



UNIVERSIDAD NACIONAL DE COLOMBIA

UN MODELO DE PROCESAMIENTO TERMINOLÓGICO PARA LA OBTENCIÓN Y VALIDACIÓN DE REQUISITOS DE SOFTWARE BASADO EN EL DIAGRAMA DE OBJETIVOS DE KAOS

Luis Alfonso Lezcano Rodríguez

Universidad Nacional de Colombia

Facultad de Minas

Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

Año 2014

UN MODELO DE PROCESAMIENTO TERMINOLÓGICO PARA LA OBTENCIÓN Y VALIDACIÓN DE REQUISITOS DE SOFTWARE BASADO EN EL DIAGRAMA DE OBJETIVOS DE KAOS

Luis Alfonso Lezcano Rodríguez

Tesis presentada como requisito parcial para optar al título de: Doctor en Ingeniería
Área: Sistemas e Informática

Director:

Ph.D., Jaime Alberto Guzmán Luna

Línea de Investigación: Ingeniería de Software y Procesamiento de Lenguaje Natural

Grupo de Investigación: SINTELWEB (Sistemas inteligentes Web)

Universidad Nacional de Colombia

Facultad de Minas

Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

Año 2014

Dedicatoria

A mi querida esposa Paula Andrea y a mis amados hijos Samuel y Santiago, por la alegría, la comprensión y el apoyo que me ofrecen cada día.

Luis Alfonso L. R.

Agradecimientos

A mi director Jaime Alberto Guzmán Luna por todo su apoyo, paciencia, comprensión, respeto e interés que demostró durante el desarrollo de mi formación doctoral.

Agradezco de manera especial a los profesores Jaime Polanía Vorenberg, Amanda Lucía Mora Martínez, Román Eduardo Castañeda Sepúlveda, María Cecilia Plested Álvarez y a los ingenieros Sebastián Alonso Gómez Arias, Carlos Andrés Vélez Carvajal y Carlos Alberto Álvarez Canchila por todo el apoyo que me ofrecieron durante el desarrollo de este proyecto doctoral.

Asimismo, agradezco a la Dirección de Investigaciones de la Universidad Nacional de Colombia Sede Medellín y a la Vicerrectoría de Investigaciones de la Universidad Nacional de Colombia por la financiación otorgada a los proyectos de investigación: (i) Un modelo de procesamiento terminológico para la obtención de requisitos de software basado en el diagrama de objetivos de KAOS. Código Hermes 15534. Financiado a través de la “Convocatoria DIME 2012 - financiación de proyectos de investigación modalidad 1”, y (ii) Modelo de procesamiento terminológico basado en ontologías para la desambiguación verbal en la educación de requisitos de software. Código Hermes 18717. Financiado a través de la “Convocatoria del programa nacional de proyectos para el fortalecimiento de la investigación, la creación y la innovación en posgrados de la Universidad Nacional de Colombia 2013-2015”, los cuales fueron formulados en el marco de esta tesis doctoral.

Resumen

La educación y validación de requisitos de software son actividades propias de la Ingeniería de Software, éstas se realizan en las primeras fases del ciclo de vida del software, es decir, en la definición y el análisis respectivamente. En estas fases se intentan capturar los requisitos que debe cumplir el sistema (software) y transformarlos en modelos conceptuales que permitan explicar los elementos significativos del área del problema. Sin embargo, los productos obtenidos a través de estas fases presentan problemas en su concepción debido a: (i) la brecha de comunicación presente entre el interesado (usuario) y el analista, es decir, debido a la comunicación limitada o insuficiente que se genera por las diferencias entre sus especialidades, y (ii) la ambigüedad verbal presente en el universo del discurso. Estos problemas, impiden una correcta identificación y validación de los requisitos que debe cumplir un sistema (software). Asimismo, hacen que el producto final sea más costoso debido a la corrección de errores en las etapas de implementación y producción del sistema (software).

Con el propósito de: (i) mejorar los resultados en la educación y validación de requisitos de software, y (ii) reducir la brecha de comunicación entre el interesado y el analista, se utilizan diversos esquemas conceptuales, entre éstos se emplea el diagrama de objetivos de KAOS (Knowledge Acquisition in Automated Specification), el cual le permite al usuario la identificación y la validación de los requisitos asociados al sistema (software). Además, a partir de este diagrama se pueden obtener diagramas UML propios de la tercera fase (diseño) del ciclo de vida del software, los cuales hacen parte del dominio del analista. Sin embargo, en los diferentes trabajos que especifican o utilizan el diagrama de diagrama de objetivos de KAOS, se logran identificar los siguientes problemas: (i) falta de consistencia entre el lenguaje natural y los elementos básicos que conforman el diagrama de objetivos de KAOS, (ii) no utilizan herramientas terminológicas que permitan minimizar la ambigüedad semántica presente en el universo del discurso, (iii) falta de validación semiautomática de requisitos de software por parte del interesado en la etapa inicial, y (iv) no obtienen de manera semiautomática el diagrama de objetivos correspondiente.

Con la intención de dar solución a los problemas identificados en el párrafo anterior, se logró realizar en esta tesis doctoral lo siguiente: (i) caracterizar los elementos que componen el diagrama de objetivos de KAOS, (ii) construir una Ontología para representar con mayor precisión los elementos del diagrama de objetivos KAOS, (iii) definir un conjunto de formas morfosintácticas que permiten caracterizar a partir de lenguaje natural los elementos básicos (objetivo, entidad, operación y agente) del diagrama de objetivos de KAOS, de tal forma, que se logró dar una solución a la necesidad de identificar la consistencia que debe existir entre el lenguaje natural y los elementos básicos que conforman diagrama de objetivos de KAOS, (iv) definir un conjunto de reglas semánticas que permiten minimizar la ambigüedad semántica presente en el universo del discurso, la cual afecta de forma negativa las actividades asociadas a la educación y validación de

requisitos de software, (v) crear un modelo de procesamiento terminológico para la educación y validación interactiva con el usuario de requisitos de software inicialmente descritos en lenguaje natural, (vi) desarrollar la aplicación NL2KAOS (elaborada en JAVA, JSP y PHP) que permite automatizar el modelo propuesto, además, incluye las formas morfosintácticas y las reglas semánticas definidas en esta tesis, (vii) obtener de manera semiautomática el diagrama de objetivos de KAOS, (viii) realizar la evaluación del modelo mediante la experimentación y el análisis de los resultados obtenidos, los cuales fueron confrontados con varios casos de estudio registrados en la literatura científica, en los experimentos se analizaron las características de la exhaustividad, la precisión, la correctitud, la completitud y la consistencia de los elementos básicos (objetivo, entidad, operación y agente).

Palabras Claves: Ingeniería de Software, procesamiento de lenguaje natural, educación de requisitos, validación de requisitos, objetivo, diagrama de objetivos de KAOS.

Abstract

The elicitation and validation of software requirements are specific activities belonging to software engineering which are performed in the early stages of the software life cycle, namely, definition and analysis. In these phases the goal is to capture the requirements to be met by the system (software) and transform them into conceptual models that explain the significant components of the problem area. However, there are problems in the conception of the products obtained in these phases because of: (i) the communication gap existing between the stakeholder (user) and the analyst due to limited or insufficient communication caused by the differences in their specialties, and (ii) