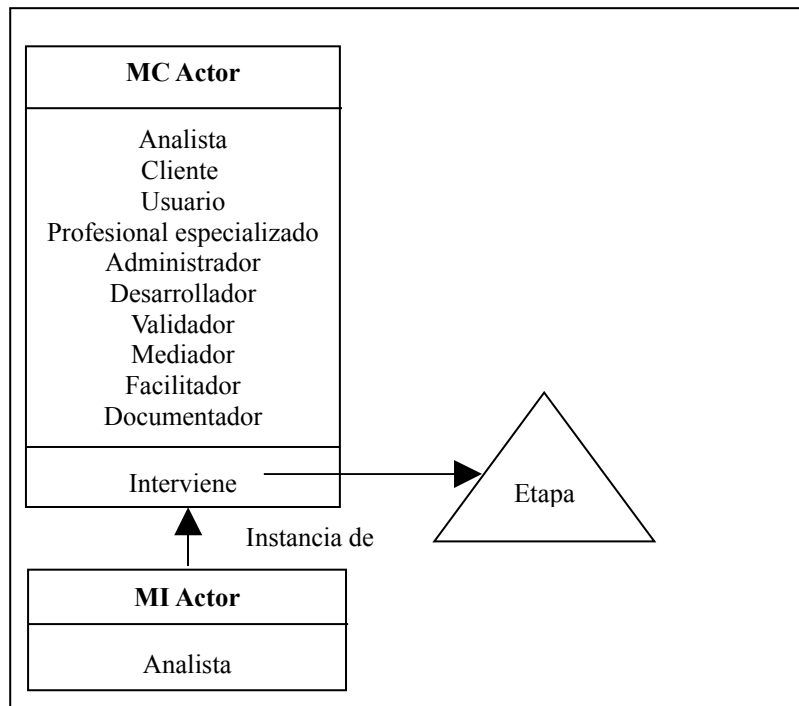


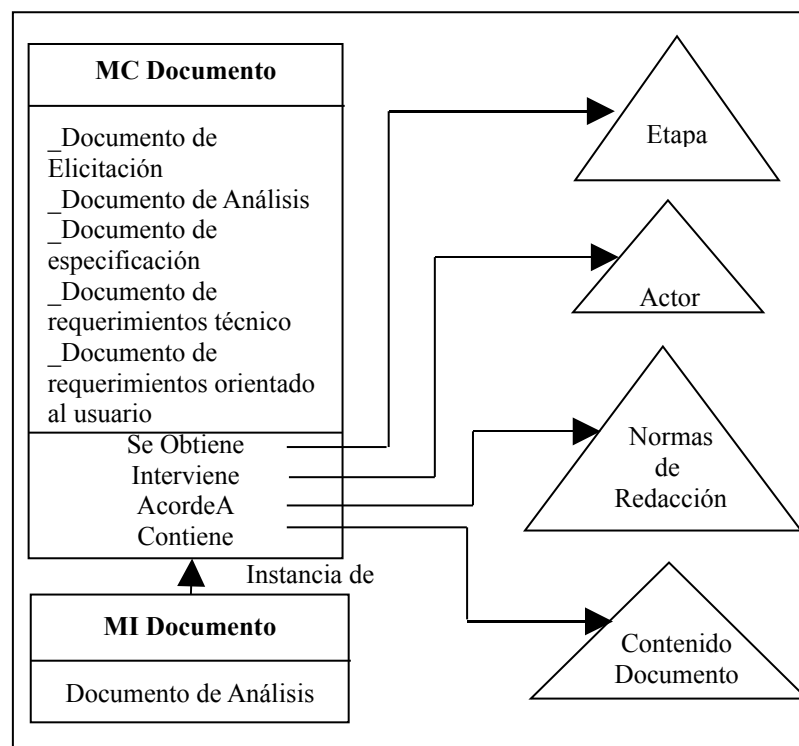
Figura 5.4. Jerarquía del marco clase Contenido de Documento

La Figura 5.4 muestra la jerarquía del marco *Contenido de Documento*, una instancia del mismo y la relación existente con el marco clase *Documento*.



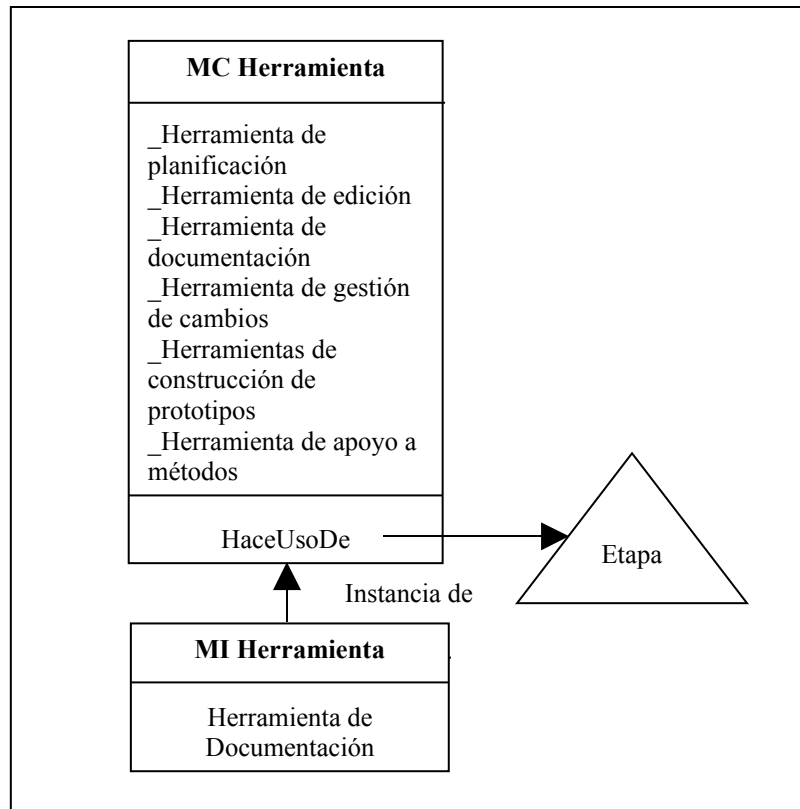
**Figura 5.5.** Jerarquía del marco clase Actor

La Figura 5.5 muestra la jerarquía del marco *Actor*, una instancia del mismo y la relación existente con el marco clase *Etapa*.



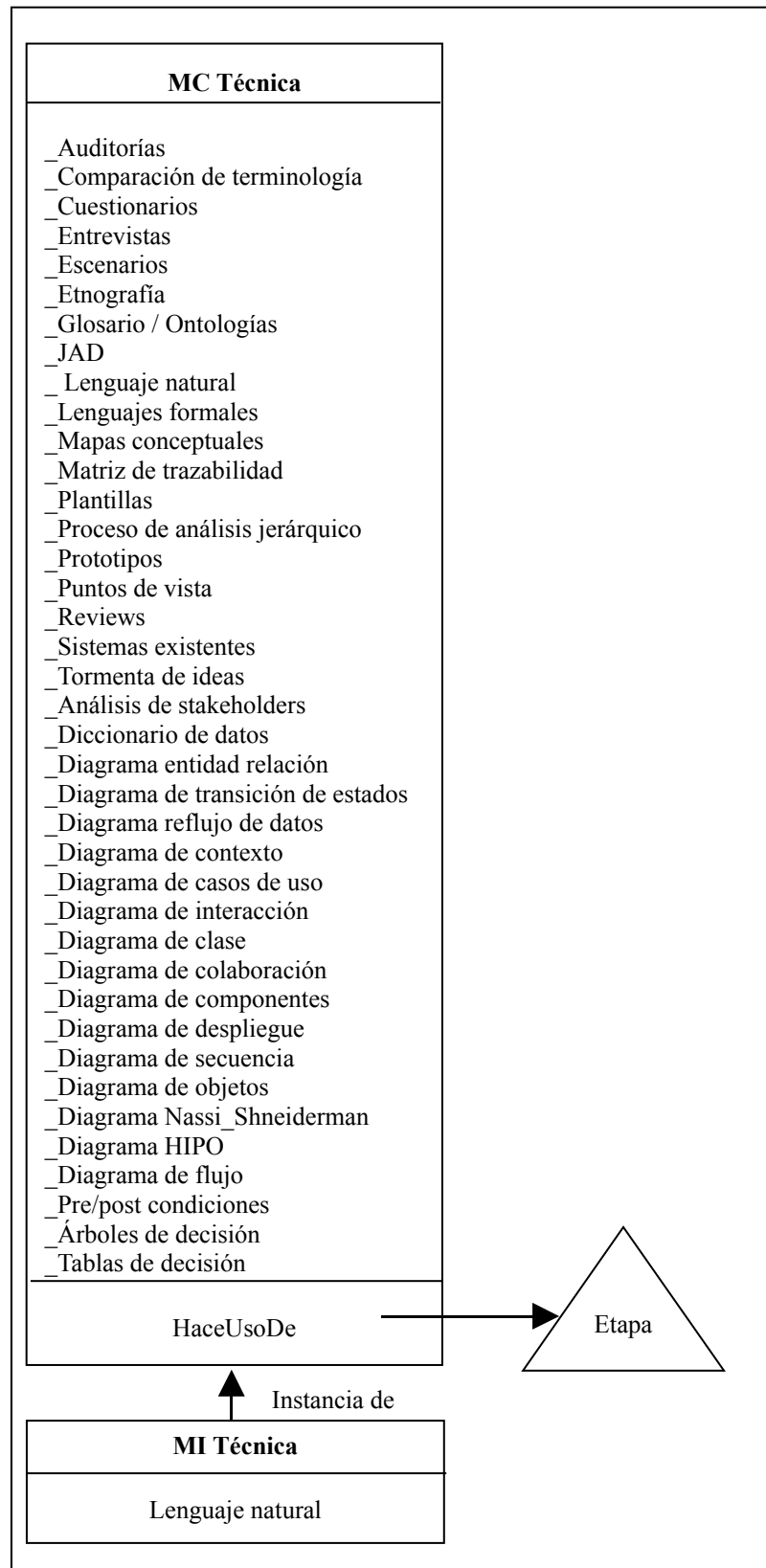
**Figura 5.6.** Jerarquía del marco clase Documento

La Figura 5.6 muestra la jerarquía del marco *Documento*, una instancia del mismo y las relaciones existentes con el marco clase *Etapa*, *Actor*, *Normas de Redacción* y *Contenido de Documento*.



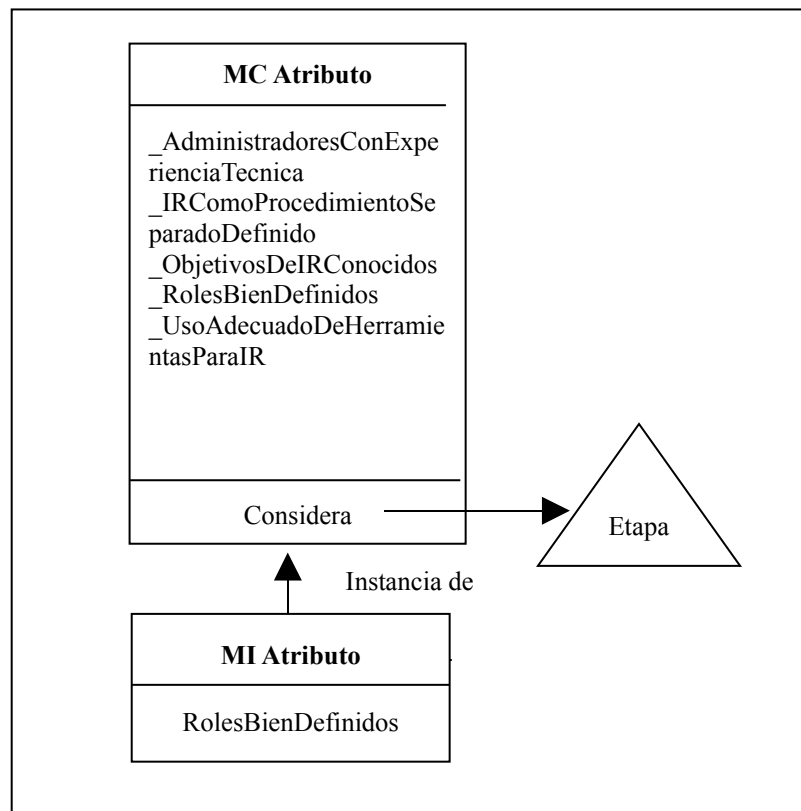
**Figura 5.7.** Jerarquía del marco clase Herramienta

La Figura 5.7 muestra la jerarquía del marco *Herramienta*, una instancia del mismo y la relación existente con el marco clase *Etapa*.



**Figura 5.8.** Jerarquía del marco clase Técnica

La Figura 5.8 muestra la jerarquía del marco *Técnica*, una instancia del mismo y la relación existente con el marco clase *Etapa*.



**Figura 5.9.** Jerarquía del marco clase Atributo

La Figura 5.9 muestra la jerarquía del marco *Atributo*, una instancia del mismo y la relación existente con el marco clase *Etapa*.

## V.5. CONCLUSIÓN

En el presente capítulo se realizó la Jerarquía de Marcos. Para poder arribar a la misma se tuvieron en cuenta las tablas de Clase, Subclases y Propiedades y la Tabla de Axiomas, todas ellas obtenidas en Capítulo 4.

Basándonos en la Jerarquía obtenida y las tablas anteriormente mencionadas se realizó la construcción de la ontología sobre la plataforma Protege OWL.

*Capítulo VI:*

*Construcción, Prueba  
y Validación de  
OntoIR*





## VI.1. INTRODUCCIÓN

En el presente capítulo se presenta una breve descripción de la construcción de OntoIR. Además, se realizan la prueba y la validación de la misma.

Para llevar a cabo tal objetivo, se tuvo en cuenta la opinión de los expertos y la documentación referente al dominio que abarca la ontología creada.

Se tendrán en cuenta los objetivos definidos para ver si se alcanzaron y en qué medida. Se considerarán los indicadores establecidos para la evaluación de cada uno de ellos.

Se analizará el Glosario de Términos, obtenido en el Capítulo 3, y se desarrollarán diferentes casos de prueba, junto con los expertos, para que sean analizados por los mismos, y así poder contrastar los resultados obtenidos con los esperados para arribar a las conclusiones correspondientes de la presente validación.

## VI.2. CONSTRUCCIÓN

Para poder volcar el conocimiento del dominio de la IR y poder compartirlo, ya sea entre agentes de software o personas, se construyó OntoIR utilizando el editor Protégé. El conocimiento representado en la ontología es el que consta en los Glosarios de términos adquiridos; para expresar las relaciones entre los esos términos se utilizó el lenguaje OWL (Web Ontology Language). OWL es un plugin de Protégé para OWL que permite desarrollar estos archivos (tipo owl), y provee de herramientas que facilitan la inferencia entre las clases permitiendo mantener correctamente la jerarquía entre ellas.

## VI.3. PRUEBA Y VALIDACIÓN

Teniendo en cuenta lo expuesto sobre *Organización para las pruebas del software* en el Capítulo 13 de [71], se realizaron con anterioridad, a los casos de pruebas presentados en la Tabla 6.1., distintas pruebas a OntoIR con el objetivo de corroborar el correcto funcionamiento de la misma. Dichas evaluaciones fueron creadas por las desarrolladoras de la Ontología y los casos de prueba fueron creados por los expertos encuestados.

### VI.3.1. OBJETIVOS FORMULADOS

Los objetivos formulados para OntoIR se encuentran definidos en el ítem III.3, del Capítulo 3: Especificación de Requisitos.

**Se definieron los siguientes indicadores para los objetivos formulados:**

- Para verificar si se cumplió con el objetivo específico a), se considera como indicador de medición al vocabulario / glosario integrado y consensuado de términos de OntoIR (Tablas: 3.1, 3.2, 3.3, 3.4, 3.5, 3.6 y 3.7 del Capítulo 3); como fuente de validación primaria se considera la opinión de los expertos (cuestionarios del Anexo C) y como fuente de validación secundaria se consideran documentos y estándares (Marcos Referenciales, Capítulo 2).
- Para verificar si se cumplió con los objetivos b) y d), se considera como indicadores de medición por una parte, técnicas, herramientas y demás recursos ofrecidos por OntoIR para cada etapa de la IR durante el uso de la misma (Tablas: 3.1, 3.2, 3.3, 3.4, 3.5, 3.6 y 3.7 del Capítulo 3); y por otra, se considera el tiempo insumido en la ejecución de OntoIR para cada caso de prueba. Como fuente de validación primaria se cuenta con los desarrolladores que ejecutaron OntoIR suministrando diversos casos de prueba reales o ficticios (Anexo A). La fuente de validación secundaria es la opinión de expertos (cuestionarios Anexo C). Se diseñó un cuestionario destinado a desarrolladores / expertos en IR.

Aspectos que se tuvieron en cuenta para analizar OntoIR:

- Grado de aceptación de la ontología por el experto: considera si todos los componentes, clases, relaciones y axiomas de OntoIR son suficientes para ayudar al proceso de IR.
- Tiempo de respuesta: tiempo en que OntoIR tarda en inferir un resultado (Racer).
- Grado de concordancia entre resultados esperados y obtenidos.
- Para verificar si se cumplió con el objetivo específico c), se consideran los indicadores de calidad de los procesos de IR definidos como la Subclase Atributos en OntoIR.

## **VI.4. VERIFICACIÓN DE OBJETIVOS**

### **VI.4.1. Verificación del objetivo a)**

Acompañando a OntoIR se entregó, a los expertos que realizaron las pruebas, el glosario integrado y consensuado de términos para su evaluación, a fin de determinar si se logró cumplir el objetivo planteado.

Tomando en cuenta la cantidad de respuestas positivas otorgadas por los expertos, a las preguntas 7 y 8 del cuestionario, se concluye que los términos utilizados por OntoIR son adecuados y contribuyen a lograr un entendimiento compartido entre los expertos durante el proceso de IR. Además, la consulta de la documentación disponible para la creación del glosario de términos, garantiza que éste contiene información válida y adecuada para el proceso de IR.

#### VI.4.2.Verificación de los objetivos b) y d)

Como fuente de validación primaria se ejecutaron los diferentes casos de prueba suministrados por los desarrolladores que colaboraron con la prueba de OntoIR.

Para corroborar la consistencia de la misma se diseñaron los casos de prueba 8, 9 y 10. Los mismos son casos de prueba inválidos que no corresponden a ninguna etapa del Proceso de IR contempladas por la ontología.

En la siguiente tabla se muestran los diferentes casos de prueba suministrados:

Nota:

NS: Condición Necesaria y Suficiente. N: Condición Necesaria.

**Tabla 6.1.** Casos de Prueba Creados

Casos de Prueba	Contenido
Prueba_1	NS: Hace uso de: Herramientas de edición o Sistemas Existentes. Actor que interviene: Facilitador N: Se Consideran: Atributos de Calidad
Prueba_2	NS: Acorde a: Disposiciones Legales o Estándar_ IEEE. Contiene: Lista términos del universo de información Contiene: Modelos lógicos. Hace uso de: Herramientas de edición. Actor que interviene: Documentador. Documento que se obtiene: Documento de Elicitación. Documento que se obtiene: Documento de Análisis. N: Se Consideran: Atributos de Calidad

**Tabla 6.1.** Casos de Prueba Creados (continuación)

Casos de Prueba	Contenido
Prueba_3	<p>N.S:</p> <p>Hace uso de: Diagramas de Clases o Herramientas de edición.</p> <p>Actor que interviene: Analista.</p> <p>Clasificación de requerimientos: la propuesta en ClasificacionRequerimientos</p> <p>N:</p> <p>Se Consideran: Atributos de Calidad</p>
Prueba_4	<p>NS:</p> <p>Acorde a: Estandar_ IEEE.</p> <p>Hace uso de: Proceso de Análisis Jerárquico.</p> <p>Actor que interviene: Analista, Desarrollador o Validador.</p> <p>Para realizar la etapa se vale de: Documento de Requerimiento Orientado Usuario.</p> <p>Para realizar la etapa se vale de: Documento de Requerimiento Técnico.</p> <p>Para realizar la etapa se vale de: Documento de Elicitación.</p> <p>N:</p> <p>Se Consideran: Atributos de Calidad</p>
Prueba_5	<p>N.S:</p> <p>Acorde a: Disposiciones Legales o Estandar_ IEEE.</p> <p>Hace uso de: Herramientas de documentación o Herramientas de edición.</p> <p>Actor que interviene: Documentador o Analista.</p> <p>Documento que se obtiene: Documento de Requerimientos Orientado al Usuario.</p> <p>N:</p> <p>Se Consideran: Atributos de Calidad</p>
Prueba_6	<p>N.S:</p> <p>Hace uso de: Herramientas de gestión de cambio.</p> <p>Actor que interviene: Analista, Cliente, Usuario, Administrador o Mediador.</p> <p>N:</p> <p>Se consideran: atributos de calidad</p>

**Tabla 6.1.** Casos de Prueba Creados (continuación)

Casos de Prueba	Contenido
Prueba_7	N.S: Hace uso de: Herramientas de edición o Herramientas de apoyo a métodos. Actor que interviene: Analista o Desarrollador. N: Se consideran: Atributos de Calidad
Prueba_8	Hace uso de: Herramientas de apoyo a métodos, Etnografía o Auditorías Actor que interviene: Desarrollador o Mediador. Etapa a la que se pasa: Etapa2 Análisis de Requerimientos
Prueba_9	N.S: Contenido del documento: Lista términos del universo e información, Modelos formales / semiformales o Documento de especificación validado. Actor que interviene: Cliente o Profesional especializado. Documento que se obtiene: Documento de Análisis o Documento de Requerimientos Orientado Usuario. N: Se consideran: Atributos de Calidad
Prueba_10	Hace uso de: Herramientas de construcción de Prototipo, Diagramas HIPO o Diagramas de Secuencia. Actor que interviene: Usuario. Para realizar la etapa se vale de: Documento de Especificación o Documento de Requerimientos Técnico. Etapa a la que se vuelve: Etapa3 Especificación de Requerimientos.

A continuación se muestran las pantallas de la ejecución de dos casos de prueba, uno válido (Prueba\_1) y otro inválido (Prueba\_10). Los demás casos, junto con sus resultados, se muestran en el Anexo A.

## Caso de Prueba\_1

En la Figura 6.1 se expone el caso de Prueba\_1 instanciado con los axiomas.

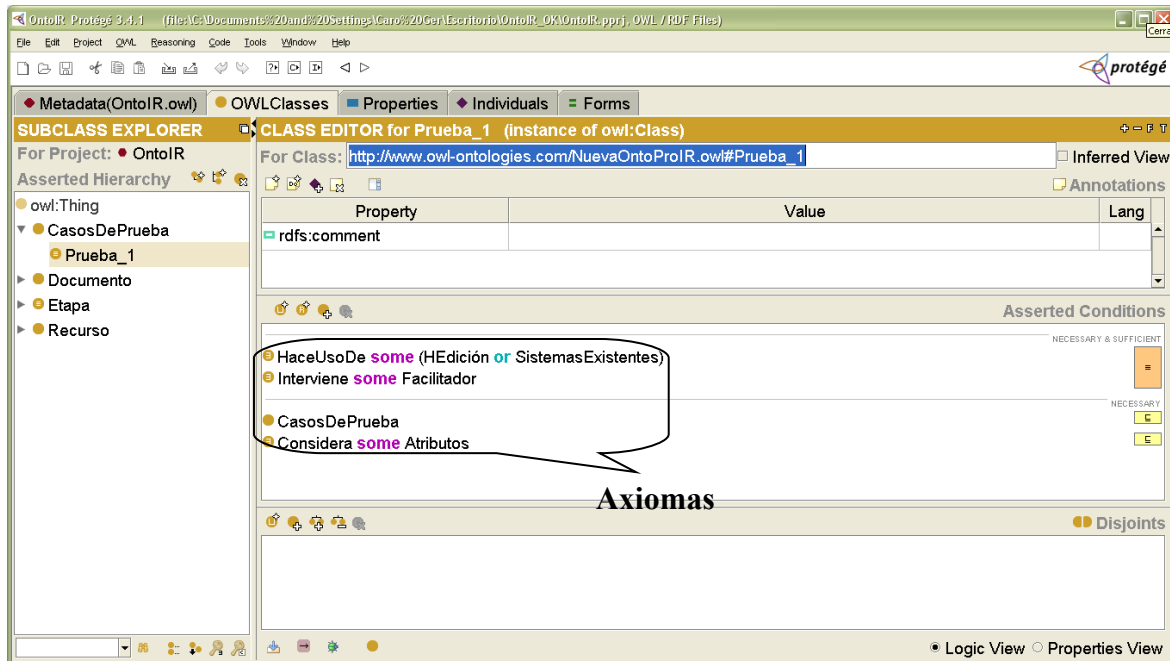


Figura 6.1. Prueba\_1

Para ejecutar la ontología se utilizó el razonador “Racer”.

En la figura 6.2 se observa el movimiento del caso de prueba\_1 hacia la subetapa 1.2: Buscar hechos.

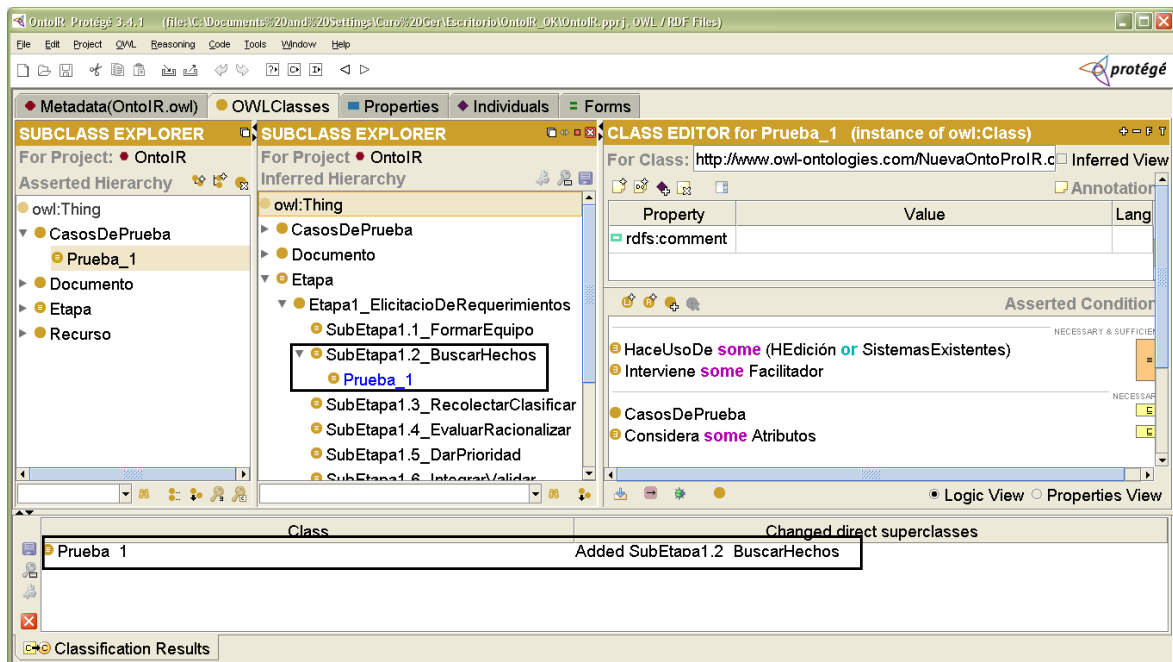


Figura 6.2. Ejecución del caso de prueba\_1

La figura 6.3 muestra el tiempo insumido por la ontología en dar un resultado.

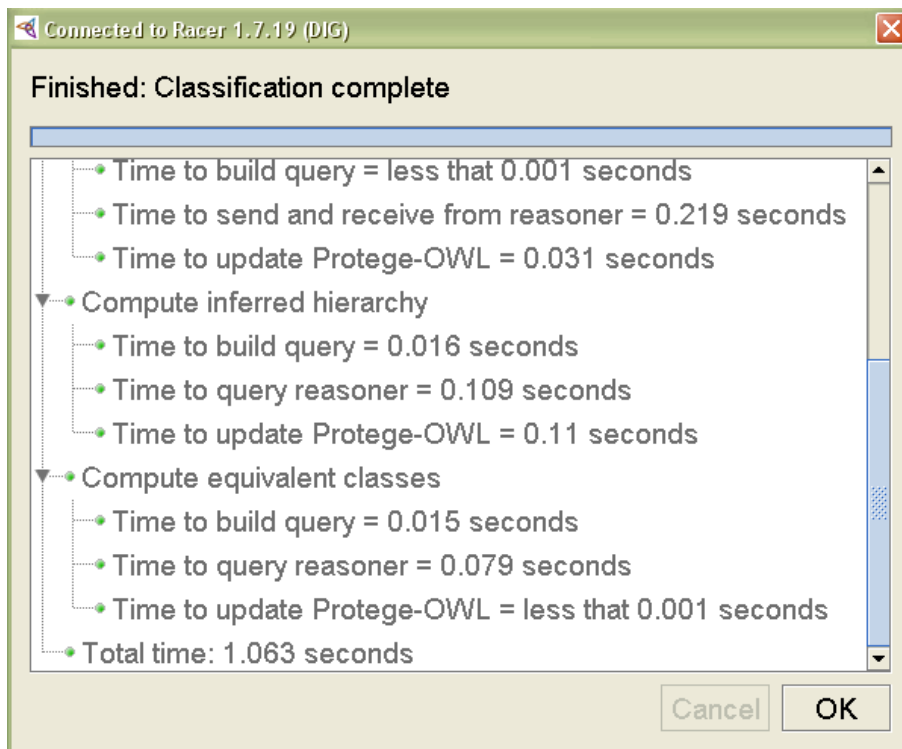


Figura 6.3. Tiempo de ejecución de prueba\_1

### Caso de Prueba\_10

En la Figura 6.4 se expone el caso de Prueba\_10 instanciado con los axiomas correspondientes al mismo.

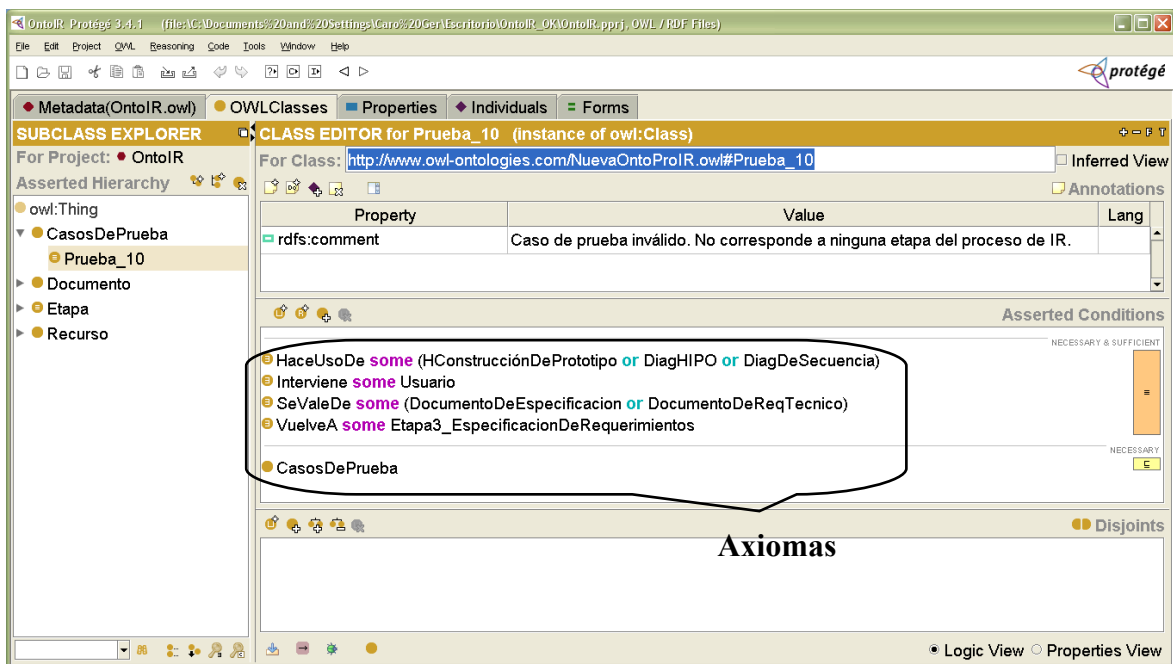


Figura 6.4. Prueba\_10

En la Figura 6.5 se observa que el caso de prueba\_10, luego de la ejecución de la ontología, no se movió hacia ninguna etapa en particular debido a que se trata de un caso de prueba inválido.

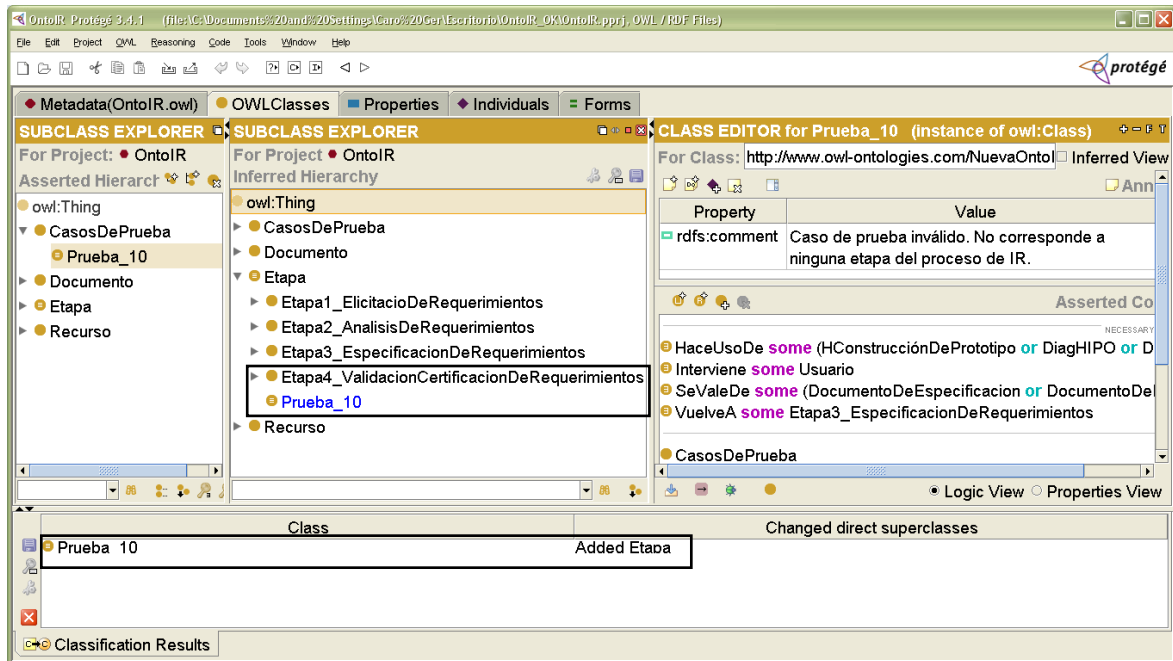


Figura 6.5. Ejecución del caso de prueba\_10

La figura 6.6 muestra el tiempo insumido por la ontología en dar un resultado para el presente caso.

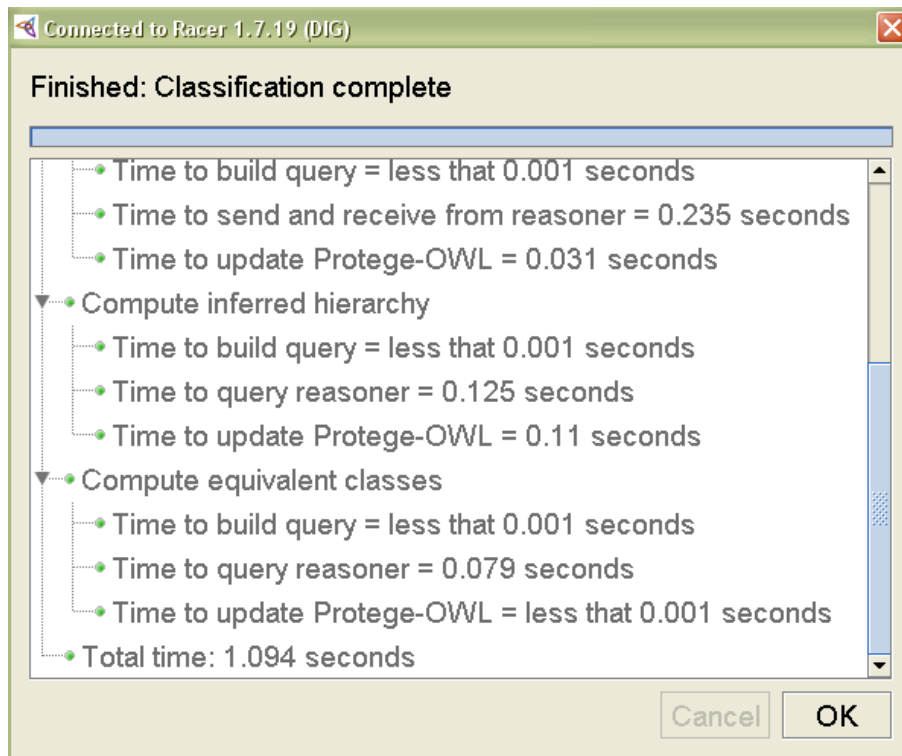


Figura 6.6. Tiempo de ejecución de prueba\_10



Se creó una tabla con los resultados que arrojó la ejecución de OntoIR, para cada caso de prueba, la misma nos servirá para realizar el análisis de los resultados obtenidos.

**Tabla 6.2.** Resultados de la ejecución de los casos de prueba

Casos de Prueba	Resultados Esperados	Resultados Obtenidos	Tiempo de Ejecución insumido
Prueba_1	Subetapa1.2: Buscar Hechos	Subetapa1.2	1.063 seg.
Prueba_2	Subetapa1.7: Documentar Elicitación Subetapa2.4: Documentar Análisis	Subetapa1.7 Subetapa2.4	1.047 seg.
Prueba_3	Subetapa3.1: Determinar tipo de requerimiento Subetapa3.2: Especificar acorde a herramienta	Subetapa3.1 Subetapa3.2	1.062 seg
Prueba_4	Subetapa4.4: Validar	Subetapa4.4	1.047 seg.
Prueba_5	Subetapa4.3: Elegir herramienta y documentar	Subetapa4.3	1.125 seg.
Prueba_6	Subetapa1.3: Recolectar y Clasificar Subetapa1.4: Evaluar y Racionalizar	Subetapa1.3 Subetapa1.4	1.078 seg.
Prueba_7	Subetapa2.1: Reducir ambigüedad Subetapa2.2: Traducir a lenguaje técnico los requerimientos Subetapa2.3: Modelo lógico	Subetapa2.1 Subetapa2.2 Subetapa2.3	1.047 seg.

**Tabla 6.2.** Resultados de la ejecución de los casos de prueba (Continuación)

Casos de Prueba	Resultados Esperados	Resultados Obtenidos	Tiempo de Ejecución insumido
Prueba_8	No debe ser movida a ninguna etapa en particular.	Etapa	1.125 seg.
Prueba_9	No debe ser movida a ninguna etapa en particular.	Etapa	1.062 seg.
Prueba_10	No debe ser movida a ninguna etapa en particular.	Etapa	1.094 seg.

#### VI.4.2.1. Análisis de la ejecución de los casos de prueba

Teniendo en cuenta la cantidad de respuestas positivas proporcionadas por los expertos encuestados, para la pregunta 4 del cuestionario, se concluye que la información referente a los diferentes recursos necesarios para llevar a cabo el proceso de IR, otorgada por OntoIR, es suficiente y adecuada.

Por otro lado, al observar la Tabla 6.2, se puede ver que los resultados esperados y obtenidos de cada ejecución de los casos de prueba suministrados son idénticos por lo que se deduce el buen funcionamiento de OntoIR.

Por todo esto, se concluye que OntoIR es una herramienta que garantiza la calidad del proceso de IR, otorga a los expertos información correcta, completa y oportuna. Por lo tanto, se puede decir que los objetivos b y d se lograron.

#### VI.4.3.Verificación del objetivo c)

Para verificar si OntoIR ayuda a maximizar la calidad del proceso de IR se tuvieron en cuenta los Atributos definidos en la ontología, la consistencia del programa y los tiempos de ejecución del mismo.

Considerando las respuestas otorgadas por los expertos a las preguntas 1, 2, 3, 5, 6 y 10 del cuestionario, se concluye que OntoIR contribuye a garantizar un Proceso de IR de calidad.

Teniendo en cuenta los indicadores de calidad propuestos a partir de la lectura de los artículos [69] y [70], además observando los tiempos de ejecución reflejados en la Tabla 6.2 en la que se puede ver que los mismos son mínimos, se asegura que OntoIR puede garantizar que la calidad del proceso de IR se maximice.

## **VI. 5. CONCLUSIÓN**

A partir de la ejecución de las pruebas realizadas por las desarrolladoras y los casos de prueba efectuados por los expertos encuestados, se pudo observar una alta confiabilidad en los resultados arrojados por lo cual se considera que dichas pruebas son suficientes para determinar que OntoIR funciona de manera adecuada.

Observando los resultados obtenidos de las ejecuciones de los diferentes casos de pruebas realizados para comprobar el cumplimiento de los objetivos definidos inicialmente, se concluye lo siguiente:

- OntoIR logra un entendimiento compartido entre los desarrolladores de un producto software en la etapa de IR.
- OntoIR orienta a los desarrolladores en la elección de técnicas, herramientas y demás recursos competentes a la IR.
- OntoIR garantiza un proceso de IR de calidad.
- OntoIR brinda asistencia a los desarrolladores a lo largo del proceso de IR.

Por lo tanto, se alcanzaron los objetivos planteados.

## *Conclusión Final*



## **CONCLUSIONES**

Observando los problemas existentes al momento de llevar a cabo el proceso de IR y los problemas en los que se puede incurrir al llevar a cabo a dicho proceso de manera inadecuada, la principal finalidad del presente trabajo fue diseñar y construir OntoIR, de tal manera de brindar un soporte para garantizar la calidad del proceso de IR.

En principio, para llevar a cabo la investigación, se realizó una investigación exploratoria y descriptiva del dominio de interés.

Mediante la adquisición de conocimientos se pudo obtener las bases teóricas y metodológicas sobre las cuales se sustenta el trabajo y los términos del dominio de interés que forman parte de la ontología.

Se definieron los objetivos y requisitos funcionales de OntoIR en base a las necesidades encontradas en el dominio de interés.

Se realizó la conceptualización del dominio para organizar la percepción informal de la especificación de dicho dominio. Para la formalización de dicha conceptualización se utilizaron los marcos que permitieron representar los conocimientos declarativos del dominio y los axiomas para representar los aspectos procedimentales.

El entorno de desarrollo utilizado fue Protégé y el lenguaje OWL que permitieron definir la semántica y precisar el comportamiento lógico de la ontología.

Para verificar si se arribaron a los objetivos definidos, se tuvieron en cuenta como indicadores de medición a los glosarios de términos, las técnicas, las herramientas y los demás recursos ofrecidos por OntoIR, y los indicadores de calidad del proceso; como fuente de validación de dichos indicadores, se consideró la opinión de los expertos como así también el análisis de Documentos y Estándares, se diseñaron diversos casos de prueba para analizar el grado de aceptación, el tiempo de respuesta y el grado de concordancia entre resultados esperados y obtenidos de la ejecución de la ontología.

Los principales aportes de este trabajo son:

- Favorecer al mejoramiento de la toma de decisiones de los desarrolladores.
- Brindar un marco común para llevar a cabo las actividades de IR.

- Contribuir al reuso de la información dado que ya se cuenta con un modelo conceptual definido.
- Contribuir al uso y reuso de ontologías en la Ingeniería de Software.

# *Bibliografía*



## Referencias

- [1] Pressman, Roger S. “Ingeniería del software, Un enfoque práctico”; 3ª Ed., McGraw-Hill, 1995.
- [2] Frank, A.U. “Spatial Ontology: A Geographical Point of View”, O. Stock (ed.), Spatial and Temporal Reasoning, Dordrecht: Kluwer Academic Publishers, 135–153, 1997.
- [3] Sommerville Ian, Ingeniería del Software; 7ª Ed. Ed. Pearson - Addison Wesley. Madrid. 2005.
- [4] Adquisición de conocimientos.  
Disponble en: <http://www.webpages.ull.es/users/jmmoreno/IG/Tema2.ppt>.  
Fecha de acceso: (15/12/2005)
- [5] Golbreich Christine, Bierlaire Olivier, Dameron Olivier, Gibaud Bernard, “Use Case: Ontology with Rules for identifying brain anatomical structures”, StanfordUniversity School of Medicine, Stanford CA 94305, USA.  
Disponble en: <http://www.w3.org/2004/12/rules-ws/paper/64/> .  
Fecha de acceso: (25/05/2006)
- [6] Drummond Nick, Moulton Georgina, Stevens Robert, Lord Phil, The University of Manchester, 2005.  
Disponble en: <http://www.co-ode.org/ontologies/amino-acid/2005/10/11/amino-acid.owl>  
y  
<http://www.daml.org/cgi-bin/dumppont?http://www.co-de.org/ontologies/amino-acid/2005/10/11/amino-acid.owl>.  
Fecha de acceso: (28/03/2006)
- [7] Hoss Allyson M. “Metodología basada en Ontología para la detección de errores en el diseño de Software”. Universidad del Estado de Louisiana, Departamento de Ciencias de la Computación, Agosto de 2006..  
Disponble en: [http://www.etd.lsu.edu/docs/available/etd-7102006103349/unrestricted/Hoss\\_dis.pdf](http://www.etd.lsu.edu/docs/available/etd-7102006103349/unrestricted/Hoss_dis.pdf).  
Fecha de acceso: (09/11/2006)
- [8] Mendes Olavo. “Ontología para la Ingeniería de Software”. (UFPB) Universidad Federal de Paraiba, Brasil. Abran Alain. (ETS) Escuela de Tecnología Superior, Canadá.  
Disponble en: <http://www.lrgl.uqam.ca/publications/pdf/839.pdf>.  
Fecha de acceso: (09/11/2006)
- [9] Damiani Ceravolo, P, Marchesi E., Pinna M., Zavatarelli S. “A ontology-basedProcess Modelling for XP”. Departamento de Tecnología de la Información, Universidad di



Milano, Italy.

Disponible en: <http://portal.acm.org/citation.cmf?id=956416.956527&coll=&dl=GUIDE&CFID=15151515&CFTOKEN=6184618#collab>.

Fecha de acceso: (20/01/2007)

[10] Pérez María A, Mendoza Luis E., Grimán Anna. “Hacia una ontología para fábricas de software”. Laboratorio de Investigación en Sistemas de Información, Dpto. de Procesos y Sistemas, Universidad Simón Bolívar, Caracas, Venezuela.

Disponible en: [http://www.Lisi.usb.ve/publicaciones/02%calidad%systemica/calidd\\_42.pdf](http://www.Lisi.usb.ve/publicaciones/02%calidad%systemica/calidd_42.pdf)

Fecha de acceso: (11/10/2007)

[11] Vizcaíno Aurora, Soto Juan Pablo, García Felix, Ruiz Francisco, Piattini Mario. “Aplicando Gestión del Conocimiento en el Proceso de Mantenimiento del Software”. Grupo ALARCOS, Departamento de Tecnologías y Sistemas de Información, Centro Mixto de Investigación y Desarrollo de Software UCLM-Solucionara, Universidad de Castilla, La Mancha (España).

Disponible en: <http://cabrillo.lsi.uned.es:8080/aepia/uploads/31/337.pdf>.

Fecha de acceso: (11/10/2007)

[12] OBITKO, M. 2003. “Ontologies. Description and Applications”.

Disponible en: <http://cyber.felk.cvut.cz/gerstner/reports/GL126.pdf>

Fecha de acceso: (15/05/2005).

[13] Pressman Roger S. “Ingeniería del software, Un enfoque práctico”; 5ª Ed., McGraw-Hill, 2002.

[14] Guarino N. "Understanding, Building, and Using Ontologies", Knowledge Acquisition Workshop, 1996.

[15] Van Heijst, Schreiber G, y Wielinga, B.J. "Using Explicit Ontologies in KBS Development". International Journal of Human and Computer Studies, 1996.

[16] Durán A., Bernández B. “Metodología para la Elicitación de Requisitos de Software”, Universidad de Sevilla. 2002.

Disponible en: <http://gimnasiobl.googlecode.com/files/Articulo.doc>.

Fecha de acceso: (05/10/2009)

[17] Goguen J. “Requirements Engineering as the Reconciliation of Social and Technical Issues”. Academic Press. 1994.

Disponible en: <http://gimnasiobl.googlecode.com/files/ARTICULO.pdf>.

Fecha de acceso: (05/10/2009)

[18] Pohl K. “Requirements Engineering: An Overview”. Encyclopedia of Computer Science and Technology. 1997.

Disponible en: <http://www.springerlink.com/index/t0642h44v7036180.pdf>.

Fecha de acceso: (10/10/2009)

[19] Palavecino Rosa, Álvarez Margarita M., Figueroa Liliana, “MeCoOn: Metodología para la construcción de Ontologías”, Proyecto de Investigación, UNSE.

[20] Young Ralph R., “The Requirements Engineering Handbook”, Ed. Artech House, 2004.

[21] Leite, J. C. S. P. “A Survey on Requirement Analysis”. Advancing Software Engineering Project Technical report RTP-071, University of California at Irvine, Department of Information and Computer Science, Junio 1987.

Disponible en: <http://requirementsviewpoints.googlepages.com/Viewpoint.pdf>.

Fecha de acceso: (10/10/2009)

[22] Loucopoulos P. y Champion R. E. M. “Knowledge – Based Support for Requirements Engineering”. Information and Software Technology. Vol. 31, num 3, Abril 1989.

[23] Institute of Electrical and Electronics Engineers. “IEEE Standard Glossary of Software Engineering Terminology. IEEE Standard 610”. 12 – 1990 (revision and redesignation of IEEE Std. 729 - 1983). New York, 1990.

[24] Dorfman M. y Thayer R. “Software Engineering”. IEEE Computer Society Press. Los Alamitos, CA, 1997.

[25] Ingeniería de Requerimientos - Ingeniería de Software – Monografias.com.

Disponible en: <http://www.monografias.com/trabajos6/resof/resof.shtml>.

Fecha de acceso: (28/11/2005)

[26] M.J.Escalona, Koch N. “Ingeniería de Requisitos en aplicaciones para la web: un estudio comparativo”. Universidad de Sevilla. Lenguajes y sistemas informáticos. España. Universidad de Munich y F.A.S.T GMBH, Munich, Alemania.

Disponible en: <http://www.lsi.us.es/docs/informes/LSI-2002-4.pdf>.

Fecha de acceso: (15/12/2005)

[27] Ontologías.

Disponible en: <http://www.hipertexto.info/documentos/ontologias.htm>.

Fecha de acceso: (16/12/2005)

[28] Golbreich Christine, Bierlaire Olivier, Dameron Olivier, Gibaud Bernard, “Use Case:

Ontology with Rules for identifying brain anatomical structures”. Stanford University School of Medicine, Stanford CA 94305, USA.

Disponible en: <http://www.w3.org/2004/12/rules-ws/paper/64/>.

Fecha de acceso: (25/05/2006)

[29] Corcho Oscar, Fernández-López Mariano, Gómez-Pérez Asunción, López-Cima Angel. “Building legal ontologies with METHONTOLOGY and WebODE”. Facultad de Informática. Universidad Politécnica de Madrid, Campus de Montegancedo, s/n. 28660 Boadilla del Monte. Madrid. Spain.

Disponible en: [http://www.cs.man.ac.uk/~ocorcho/documents/LawSemWeb2004\\_CorchoEtAl.pdf](http://www.cs.man.ac.uk/~ocorcho/documents/LawSemWeb2004_CorchoEtAl.pdf)

Fecha de acceso: (28/03/2008)

[30] “Ingeniería de Requisitos”. Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla.

Disponible en: [http://www.sli.us.es/docencia/pagina\\_asignatura.php?id=48](http://www.sli.us.es/docencia/pagina_asignatura.php?id=48).

Fecha de acceso: (15/12/2005)

[31] Ontologías.

Disponible en: <http://www.lsi.upc.es/~bejar/ia/material/teoria/3-rc2-ontologias-2.pdf>.

Fecha de acceso: (18/03/2006)

[32] Muñoz Silva Lydia, “Estudio de ontologías para representação de conteúdos de ensino baseado na www”. Porto Alegre. Septiembre de 2002.

Disponible en: [http://www.inf.ufrgs.br/~tapejara/publicacoes/TI\\_Lydia.pdf](http://www.inf.ufrgs.br/~tapejara/publicacoes/TI_Lydia.pdf).

Fecha de acceso: (24/04/06).

[33] “Técnicas y Herramientas utilizadas en la Ingeniería de Requerimientos”.

Disponible en: <http://www.monografias.com/trabajos6/resof/tecnicasyherramientas.shtml>

Fecha de acceso: (28/11/2005).

[34] Báez M. Griselda, Brunner Silvia I. “Metodología DoRCU para la Ingeniería de Requerimientos”. Instituto Superior politécnico “José Antonio Echeverría”. La Habana, CU, Facultad de Ciencia y Tecnología, Universidad Autónoma de Entre Ríos, AR.

Disponible en: [http://wer.inf.puc-rio.br/WERpapers/artigos/artigos\\_WER01/baez.pdf](http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER01/baez.pdf).

Fecha de acceso: (30/10/2006).

[35] Hoss Allyson M. “Metodología basada en ontología para la detección de errores en el diseño de software”. Universidad del Estado de Louisiana, Departamento de Ciencias de la Computación. Agosto 2006.

Disponible en: [http://etd.lsu.edu/docs/available/etd07102006103349/unrestricted/Hoss\\_dis.pdf](http://etd.lsu.edu/docs/available/etd07102006103349/unrestricted/Hoss_dis.pdf).

Fecha de acceso: (09/11/2006)

[36] Fridman Natalya, and Monica Crubézy Noy, and Monica Crubézy, “Protégé-2000: A flexible and extensible ontology-editing environment”.

Disponible en: <http://www-sop.inria.fr/acacia/ekaw2000/demo.html>.

Fecha de acceso: (10/09/2006)

[37] What is Protégé 2000?

Disponible en: [http://protege.stanford.edu/doc/users\\_guide/index.html](http://protege.stanford.edu/doc/users_guide/index.html).

Fecha de acceso: (09/09/2006)

[38] Mizoguchi R. “Ambientes para desarrollo de ontologías”.

Disponible en: <http://www.ei.sanken.osaka-u.ac.jp/pub/miz/HandBookMiz3.pdf>.

Fecha de acceso: (09/09/2006)

[39] Frankovic Baltasar, Budinska Ivana, “El rol de las ontologías en la construcción de sistemas de conocimientos para aplicaciones industriales”.

Disponible en: <http://www.bmf.hu/conferences/sami2006/Frankovic.pdf>.

Fecha de acceso: (10/09/2006)

[40] “La Ingeniería de Requerimientos y su importancia en el desarrollo de proyectos de software”. InterSedes: Revista de las sedes regionales, año/vol. VI, número 10 y 11. Universidad de Costa Rica.

Disponible en: <http://redalyc.uaemex.mx/redalyc/pdf/666/66661111.pdf>

Fecha de acceso: (10/10/2009)

[41] Raisa Socorro, Alfredo Simón, Reinier Valdés, Félix O. Fernández, Alejandro Rosete, Mailyn Moreno, Exiquio Leyva, Joaquín Pina. “Las ontologías en la representación del conocimiento”. Centro de Estudios de Ingeniería de Sistemas (CEIS), Instituto Superior Politécnico “José Antonio Echeverría” (CUJAE).

Disponible en: <http://nopiedra.files.wordpress.com/2008/05/rep-con-ontologias.pdf>

Fecha de acceso: (23/09/2008)

[42] Ontologías en Documentación.

Disponible en: <http://personales.upv.es/ccarrasc/doc/2001-2002/Ontologias/INICIO.htm>.

Fecha de acceso: (10/05/2008)

[43] Dña. Inés Friss de Kereki Guerrero. Tesis Doctoral: “Modelo para la Creación de Entornos de Aprendizaje basados en técnicas de Gestión del Conocimiento”. FACULTAD DE INFORMÁTICA de la UNIVERSIDAD POLITÉCNICA DE MADRID para la

obtención del GRADO DE DOCTOR EN INGENIERÍA INFORMÁTICA.

Disponible en: <http://www.ort.edu.uy/fi/pdf/Tesis.pdf>.

Fecha de acceso: (15/10/2009)

[44] Jesús Contreras ISOCO, Juan Antonio Martínez Comeche. “Tutorial ontologías”. Universidad Complutense de Madrid.

Disponible en: [www.sedic.es/gt\\_normalizacion\\_tutorial\\_ontologias.pdf](http://www.sedic.es/gt_normalizacion_tutorial_ontologias.pdf).

Fecha de acceso: (10/10/2009)

[45] Noy N. and Mc Guinness D. “Ontology development 101: A Guide to creating your first ontology”. Stanford University. Stanford knowledge Systems Laboratory. Technical Report KSL-01-05. (2001).

Disponible en: [http://protege.stanford.edu/.../ontology\\_development/ontology101.pdf](http://protege.stanford.edu/.../ontology_development/ontology101.pdf)

Fecha de acceso: (10/10/2009)

[46] Ramos Esmeralda y Nuñez Haydemar. “ONTOLOGÍAS: componentes, metodologías, lenguajes, herramientas y aplicaciones”. Universidad Central de Venezuela Facultad de Ciencias Escuela de Computación.

Disponible en: <http://dircompucv.ciens.ucv.ve/Documentos/RT-2007-12.pdf>.

Fecha de acceso: (08/05/ 2008)

[47] Ger Sylvia Carolina. “Ontología para el Diagnóstico de Trastornos Mentales” (ODiTMe). Trabajo Final de Graduación. Licenciatura en Sistemas de Información. UNSE.

[48] Yourdon Edward. “Análisis Estructurado Moderno”. Ed. Prentice\_Hall Hispanoamericana S.A. Mexico.

[49] Jacobson Ivar, Booch Grady, Rumbaugh James. “UML, El proceso unificado de desarrollo de software”. Ed. Addison Wesley. España.

[50] Gane-Sarson. “Análisis Estructurado de Sistemas”. Ed. El Ateneo. Buenos Aires.

[51] ¿Whats is protege?

Disponible en: <http://protege.stanford.edu/overview/index.html>.

Fecha de acceso: (12/10/2007).

[52] What is protege owl?

Disponible en: <http://protege.stanford.edu/overview/protege-owl.html>.

Fecha de acceso: (12/10/2007).

[53] OWL Web Ontology Language.

Disponible en: <http://www.w3.org/TR/owl-features/>.

Fecha de acceso: (15/10/2007).

[54] Ontologías 2, Ontologías en acción, Protege, OWL.

Disponible en: <http://www.matem.unam.mx/~rajsbaum/cursos/web/ontologias2.pdf>.

Fecha de acceso: (15/10/2007).

[55] OWL Web Ontology Language Guide.

Disponible en: <http://www.w3.org/TR/owl-guide/>.

Fecha de acceso: (15/10/2007).

[56] Lauesen Soren, “Software Requirement. Styles and Techniques”. Ed. Addison Wesley.

[57] Wikipedia, la enciclopedia libre. Etnografía.

Disponible en: <http://es.wikipedia.org/wiki/Etnografia>

Fecha de acceso: (25/10/2008)

[58] Ingeniería de Requerimientos – Ingeniería de Software.

Disponible en: <http://www.emagister.com/ingenieria-requerimientos-ingenieria-software-cursos-316276.htm>

Fecha de acceso: (17/08/2008)

[59] Sommerville Ian. “Ingeniería del Software”. 3° Ed. Ed. Pearson - Addison Wesley.

[60] Maté José Luis, Silva Andrés. “Requirements Engineering for Sociotechnical Systems”. Universidad Politécnica de Madrid. Idea Group Inc. 2005.

[61] Wikipedia, la enciclopedia libre. IEEE

Disponible en: [http://es.wikipedia.org/wiki/Computer\\_Society](http://es.wikipedia.org/wiki/Computer_Society)

Fecha de acceso: (12/01/2009)

[62] Wikipedia, la enciclopedia libre. Normas ISO

Disponible en: [http://es.wikipedia.org/wiki/Normas\\_ISO](http://es.wikipedia.org/wiki/Normas_ISO)

Fecha de acceso: (12/01/2009)

[63] Organización Internacional para la Estandarización

Disponible en: [http://www.wikipedia.es/enciclopedia/Organizacion\\_Internacional\\_para\\_la\\_Estandarizacion](http://www.wikipedia.es/enciclopedia/Organizacion_Internacional_para_la_Estandarizacion)

Fecha de acceso: (12/01/2009)

[64] Wikipedia, la enciclopedia libre. Normas IRAM

Disponible en: [http://es.wikipedia.org/wiki/Norma\\_IRAM](http://es.wikipedia.org/wiki/Norma_IRAM)

Fecha de acceso: (12/01/2009)

[65] Guía de usuario Enterprise Architect 7.1.

Disponible en: <http://www.sparxsystems.com.ar/download/Ayuda%20HTML/index.html?classdiagram.htm>

Fecha de acceso: (10/03/2009)

[66] Sistema Experto aplicado al control del espacio aéreo.

Disponible en: [www.itba.edu.ar/.../50cacic2004-Sistema-Experto-Control-Espacio-Aereo.pdf](http://www.itba.edu.ar/.../50cacic2004-Sistema-Experto-Control-Espacio-Aereo.pdf)

Fecha de acceso: (25/03/2008)

[67] Rosenzvaig Federico, “Modelo de captura y representación del conocimiento quirúrgico experto en patología biliares”. Trabajo Final de Graduación en Licenciatura de Sistemas de Información. UNSE.

[68] Gómez Asunción, Juristo Natalia, Montes César, Pazos Juan. “Ingeniería del Conocimiento”. Ed. Centro de Estudios Ramón Areces S.A. Madrid.

[69] Palyagar Bhavani. “Measuring and Influencing Requirements Engineering Process Quality in Organizations”. Information and Communication Sciences. Macquarie University. Sydney, NSW 2109 Australia.

[70] Gorschek Tony, Davis Alan M. “Requirements engineering: In search of the dependent variables”. Information and Software Technology 50 (2008) 67–75. ScienceDirect.

[71] Pressman, Roger S. “Ingeniería del Software”; 6ª Ed, Ed. McGraw-Hill, 2005.

[72] Construcción de ontologías OWL.

Disponible en: <http://gemini.udistrital.edu.co/comunidad/dependencias/revistavinculos/VINCULOS/revista/7edicion/22007702.pdf>

Fecha de acceso: (20/05/2010)

# *Anexo A*





## A.1. CASOS DE PRUEBA

En el presente anexo se publican los restantes casos de prueba que fueron mencionados en el Capítulo 6: Construcción, Prueba y Validación de OntoIR.

### Caso de Prueba\_3

En la Figura A.1 se expone el caso de Prueba\_3 instanciado con los axiomas correspondientes al mismo.

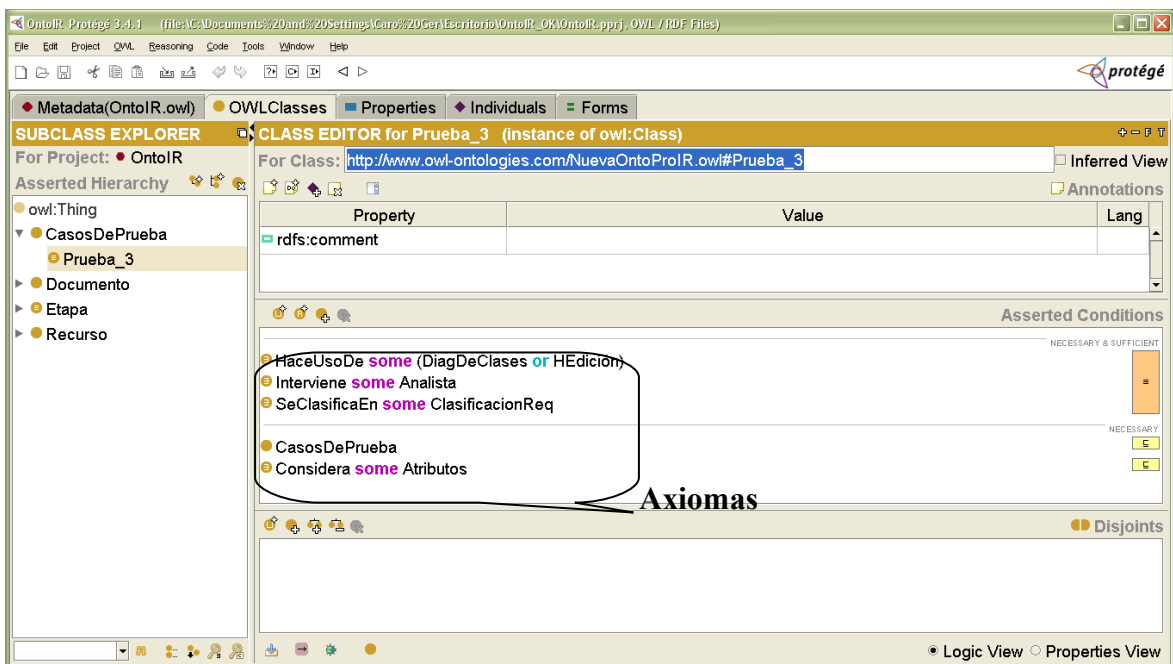


Figura A.1. Prueba\_3

En la Figura A.2 se observa el movimiento del caso de prueba\_3, luego de la ejecución de la ontología, hacia la subetapa3.1 Determinar tipo de requerimiento.

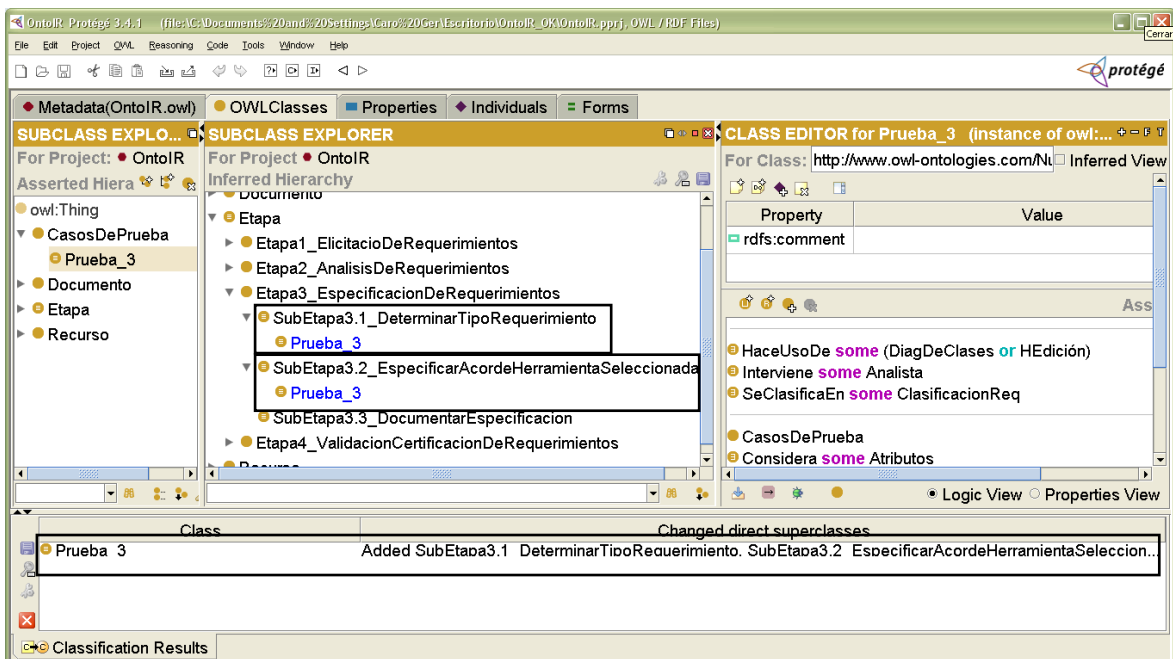


Figura A.2. Ejecución de casos de Prueba\_3

La figura A.3 muestra el tiempo insumido por la ontología en dar un resultado para el presente caso.

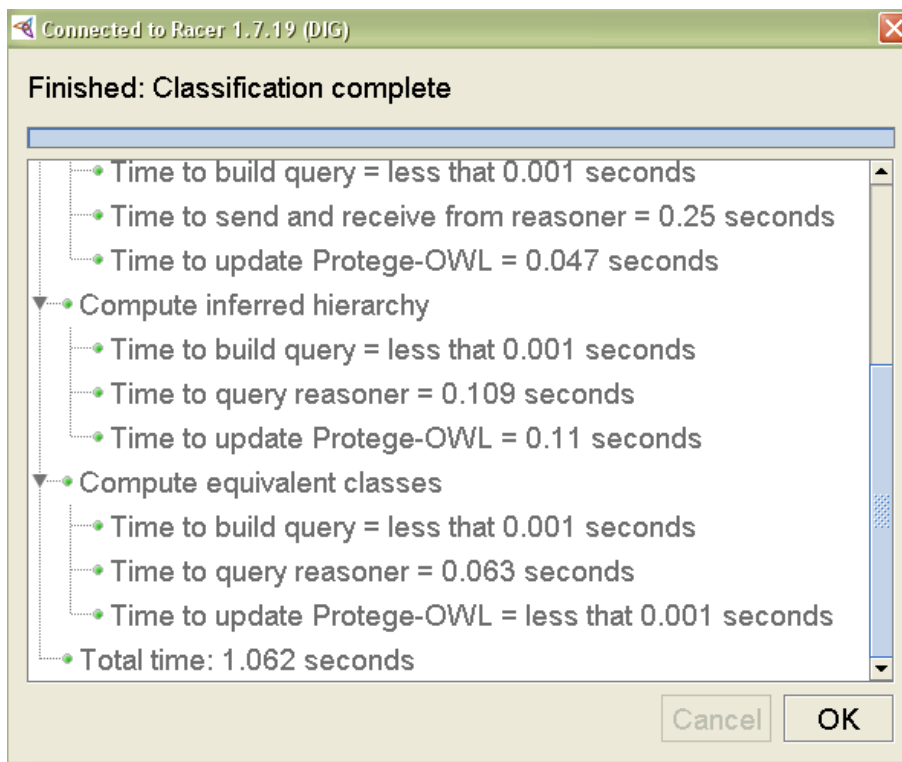


Figura A.3. Tiempo de ejecución de prueba\_3

### Caso de Prueba\_4

En la Figura A.4 se expone el caso de Prueba\_4 instanciado con los axiomas correspondientes al mismo.

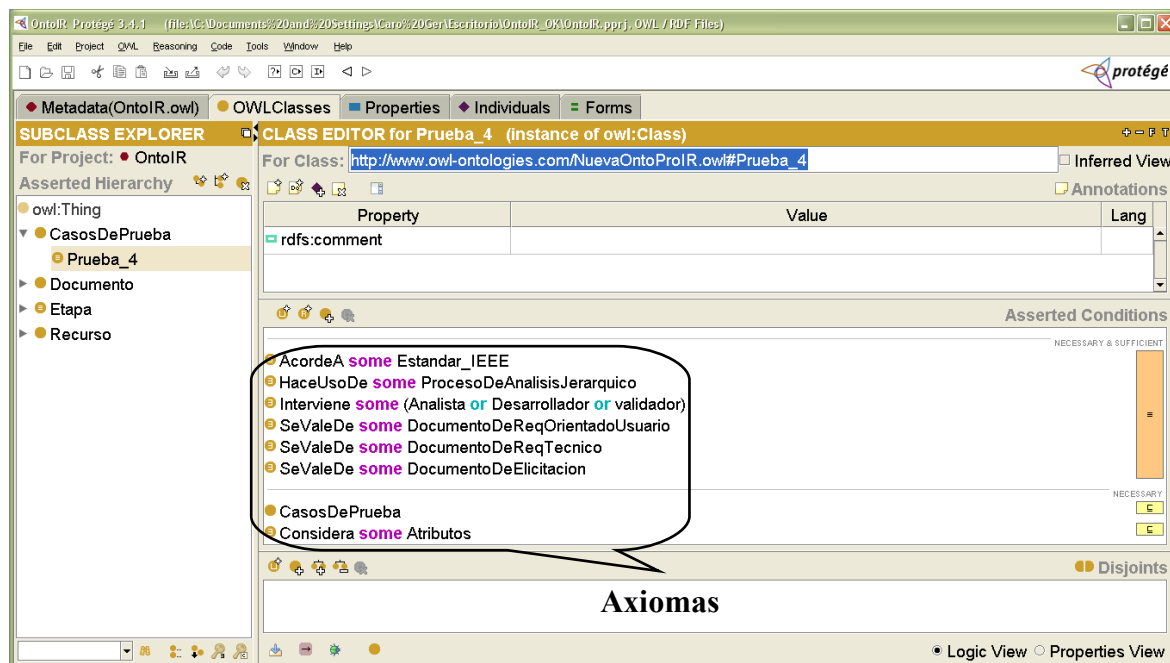


Figura A.4. Prueba\_4

En la Figura A.5 se observa el movimiento del caso de prueba\_4, luego de la ejecución de la ontología, hacia la subetapa4.4 Validar.

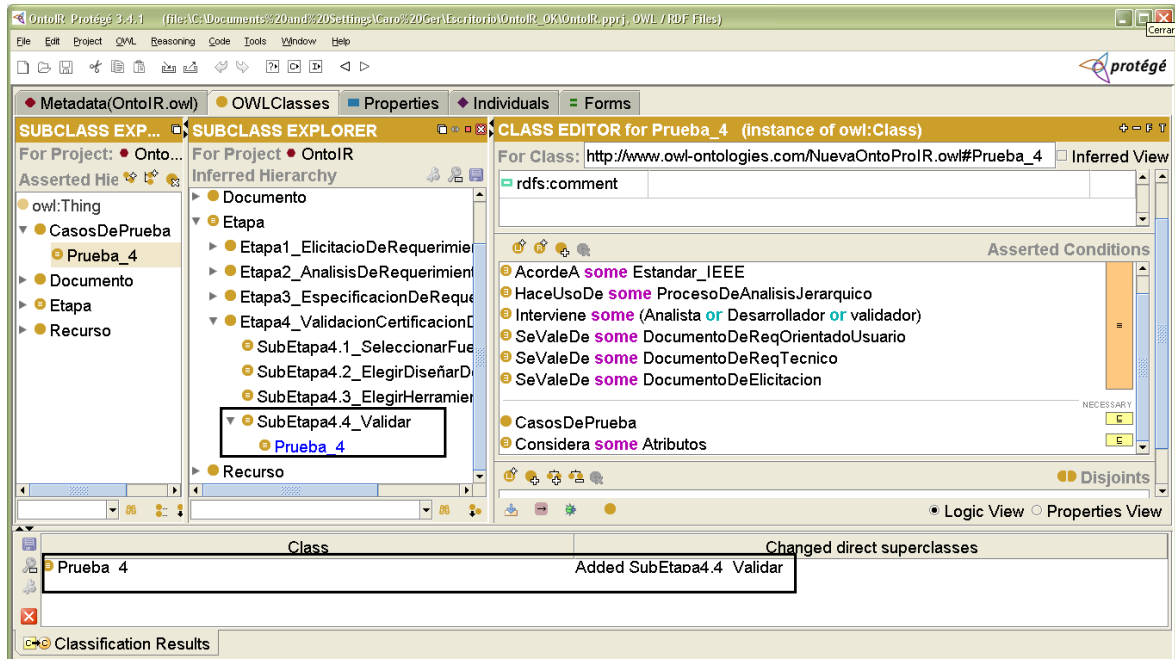


Figura A.5. Ejecución del caso de prueba\_4

La figura A.6 muestra el tiempo insumido por la ontología en dar un resultado para el presente caso.

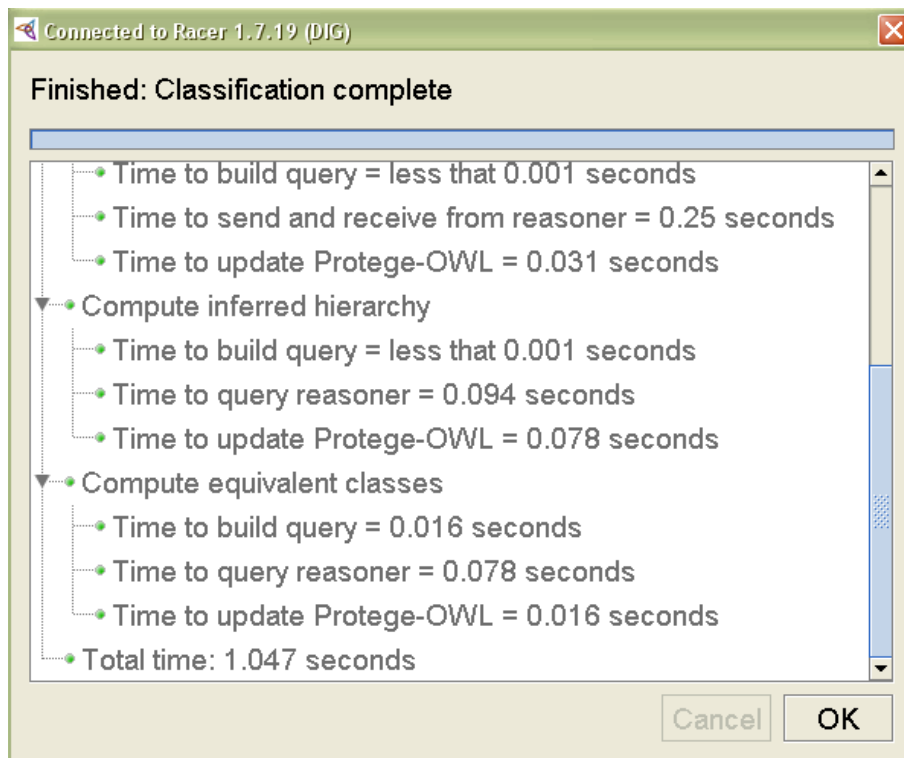


Figura A.6. Tiempo de ejecución de prueba\_4

### Caso de Prueba\_5

En la Figura A.7 se expone el caso de Prueba\_5 instanciado con los axiomas correspondientes al mismo.

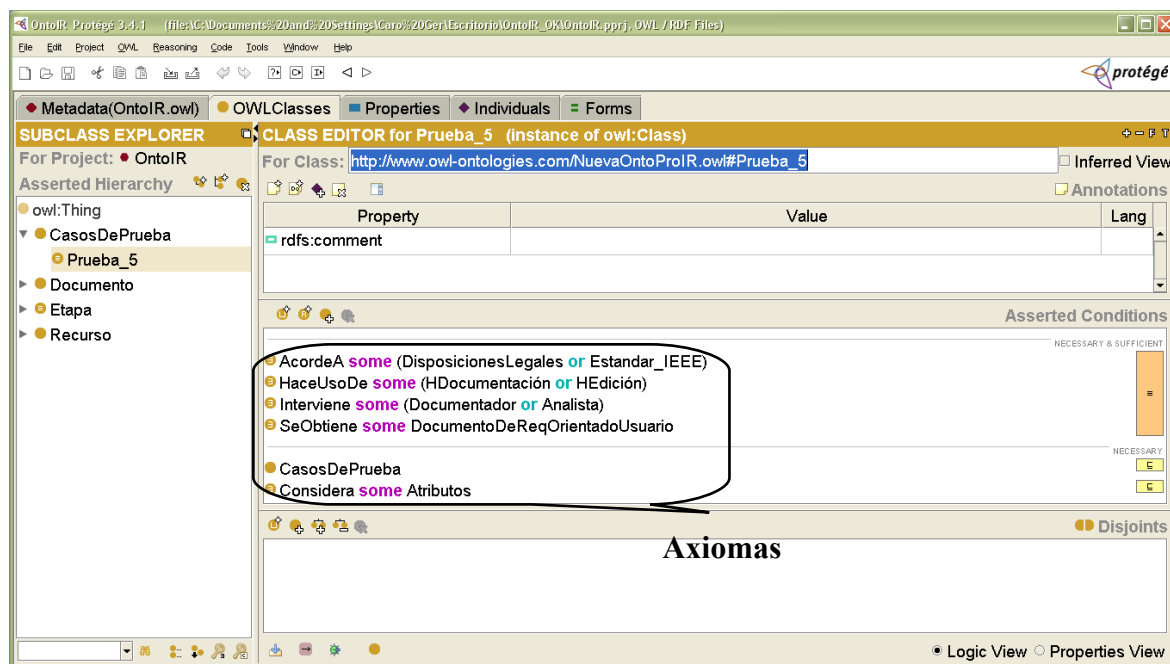


Figura A.7. Prueba\_5

En la Figura A.8 se observa el movimiento del caso de prueba\_5, luego de la ejecución de la ontología, hacia la subetapa4.3 Elegir Herramienta y Documentar.

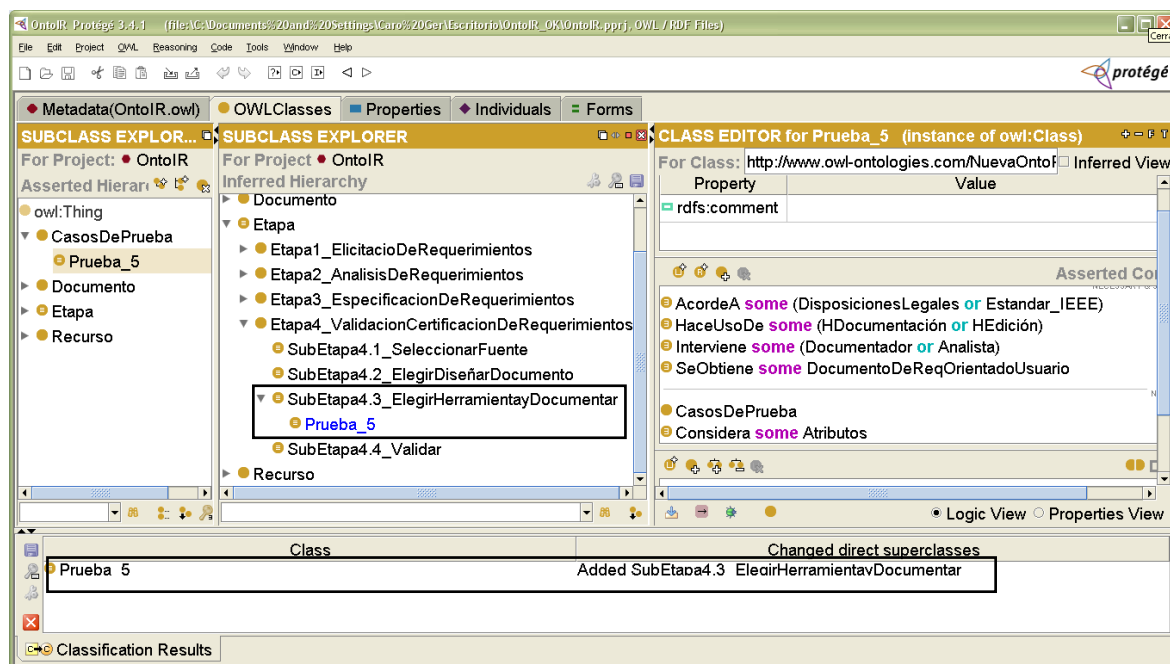


Figura A.8. Ejecución del caso de prueba\_5

La figura A.9 muestra el tiempo insumido por la ontología en dar un resultado para el presente caso.

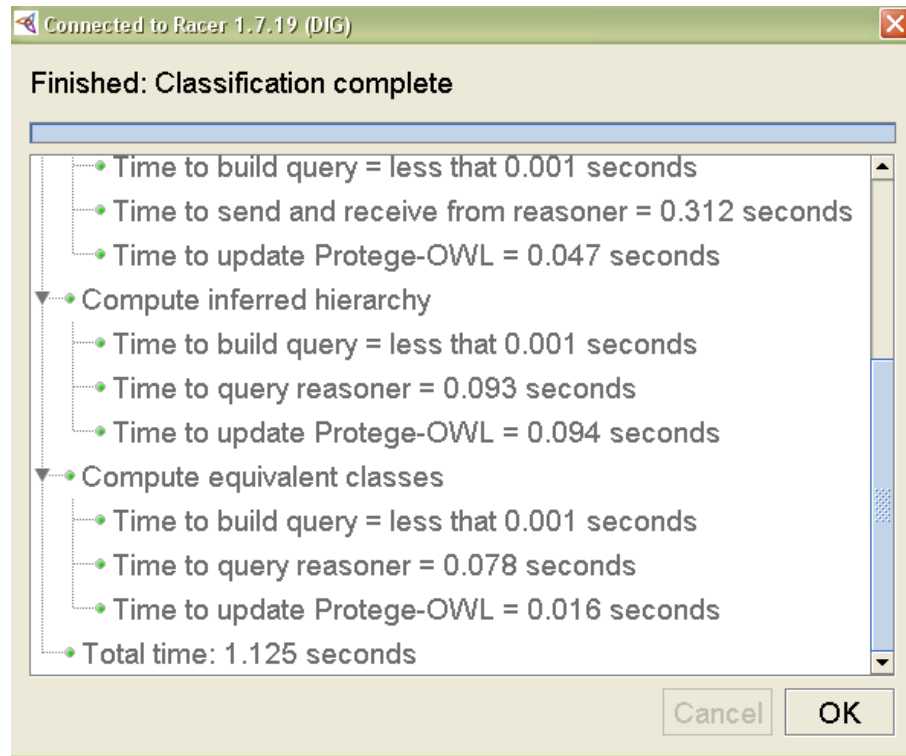


Figura A.9. Tiempo de ejecución de prueba\_5

### Caso de Prueba\_6

En la Figura A.10 se expone el caso de Prueba\_6 instanciado con los axiomas correspondientes al mismo.

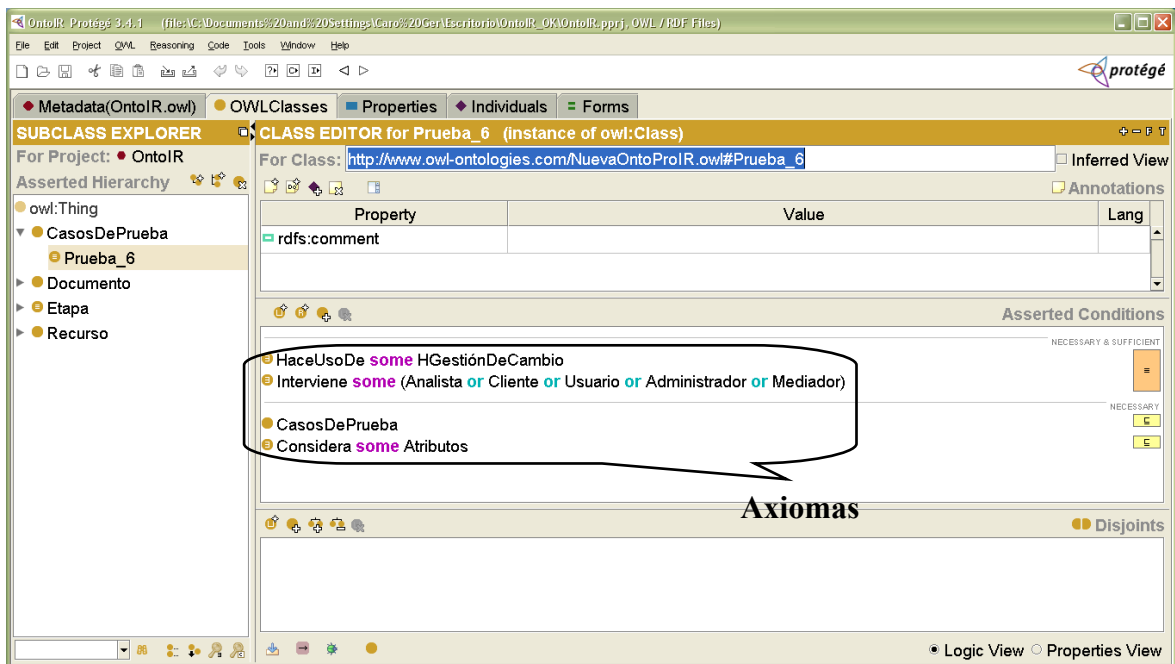


Figura A.10. Prueba\_6

En la Figura A.11 se observa el movimiento del caso de prueba\_6, luego de la ejecución de la ontología, hacia: subetapa1.3: Recolectar y Clasificar y subetapa1.4: Evaluar y racionalizar.

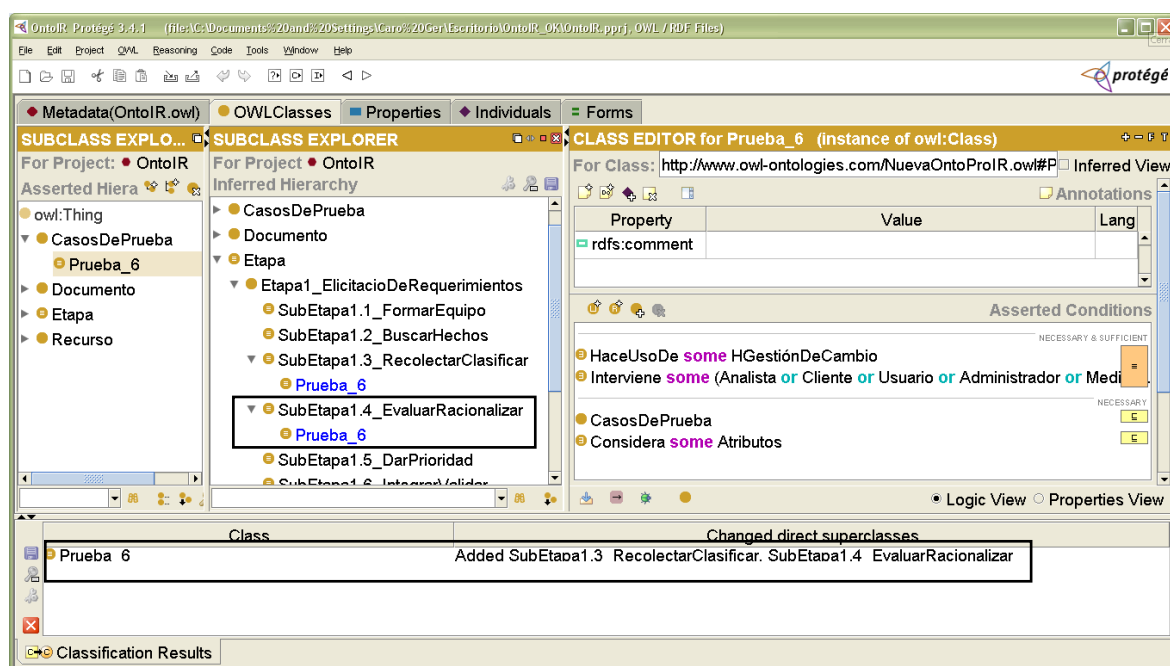


Figura A.11. Ejecución del caso de prueba\_6

La Figura A.12 muestra el tiempo insumido por la ontología en dar un resultado para el presente caso.

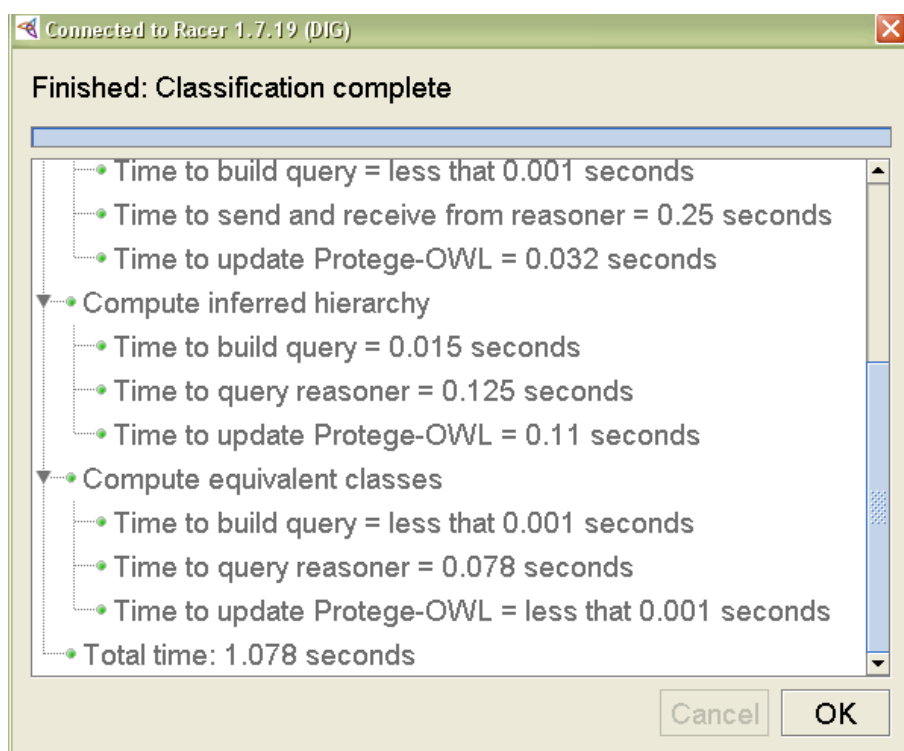


Figura A.12. Tiempo de ejecución de prueba\_6

### Caso de Prueba\_7

En la Figura A.13 se expone el caso de Prueba\_7 instanciado con los axiomas correspondientes al mismo.

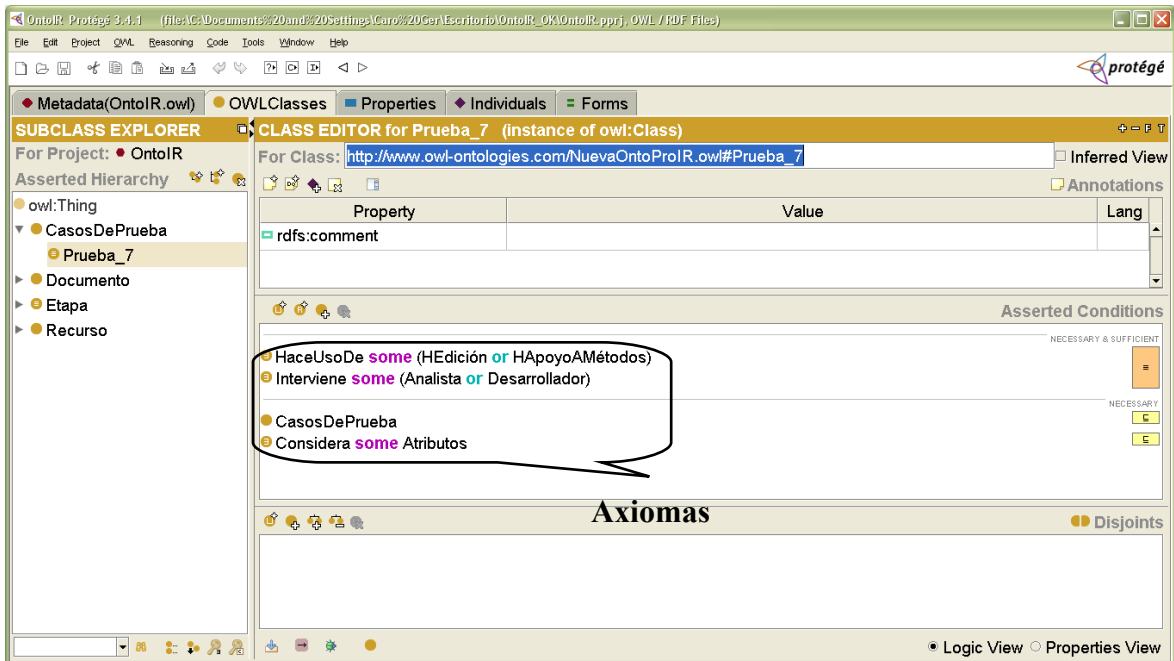


Figura A.13. Prueba\_7

En la Figura A.14 se observa el movimiento del caso de prueba\_7, luego de la ejecución de la ontología, hacia: subetapa2.1: Reducir ambigüedad, subetapa2.2: Traducir a lenguaje técnico y subetapa2.3: Modelo lógico.

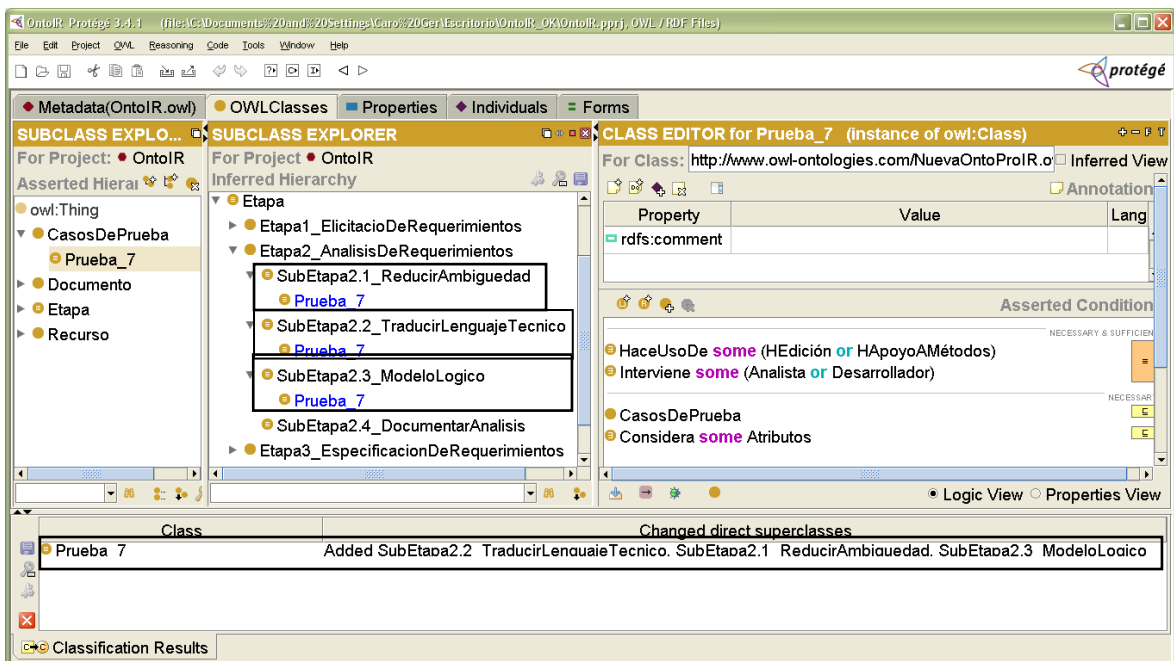


Figura A.14. Ejecución del caso de prueba\_7

La Figura A.15 muestra el tiempo insumido por la ontología en dar un resultado para el presente caso.

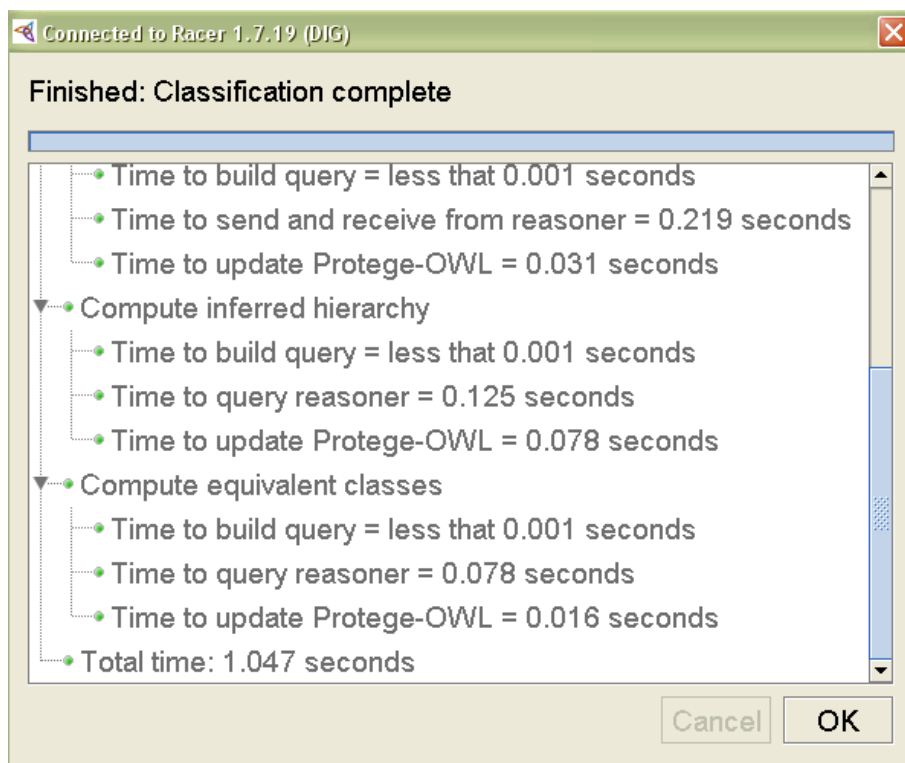


Figura A.15. Tiempo de ejecución de prueba\_7

### Caso de Prueba\_8

En la Figura A.16 se expone el caso de Prueba\_8 instanciado con los axiomas correspondientes al mismo.

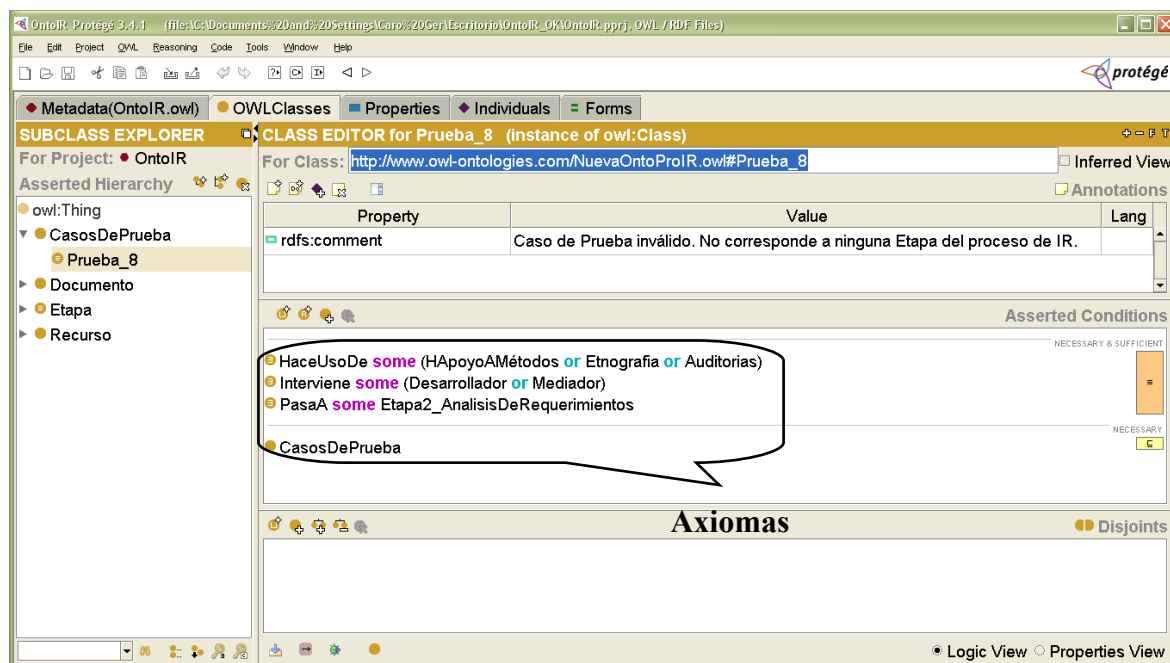


Figura A.16. Prueba\_8



En la Figura A.17 se observa que el caso de prueba\_8, no se movió hacia ninguna etapa en particular, luego de la ejecución de la ontología, debido a que se trata de un caso de prueba inválido.

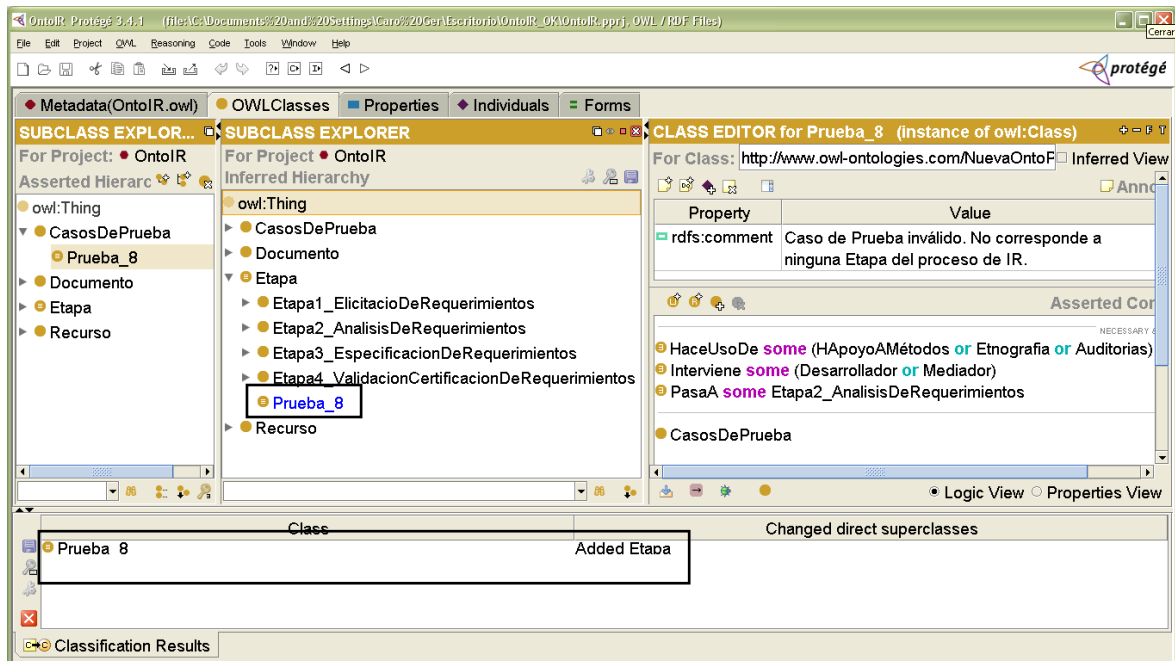


Figura A.17. Ejecución del caso de prueba\_8

La Figura A.18 muestra el tiempo insumido por la ontología en dar un resultado para el presente caso.

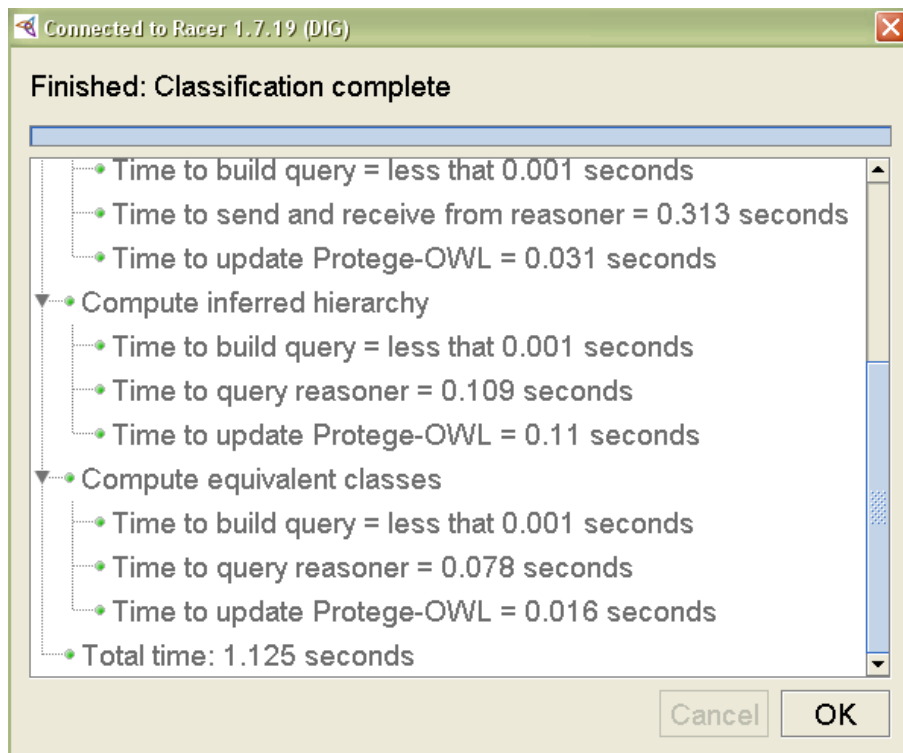


Figura A.18. Tiempo de ejecución de prueba\_8

### Caso de Prueba\_9

En la Figura A.19 se expone el caso de Prueba\_9 instanciado con los axiomas correspondientes al mismo.

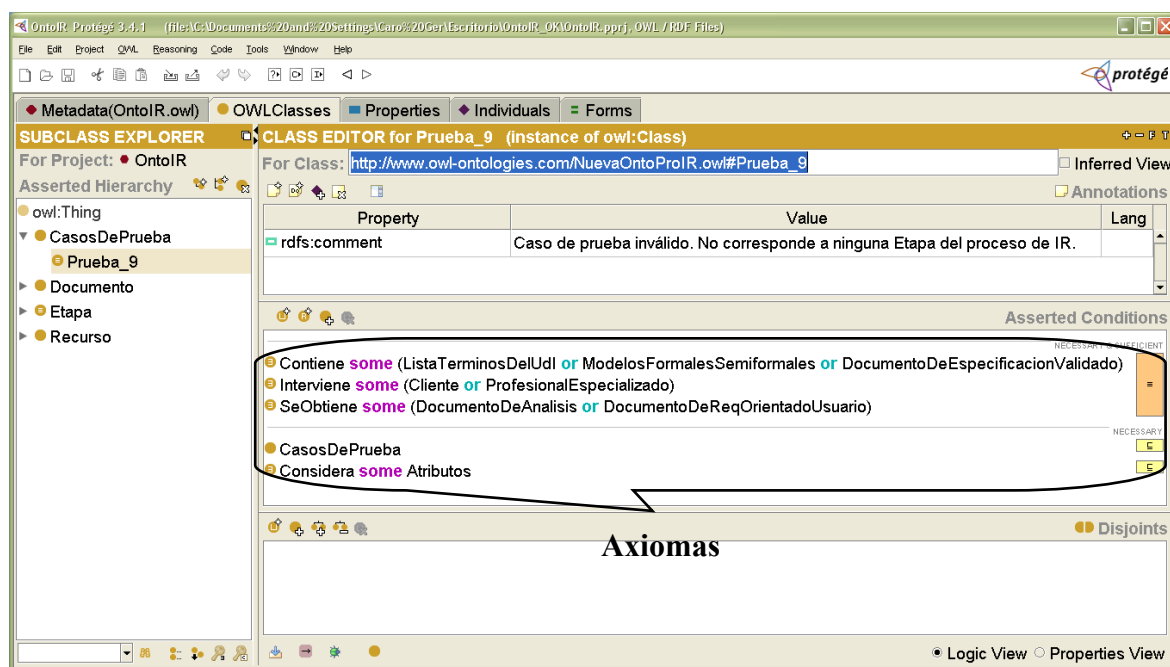


Figura A.19. Prueba\_9

En la Figura A.20 se observa que el caso de prueba\_9, no se movió hacia ninguna etapa en particular, luego de la ejecución de la ontología, debido a que se trata de un caso de prueba inválido.

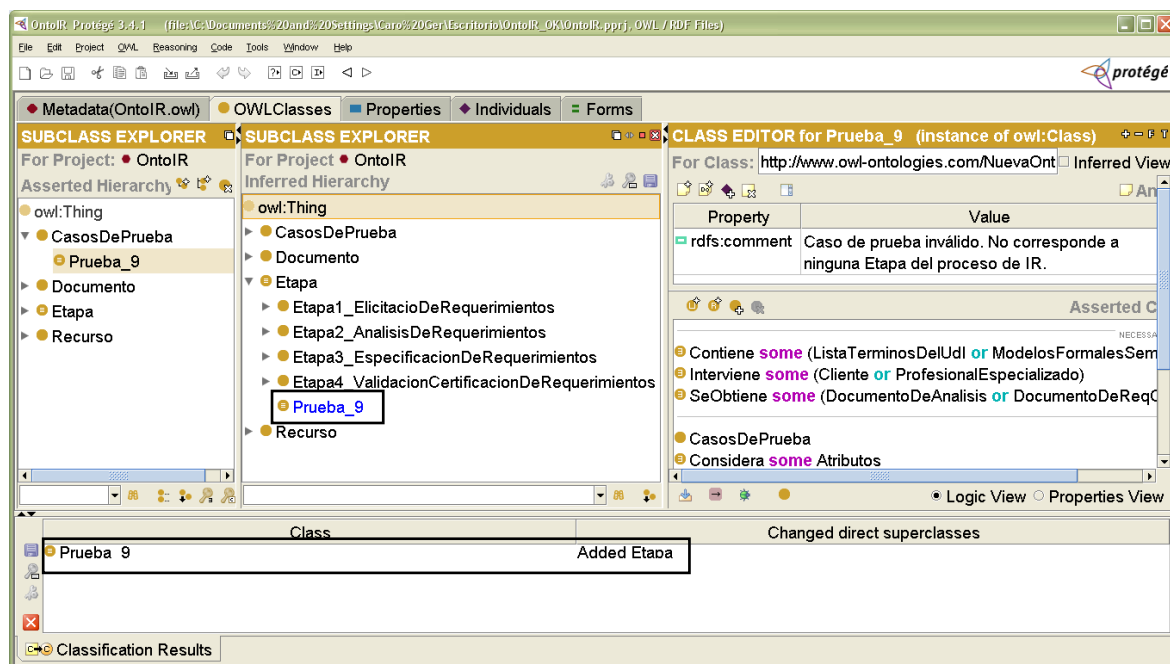


Figura A.20. Ejecución del caso de prueba\_9

La Figura A.21 muestra el tiempo insumido por la ontología en dar un resultado para el presente caso.

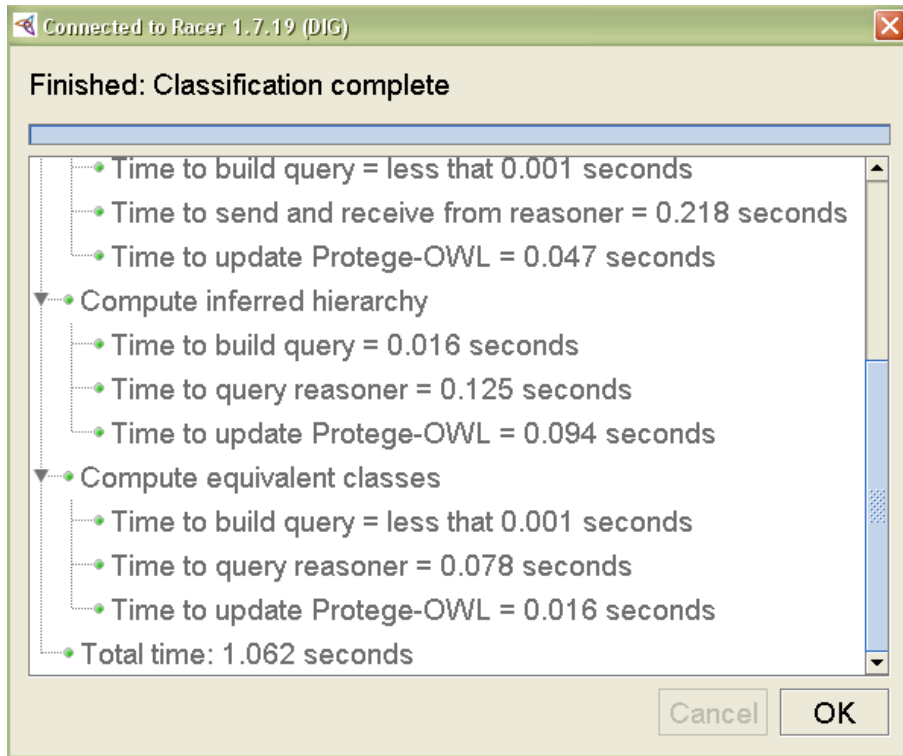


Figura A.21. Tiempo de ejecución de prueba\_9

### Caso de Prueba\_2

En la Figura A.22 se expone el caso de Prueba\_2 instanciado con los axiomas correspondientes al mismo.

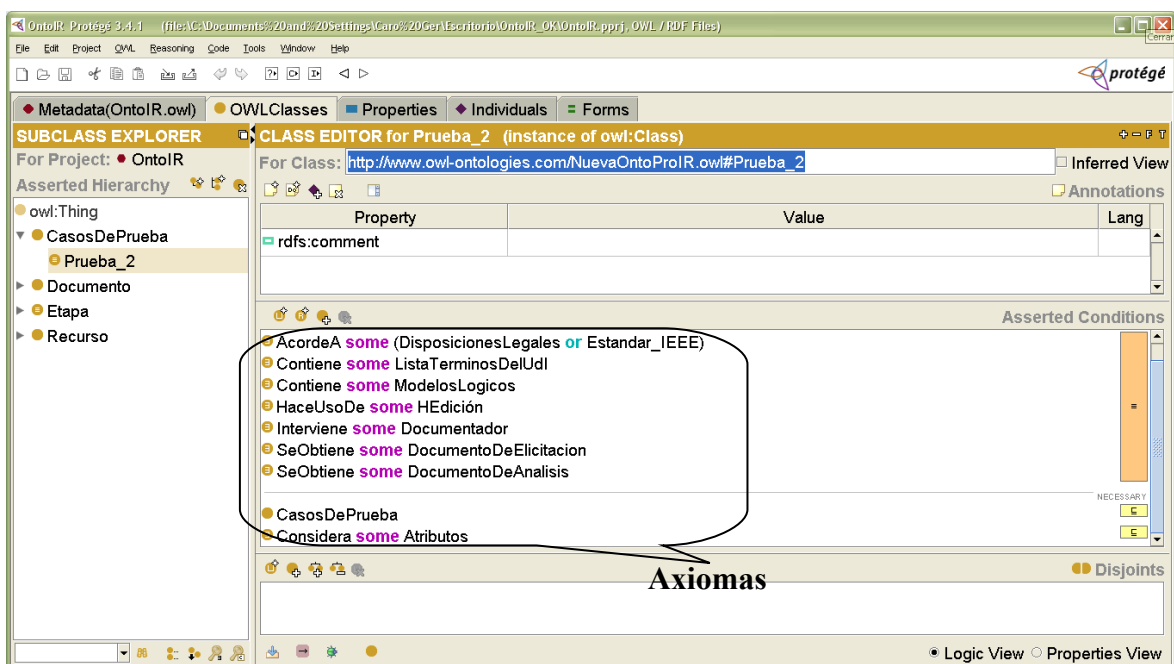


Figura A.22. Prueba\_10

En la Figura A.23 se observa el movimiento del caso de prueba\_2, después de la ejecución de la ontología, hacia la subetapa1.7: Documentar Elicitación y la subetapa2.4: Documentar Análisis.

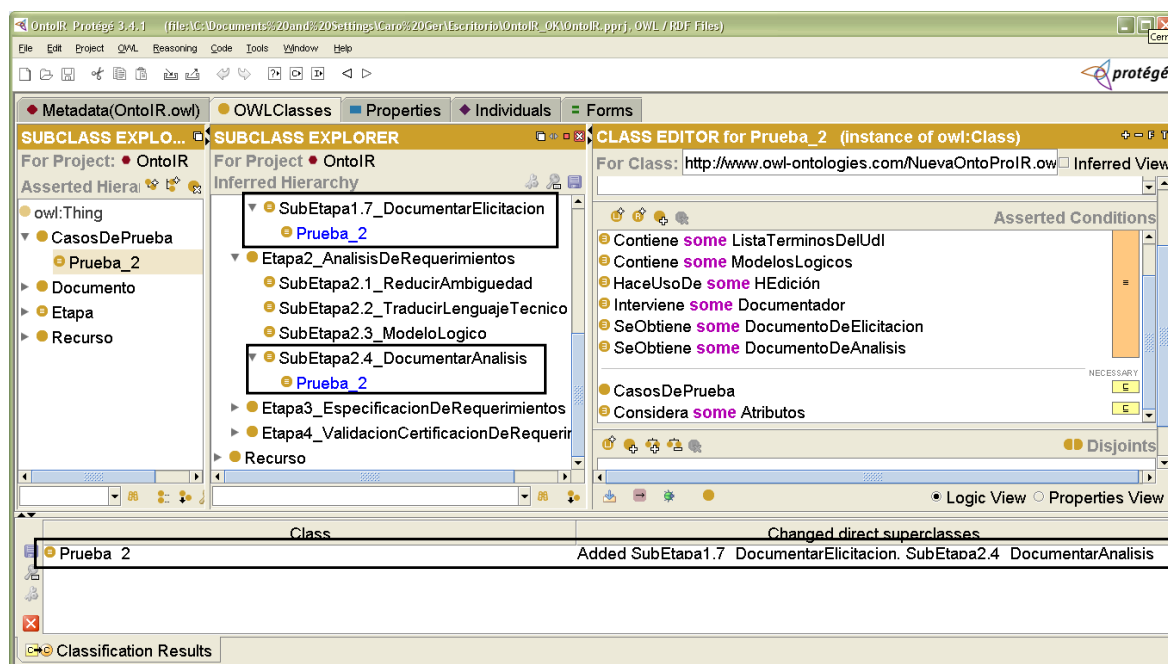


Figura A.23. Ejecución del caso de prueba\_2

La Figura A.24 muestra el tiempo insumido por la ontología en dar un resultado para el presente caso.

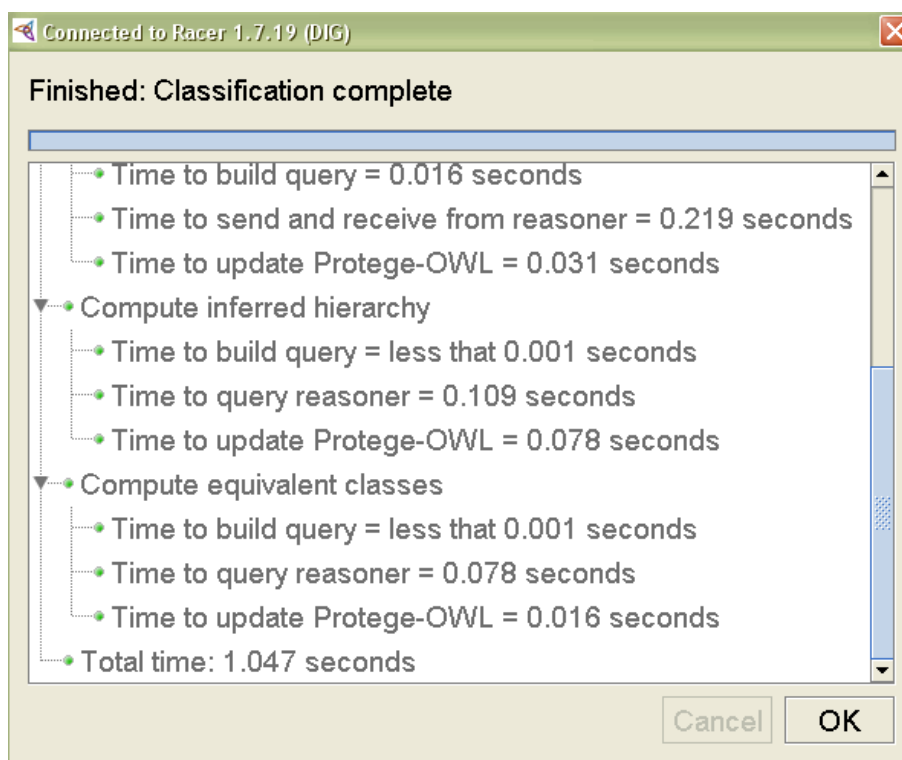


Figura A.24. Tiempo de ejecución de prueba\_2

## *Anexo B*



## B.1. SOPORTE NECESARIO PARA EJECUCION DE ONTOIR

Para poder ejecutar OntoIR es necesario:

1. Instalar **Protégé 3.4**. El instalador se lo puede obtener de: <http://protege.stanford.edu>. El instalador ofrece cuatro opciones de instalación, debe elegirse la Forma Básica + OWL o la Forma Completa, ambas son las que ofrecen los componentes necesarios para el buen funcionamiento de OntoIR.
2. Antes de ejecutar OntoIR, debe estar ejecutándose el razonador “Racer”. El mismo se anexa en la carpeta creada por la instalación del Protégé.

Una vez instalado Protégé, se ejecuta haciendo doble click sobre el ícono del mismo. Luego, en la pantalla siguiente, ir al menú *File, Open* y elegir a continuación el archivo OntoIR.owl.

Se abrirá una pantalla, como la siguiente, en la pestaña Metadata.

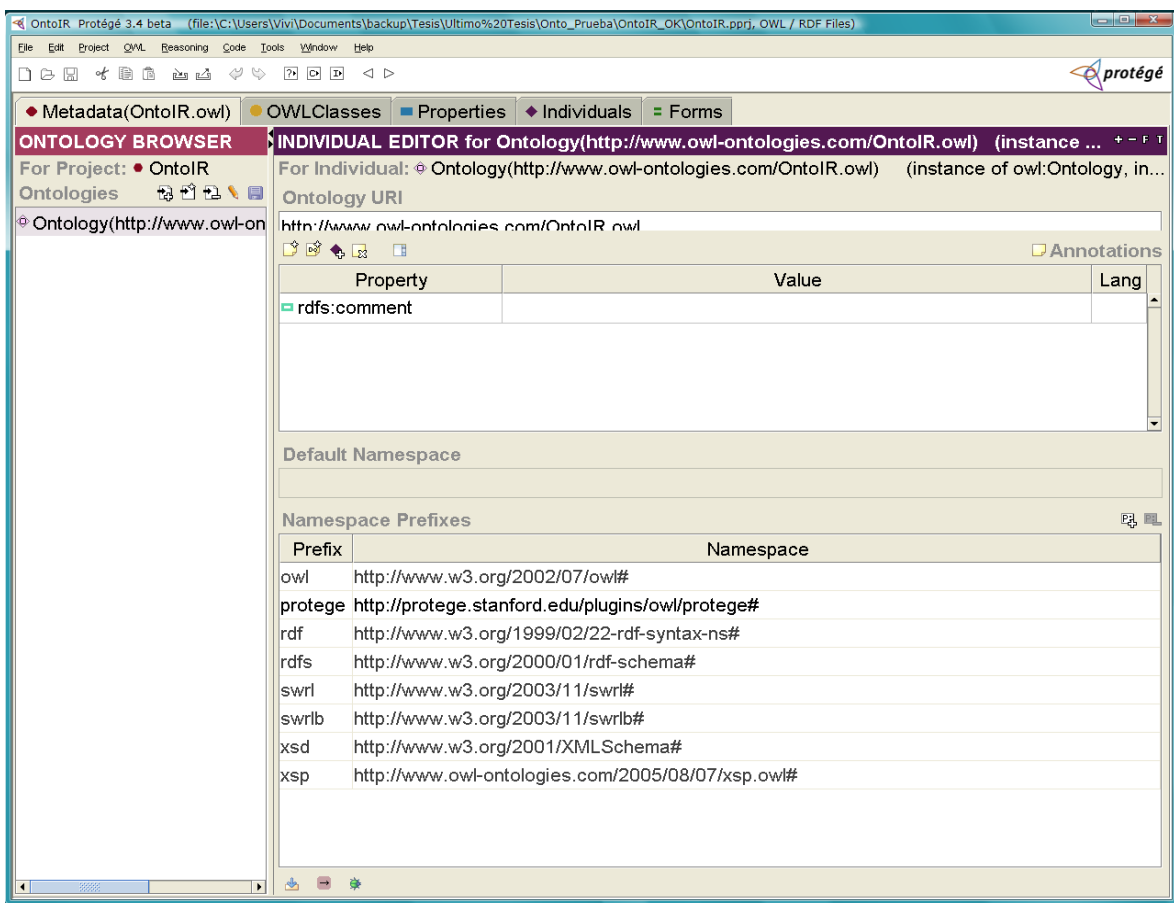


Figura B.1. Pantalla “Metadata”

En la Figura B.1, elegimos la pestaña OWL Classes, aquí se podrá ver la estructura y la información contenida por OntoIR, como se muestra en la Figura B.2

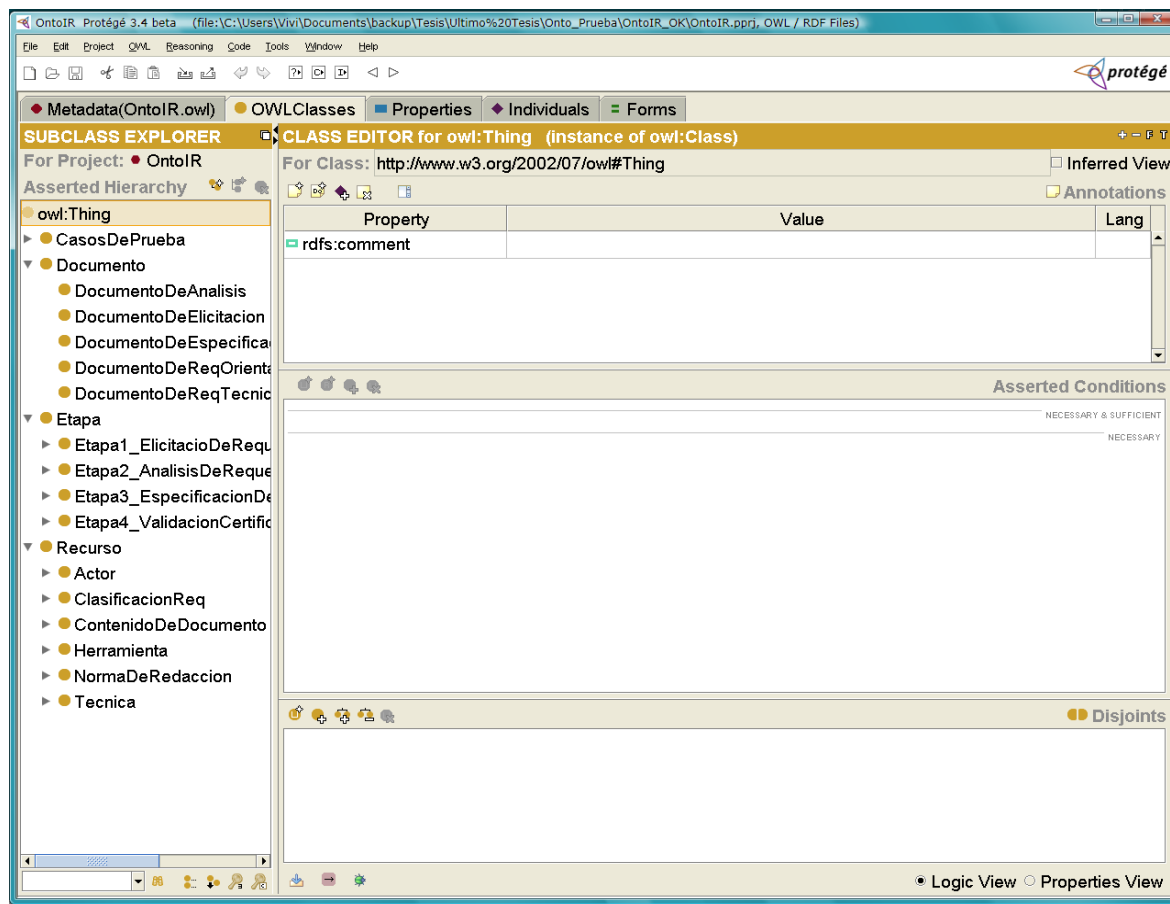


Figura B.2. Pantalla OWLClasses

## B.2. INGRESO DE CASOS DE PRUEBA FICTICIOS O REALES

Para ingresar un nuevo caso, seleccionar la clase *CasosDePrueba*, y hacer click en el ícono de crear subclase. Se crea una subclase que tiene por defecto el nombre de Class\_1, se cambia el nombre por el que se desee, por ejemplo Caso\_1. Para ingresar los axiomas correspondientes a la información que queremos averiguar, hacer click en el ícono de *Agregar Restricción*, aparecerá la pantalla para agregar las restricciones (Figura B.4). La explicación del proceso de ingreso de casos, se muestra en la Figura B.3.

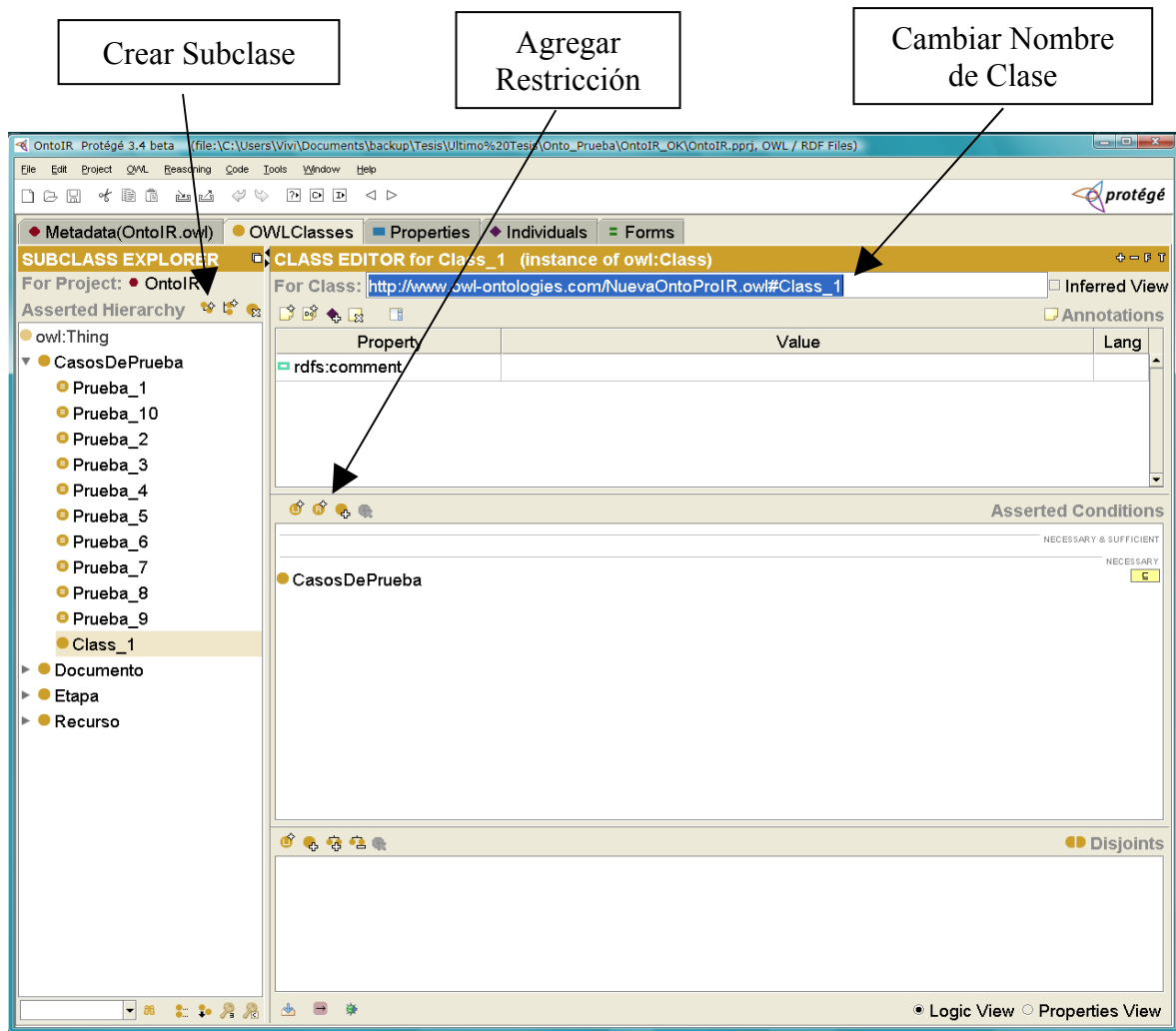


Figura B.3. Ingreso de Nuevos Casos

En la Figura B.4, se van construyendo los axiomas. Por ejemplo si se quiere agregar un axioma referente a las normas de redacción de documentos se elige la propiedad *AcordeA*, si lo que se agrega algún axioma referente al contenido de los documentos se elige *Contiene*, también se debe elegir la restricción correspondiente (en este caso se eligió  $\exists$ ), luego se hace click sobre el ícono para insertar clase. Se abrirá una ventana como la que muestra la Figura B.5, se debe elegir una norma de redacción (en el caso de que se haya elegido la propiedad *AcordeA*) de las especificadas en la clase *NormasDeRedaccion*, la Figura B.6 muestra como va quedando el axioma. Presionar OK, si el axioma es correcto o CANCEL para anular la creación del axioma. El axioma creado se mostrará como en la Figura B.7.



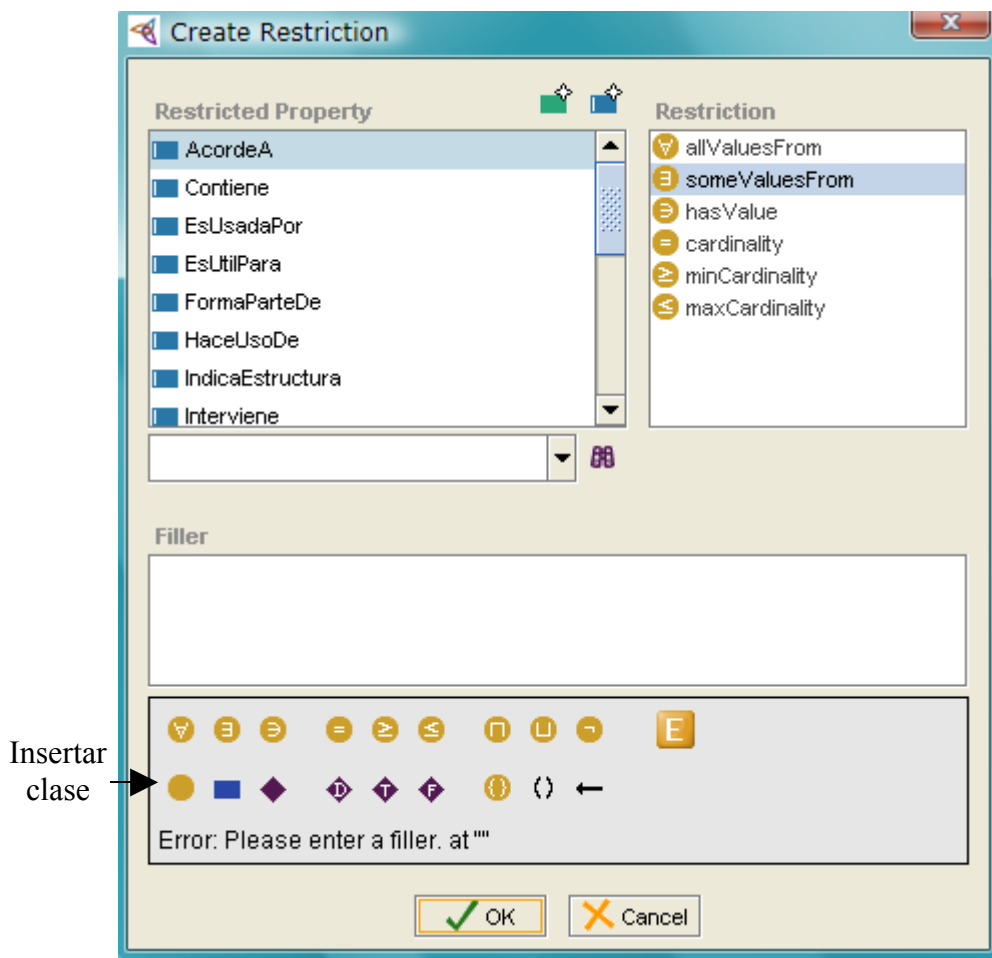


Figura B.4. Agregar Restricción

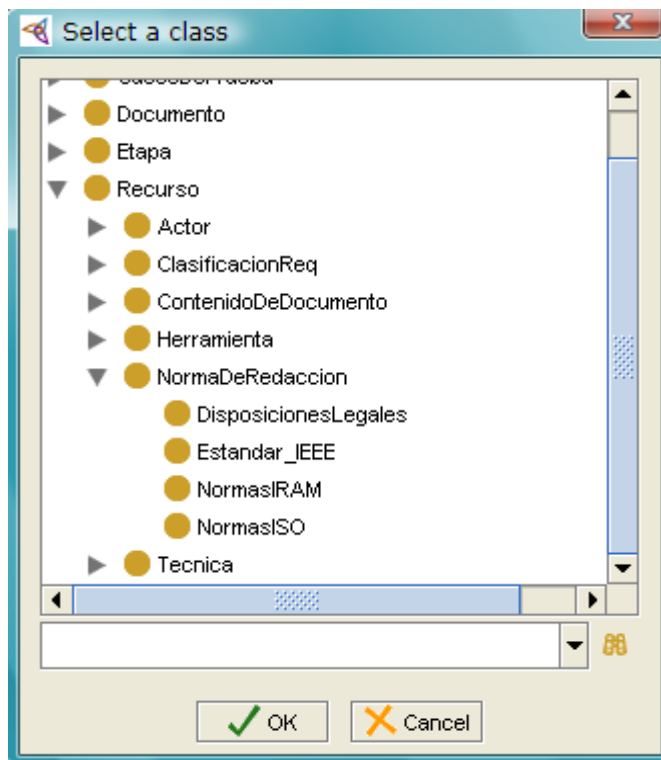


Figura B.5. Insertar Clase

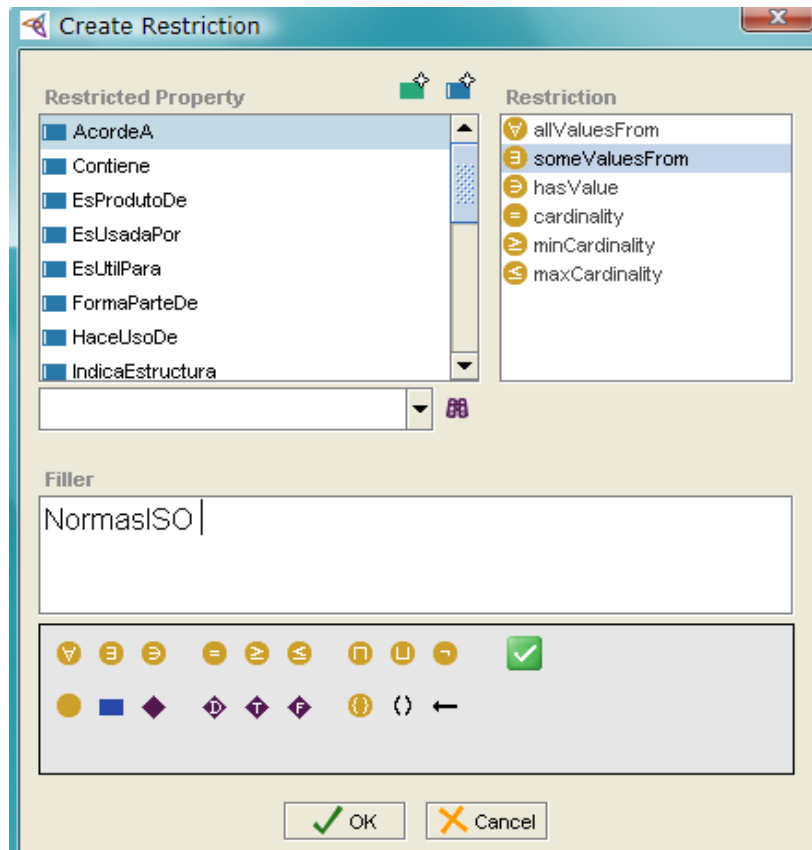


Figura B.6. Axioma creado

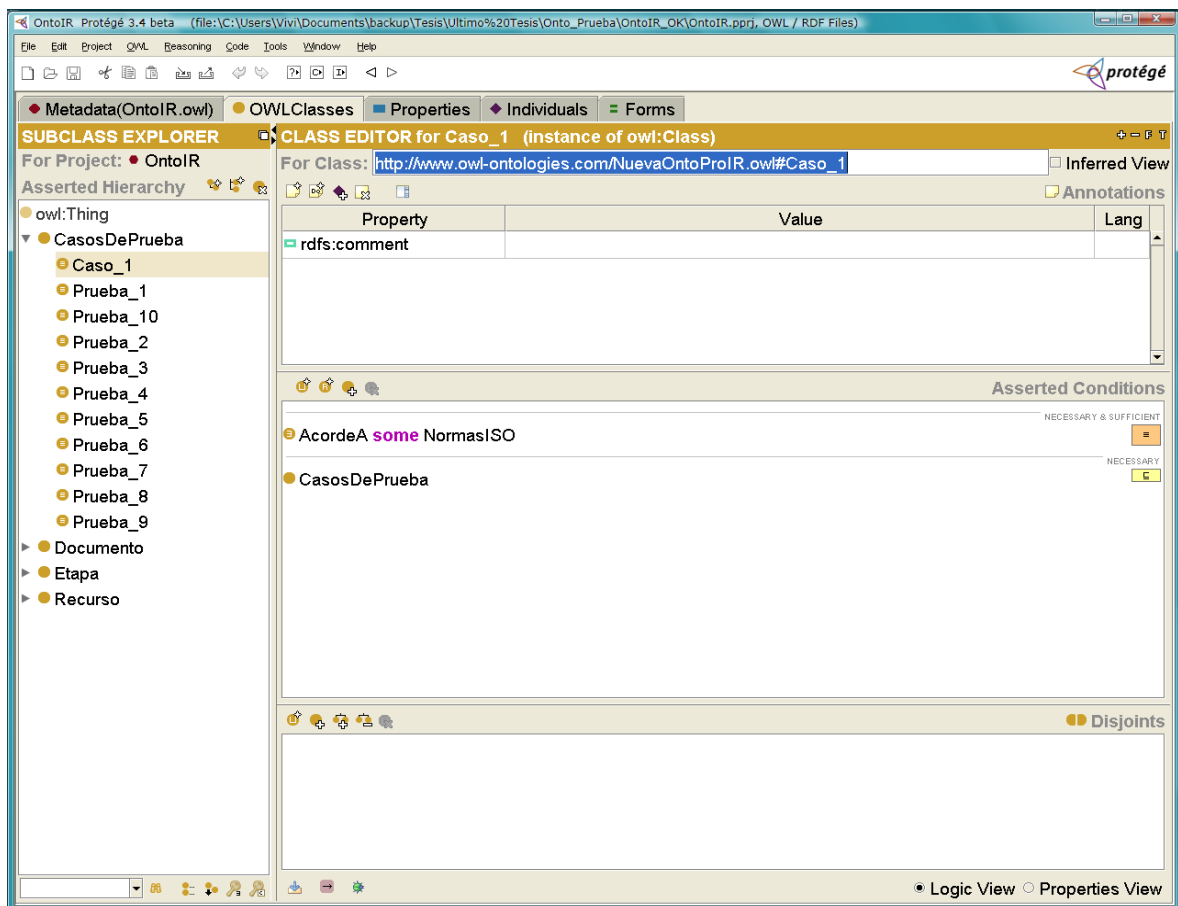


Figura B.7. Axioma creado en Caso\_1

### B.3. EXPLICACIÓN DE PROPIEDADES

A continuación, se realiza una breve explicación de cada una de las propiedades establecidas en OntoIR, esto servirá de ayuda al momento de la creación de axiomas.

Propiedades	Explicación
AcordeA	Indica las normas de redacción que se pueden seguir para crear un documento. Relaciona las clases <i>Etapa</i> y <i>NormasdeRedacción</i>
Contiene	Indica el contenido básico de los documentos que se crean en cada una de las etapas. Relaciona las clases <i>Etapa</i> y <i>ContenidoDeDocumento</i> .
HaceUsoDe	Indica para que etapa del proceso son útiles los recursos disponibles. Relaciona las clases <i>Etapa</i> y <i>Recurso</i> .
Interviene	Indica en que etapa del proceso interviene cada uno de los diferentes actores. Relaciona las clases <i>Etapa</i> y <i>Actor</i> .
PasaA	Indica cual es la etapa a la cual se debe pasar luego de finalizada otra.
VuelveA	Indica la/las etapa/s a la/s que se puede/n volver en caso de ser necesario.
SeClasificaEn	Indica la clasificación de los requerimientos. Relaciona las clases <i>Etapa</i> y <i>ClasificacionRequerimientos</i> .
SeObtiene	Indica en que etapa del proceso se obtiene cada documento. Relaciona las clases <i>Documento</i> y <i>Etapa</i> .
SeValeDe	Indica los documentos de los que se vale una etapa para realizar la validación de otros documentos.

### B.4. EJECUCIÓN DE ONTOIR

Para ejecutar OntoIR, se debe hacer click sobre el ícono *Clasificación Taxonómica* (recordar que debe estar corriendo el razonador *Racer*). A continuación se verá una pantalla como muestra la Figura B.8, la misma muestra todos los casos ingresados movidos hacia las etapas correspondientes.

En el recuadro *Subclass Explorer*, se muestran todas las etapas con los correspondientes casos de prueba ingresados.

En el recuadro *Classification Results*, se muestran todos los casos y hacia donde fueron movidos (etapas a las cuales corresponden).

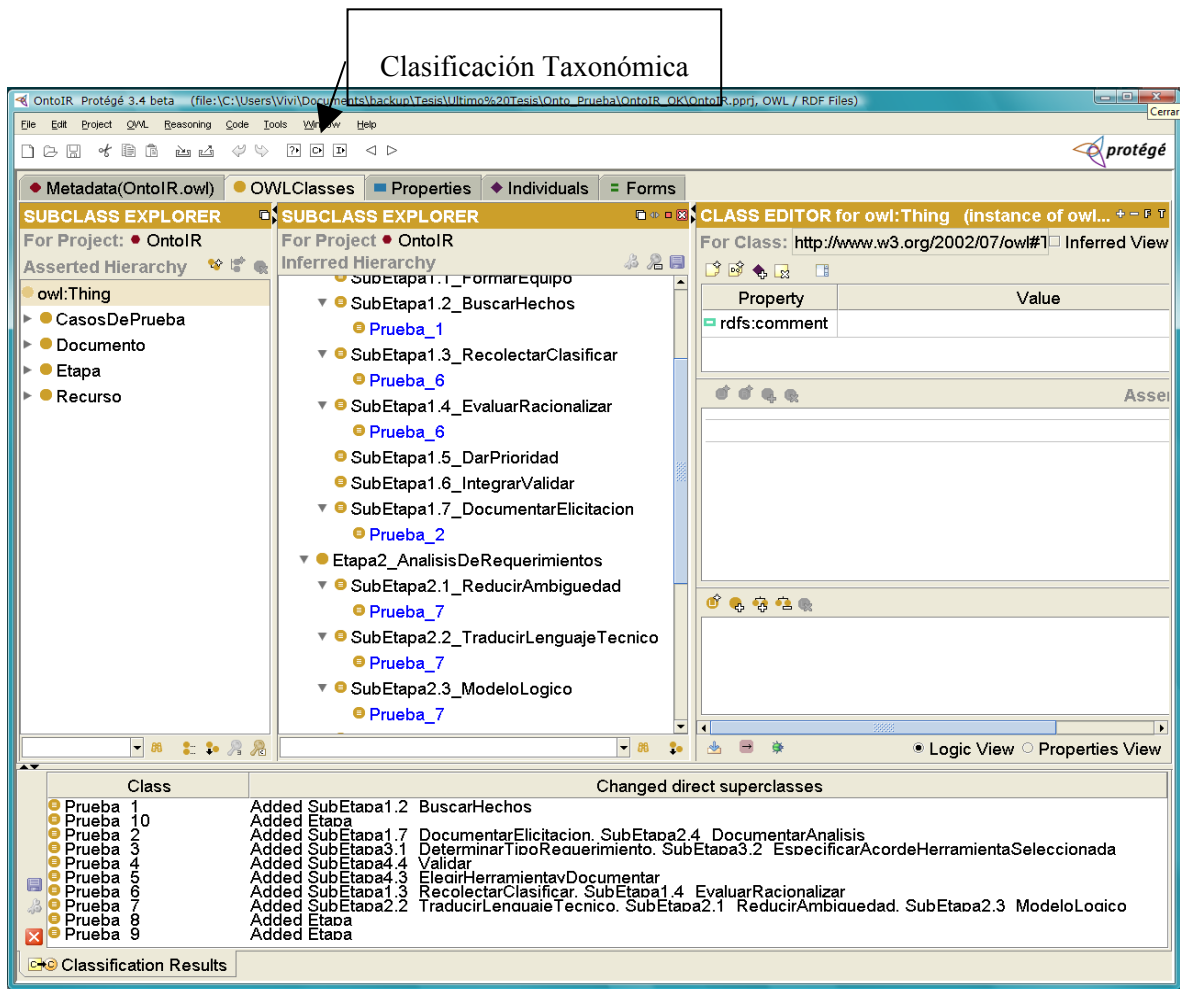


Figura B.8. Ejecución de OntoIR

# *Anexo C*



## CUESTIONARIO PARA EVALUACIÓN DE OntoIR

Lugar: SANTIAGO DEL ESTERO.-

Fecha: 10 / 03 / 2010

Nombre y Apellido: AGUSTÍN BILLAUD.-

Título: TEC. PROGRAMADOR.-

Tarea Principal: PROGRAMADOR.-

Años de Experiencia: 5.-

Responda a las siguientes preguntas por: SI / NO y justifique su respuesta	SI	NO	Justificación
1. ¿La Ontología ayuda al seguimiento del Proceso de IR?	X		REFLEJA CADA ETAPA DEL PROCESO DE I.R.-
2. ¿La Ontología refleja el funcionamiento del proceso de IR?	X		
3. ¿La Ontología aporta la información necesaria para llevar a cabo el proceso de IR?	X		BRINDA INFORMACIÓN PRECISA EN CADA UNO DE LOS PASOS.-
4. Considera que la Ontología contiene la suficiente información referente a los siguientes ítems:			LA INFORMACIÓN ES CORRECTA EN LOS DIFERENTES ÍTEMS.-
Herramientas	X		
Técnicas	X		
Actores	X		
Normas de Redacción	X		
Documentos	X		
Clasificación de Requerimientos	X		

5. ¿Surgieron inconsistencias durante la ejecución de OntoIR?		X	
6. ¿El tiempo insumido en arrojar un resultado es apropiado?	X		PRUDENTE.-
7. ¿El vocabulario utilizado por OntoIR ayuda a unificar los términos competentes a la Ingeniería de Requerimientos?	X		EL VOCABULARIO PERMITE LA UNIFICACIÓN CON LOS TÉRMINOS COMPETENTES A LA I.R.-
8. A su criterio, ¿el vocabulario de términos usado, para crear OntoIR, es adecuado?	X		COMPENSIBLE.-
9. Teniendo en cuenta los casos de prueba suministrados, ¿considera que OntoIR arroja resultados válidos?	X		
10. ¿Considera a la ontología como una herramienta que maximiza la calidad del proceso de IR?	X		ES UNA HERRAMIENTA EFICAZ PARA EL DESARROLLO DE LOS PROCESOS DE I.R.-

Observaciones: —

  
**ING. PROG. AGUSTÍN BILLAUD**  
 COORDINADOR  
 DESARROLLO DE SISTEMAS

## CUESTIONARIO PARA EVALUACIÓN DE OntoIR

**Lugar:** Santiago del Estero, Capital

**Fecha:** 10 / 03 / 2020

**Nombre y Apellido:** Beatriz Fernández

**Título:** Ingeniería en Computación

**Tarea Principal:** Desarrollo de Sistemas

**Años de Experiencia:** 4

Responda a las siguientes preguntas por: SI / NO y justifique su respuesta	SI	NO	Justificación
1. ¿La Ontología ayuda al seguimiento del Proceso de IR?	X		Porque la ontología brinda un soporte para la realización del Proceso de IR.
2. ¿La Ontología refleja el funcionamiento del proceso de IR?	X		Debido a que la Ontología contempla las diferentes etapas del Proceso.
3. ¿La Ontología aporta la información necesaria para llevar a cabo el proceso de IR?	X		ya que guía en cada una de las etapas identificando los elementos involucrados en cada una.
4. Considera que la Ontología contiene la suficiente información referente a los siguientes ítems:			Aporta todos los elementos necesarios para reducir los errores al momento de realizar el Proceso.
Herramientas	X		
Técnicas	X		
Actores	X		
Normas de Redacción	X		
Documentos	X		
Clasificación de Requerimientos	X		



5. ¿Surgieron inconsistencias durante la ejecución de OntoIR?		X	Las pruebas se realizaron correctamente
6. ¿El tiempo insumido en arrojar un resultado es apropiado?	X		Los resultados se obtienen de forma rápida.
7. ¿El vocabulario utilizado por OntoIR ayuda a unificar los términos competentes a la Ingeniería de Requerimientos?	X		Debido a que <del>se</del> brindan las definiciones <del>por</del> ayuda en cada una de las etapas.
8. A su criterio, ¿el vocabulario de términos usado, para crear OntoIR, es adecuado?	X		Porque es simple y concreto.
9. Teniendo en cuenta los casos de prueba suministrados, ¿considera que OntoIR arroja resultados válidos?	X		Son los resultados esperados.
10. ¿Considera a la ontología como una herramienta que maximiza la calidad del proceso de IR?	X		Es una herramienta que guía en cada una de las etapas, evita que se omitan datos relevantes y de esta forma obtener un resultado de calidad.

**Observaciones:**

*[Handwritten signature]*

## CUESTIONARIO PARA EVALUACIÓN DE OntoIR

**Lugar:**

**Fecha:** 09 / 03 / 2010

**Nombre y Apellido:** STELLA MARIS CORREA CAMUS

**Título:** Est. avanzada Lic. en Sist. de Inform. **Tarea Principal:** DESARROLLO DE SOFTWARE

**Años de Experiencia:** 4

Responda a las siguientes preguntas por: SI / NO y justifique su respuesta	SI	NO	Justificación
1. ¿La Ontología ayuda al seguimiento del Proceso de IR?	X		La ontología permite representar claramente las etapas del proceso de IR
2. ¿La Ontología refleja el funcionamiento del proceso de IR?	X		La ontología describe las funciones y acciones a llevar a cabo en cada etapa del proceso
3. ¿La Ontología aporta la información necesaria para llevar a cabo el proceso de IR?	X		La ontología representa una guía al trabajo del proceso IR, brindando la inf. suficiente en/etapa.
4. Considera que la Ontología contiene la suficiente información referente a los siguientes ítems:			Brinda un informe completo sobre cada uno de los ítems -
Herramientas	X		
Técnicas	X		
Actores	X		
Normas de Redacción	X		
Documentos	X		
Clasificación de Requerimientos	X		

5. ¿Surgieron inconsistencias durante la ejecución de OntoIR?		X	Los resultados se obtuvieron de manera satisfactoria.
6. ¿El tiempo insumido en arrojar un resultado es apropiado?	X		el tiempo total de ejecución fue moderado
7. ¿El vocabulario utilizado por OntoIR ayuda a unificar los términos competentes a la Ingeniería de Requerimientos?	X		Los terminos utilizados en ontoIR unifican con la terminologia de la IR
8. A su criterio, ¿el vocabulario de términos usado, para crear OntoIR, es adecuado?	X		
9. Teniendo en cuenta los casos de prueba suministrados, ¿considera que OntoIR arroja resultados válidos?	X		Los resultados obtenidos de los casos de prueba permitieron obtener resultados válidos
10. ¿Considera a la ontología como una herramienta que maximiza la calidad del proceso de IR?	X		Maximizo la calidad del proceso IR ya que constituye una herramienta util para el proceso de IR.

**Observaciones:**

## CUESTIONARIO PARA EVALUACIÓN DE OntoIR

Lugar: SANTIAGO DEL ESTERO

Fecha: 09 / 03 / 2010

Nombre y Apellido: C. VERONICA SANZELMAN BOLDAN

Título: TERCERA INFORMÁTICA

Tarea Principal: DESARROLLO SOFTWARE

Años de Experiencia: 2

Responda a las siguientes preguntas por: SI / NO y justifique su respuesta

SI

NO

Justificación

1. ¿La Ontología ayuda al seguimiento del Proceso de IR?

X

La ontología explica las etapas del proceso de IR.

2. ¿La Ontología refleja el funcionamiento del proceso de IR?

X

La ontología detalla las actividades que se realizan en las etapas del proceso

3. ¿La Ontología aporta la información necesaria para llevar a cabo el proceso de IR?

X

La ontología conduce a través de las etapas brindando la inf. necesaria durante todo el proceso

4. Considera que la Ontología contiene la suficiente información referente a los siguientes ítems:

Herramientas

X

Técnicas

X

Actores

X

Normas de Redacción

X

Documentos

X

Clasificación de Requerimientos

X

Proporciona la información indispensable sobre los ítems

5. ¿Surgieron inconsistencias durante la ejecución de OntoIR?		X	No se registraron inconsistencias durante la corrida.
6. ¿El tiempo insumido en arrojar un resultado es apropiado?	X		El tiempo insumido en la ejecución para arrojar el resultado fue rápido
7. ¿El vocabulario utilizado por OntoIR ayuda a unificar los términos competentes a la Ingeniería de Requerimientos?	X		Si ayuda a unificar términos y comprender las relaciones entre ellos
8. A su criterio, ¿el vocabulario de términos usado, para crear OntoIR, es adecuado?	X		El vocabulario de términos usado es conciso y su entendimiento no requiere esfuerzo alguno.
9. Teniendo en cuenta los casos de prueba suministrados, ¿considera que OntoIR arroja resultados válidos?	X		Los resultados obtenidos a partir de los casos de prueba suministrados son válidos
10. ¿Considera a la ontología como una herramienta que maximiza la calidad del proceso de IR?	X		Si ya se hace para ello tiene en cuenta los diferentes etapas de IR

**Observaciones:**

## CUESTIONARIO PARA EVALUACIÓN DE OntoIR

**Lugar:**

**Fecha:** 11 / 03 / 2012

**Nombre y Apellido:** Isabel Galván

**Título:** Est. avanzada de Ing. en Comp.

**Tarea Principal:** Desarrollo de software

**Años de Experiencia:** 3...

**Responda a las siguientes preguntas por: SI / NO y justifique su respuesta**

SI

NO

**Justificación**

1. ¿La Ontología ayuda al seguimiento del Proceso de IR?

X

Ya q. la ontología permite la representación clara del proc. IR

2. ¿La Ontología refleja el funcionamiento del proceso de IR?

X

Ad. detallar las etapas del proc. IR, detallando con ello también las actividades e llevar a cabo.

3. ¿La Ontología aporta la información necesaria para llevar a cabo el proceso de IR?

X

La ontología brinda en cada etapa los detalles e inf. necesaria. para llevar el proc. IR. a cabo correctamente.

4. Considera que la Ontología contiene la suficiente información referente a los siguientes ítems:

Herramientas

X

Técnicas

X

Actores

X

Normas de Redacción

X

Documentos

X

Clasificación de Requerimientos

X

Así es, la ontología proporciona la información necesaria de cada uno de los ítems p. se debe. con a fin de llevar a cabo un proceso correcto de desarrollo de soft.

5. ¿Surgieron inconsistencias durante la ejecución de OntoIR?		X	No se presentaron inconsistencias durante su ejecución.
6. ¿El tiempo insumido en arrojar un resultado es apropiado?	X		El tiempo de ejecución fue rápido.
7. ¿El vocabulario utilizado por OntoIR ayuda a unificar los términos competentes a la Ingeniería de Requerimientos?	X		Ya q' OntoIR no solamente ayuda a entender el concepto sino q' además las relaciones entre los términos.
8. A su criterio, ¿el vocabulario de términos usado, para crear OntoIR, es adecuado?	X		Sí ya q' es claro.
9. Teniendo en cuenta los casos de prueba suministrados, ¿considera que OntoIR arroja resultados válidos?	X		Los resultados q' se obtienen son los esperados, resultados válidos.
10. ¿Considera a la ontología como una herramienta que maximiza la calidad del proceso de IR?	X		Absolutamente, ya q' por tener en cuenta los etapas del proc IR y brindar info de sus etapas contribuye a la realización de un soft. de eficiente.

Observaciones:

## CUESTIONARIO PARA EVALUACIÓN DE OntoIR

**Lugar:** Santiago del Estero, Capital

**Fecha:** 12/03/2010

**Nombre y Apellido:** María Silvana Álvarez

**Título:** Licenciada en Sistemas

**Tarea Principal:** Desarrollo software

**Años de Experiencia:** 8 años

**Responda a las siguientes preguntas por: SI / NO y justifique su respuesta**

SI

NO

Justificación

1. ¿La Ontología ayuda al seguimiento del Proceso de IR?

x

Da soporte a las diferentes etapas del proceso del IR.

2. ¿La Ontología refleja el funcionamiento del proceso de IR?

x

Si, porque utiliza conceptos básicos del proceso

3. ¿La Ontología aporta la información necesaria para llevar a cabo el proceso de IR?

x

Si, porque abarca todo los componentes necesarios

4. Considera que la Ontología contiene la suficiente información referente a los siguientes ítems:

Herramientas

x

Técnicas

x

Actores

x

Normas de Redacción

x

Documentos

x

Clasificación de Requerimientos

x

La Ontología contiene los fundamentos suficientes para abordar todos los ítems mencionados.



5. ¿Surgieron inconsistencias durante la ejecución de OntoIR?		X	No se presentaron inconsistencias durante la ejecución
6. ¿El tiempo insumido en arrojar un resultado es apropiado?	X		Si, el tiempo insumido fue apropiado de acuerdo a las expectativas.
7. ¿El vocabulario utilizado por OntoIR ayuda a unificar los términos competentes a la Ingeniería de Requerimientos?	X		Los componentes y los términos pueden relacionarse por su similitud y obtener unificación
8. A su criterio, ¿el vocabulario de términos usado, para crear OntoIR, es adecuado?	X		Si, ya que no requiere demasiado esfuerzo para su comprensión.
9. Teniendo en cuenta los casos de prueba suministrados, ¿considera que OntoIR arroja resultados válidos?	X		Si, los resultados coinciden con lo esperado
10. ¿Considera a la ontología como una herramienta que maximiza la calidad del proceso de IR?	X		Si, todas las características de la ontología permite maximizar la calidad del proceso de IR

**Observaciones:**

## CUESTIONARIO PARA EVALUACIÓN DE OntoIR

**Lugar:** Santiago del Estero, Capital

**Fecha:** 09 / 03 / 2010

**Nombre y Apellido:** Pacheco Toledo, Ricardo Daniel

**Título:** Estudiante - Avanzado - LSI - UNSE **Tarea Principal:** Desarrollo de sistemas software **Años de Experiencia:** 3

Responda a las siguientes preguntas por: SI / NO y justifique su respuesta	SI	NO	Justificación
1. ¿La Ontología ayuda al seguimiento del Proceso de IR?	X		La ontología es un soporte más, ya que contempla diferentes etapas de la IR.
2. ¿La Ontología refleja el funcionamiento del proceso de IR?	X		Al contemplar las etapas de la IR, la ontología representa a las secuencias de actividades que se llevan a cabo en la IR, pero que refleja el funcionamiento.
3. ¿La Ontología aporta la información necesaria para llevar a cabo el proceso de IR?	X		La ontología, no solo describe las diferentes etapas del IR, sino también provee de la información en cada etapa. Indicando como se debe realizar la misma.
4. Considera que la Ontología contiene la suficiente información referente a los siguientes ítems:			La ontología presenta la información necesaria y justa para poder realizar una captura de requisitos, que nos permita comprender el dominio del problema y reducir el riesgo de cambios en las siguientes etapas del proceso de desarrollo del software.
Herramientas	X		
Técnicas	X		
Actores	X		
Normas de Redacción	X		
Documentos	X		
Clasificación de Requerimientos	X		

5. ¿Surgieron inconsistencias durante la ejecución de OntoIR?		X	Los casos de pruebas fueron ejecutados correctamente. No presentaron inconsistencias durante la ejecución.
6. ¿El tiempo insumido en arrojar un resultado es apropiado?	X		El tiempo de ejecución fue rápido, con un tiempo de ejecución máximo de 2 minutos.
7. ¿El vocabulario utilizado por OntoIR ayuda a unificar los términos competentes a la Ingeniería de Requerimientos?	X		Porque OntoIR, no solo permite entender concepto sino también relaciones entre términos los cuales pueden llegar a unificarse por su similitud.
8. A su criterio, ¿el vocabulario de términos usado, para crear OntoIR, es adecuado?	X		Porque es simple exacto y no demanda demasiado esfuerzo y complicaciones.
9. Teniendo en cuenta los casos de prueba suministrados, ¿considera que OntoIR arroja resultados válidos?	X		Los resultados coinciden con los <del>que</del> esperados, entonces estos resultados son válidos.
10. ¿Considera a la ontología como una herramienta que maximiza la calidad del proceso de IR?	X		Porque es una herramienta más que se tiene durante el proceso de desarrollo del software. Al tener en cuenta las diferentes etapas del IR y suministrar información de las mismas contribuye a la producción de un software de calidad.

de un software de calidad.

**Observaciones:**



Pacheco Toledo, Ricardo.

## CUESTIONARIO PARA EVALUACIÓN DE OntoIR

**Lugar:** SANTIAGO DEL ESTERO

**Fecha:** 08/03/2010

**Nombre y Apellido:** GUILLERMO ANTONIO ARANDA

**Título:** INGENIERO EN COMPUTACION


**Tarea Principal:** DESARROLLADOR DE SOFTWARE

**Años de Experiencia:** 10

Responda a las siguientes preguntas por: SI/NO y justifique su respuesta	SI	NO	Justificación
1. ¿La Ontología ayuda al seguimiento del Proceso de IR?	X		DESCRIBE CON CLARIDAD LAS ETAPAS Y SUBETAPAS DEL PROCESO DE IR.
2. ¿La Ontología refleja el funcionamiento del proceso de IR?	X		REFLEJA LA SECUENCIALIDAD DE LOS PASOS Y ACCIONES A LLEVAR EN CADA UNA DE ELLAS.
3. ¿La Ontología aporta la información necesaria para llevar a cabo el proceso de IR?	X		LA ONTOLOGIA ES UNA GUIA PRACTICA QUE NOS DESCRIBE LOS PASOS Y DEMAS INFORMACION NECESARIA REQUERIDA EN EL PROCESO IR.
4. Considera que la Ontología contiene la suficiente información referente a los siguientes ítems:			CONTIENE TODA LA INFORMACION NECESARIA SOBRE LOS DIFERENTES RECURSOS QUE SE CUENTA EN LA IR.
Herramientas	X		
Técnicas	X		
Actores	X		
Normas de Redacción	X		
Documentos	X		
Clasificación de Requerimientos	X		

5. ¿Surgieron inconsistencias durante la ejecución de OntoIR?		X	NO SURGIERON INCONSISTENCIA DURANTE LA EJECUCION ONTOIR DADO QUE LOS RESULTADOS DE LA TOTALIDAD DE LOS CASOS DE PRUEBA FUERON VALIDOS.
6. ¿El tiempo insumido en arrojar un resultado es apropiado?	X		TODOS LOS TIEMPOS DE EJECUCION EN LAS PRUEBAS FUERON DENTRO DEL RANGO DE 1,047" y 1,125".
7. ¿El vocabulario utilizado por OntoIR ayuda a unificar los términos competentes a la Ingeniería de Requerimientos?	X		GRACIAS A SU PRECISO VOCABULARIO CON EL QUE CUENTA ONTOIR, EL DESARROLLADOR UNIFICA LA TERMINOLOGIA QUE NECESITA EN EL PROCESO
8. A su criterio, ¿el vocabulario de términos usado, para crear OntoIR, es adecuado?	X		EL VOCABULARIO USADO ES CLARO, COMPLETO, FACIL DE COMPRENDER Y EXACTO.
9. Teniendo en cuenta los casos de prueba suministrados, ¿considera que OntoIR arroja resultados válidos?	X		LUEGO DE EJECUTADAS LAS PRUEBAS SE OBSERVO QUE LOS RESULTADOS OBTENIDOS COINCIDEN CON LOS RESULTADOS OBTENIDOS.
10. ¿Considera a la ontología como una herramienta que maximiza la calidad del proceso de IR?	X		ES UNA HERRAMIENTA UTIL Y NECESARIA ENRIQUECE LOS CONOCIMIENTOS DEL DESARROLLO RESPECTO AL PROCESO.

Observaciones:

  
 ING. GUILLERMO ARANDA  
 COORDINADOR  
 AREA ASISTENCIA TECNICA Y CAPACITACION  
 DE USUARIOS - PODER JUDICIAL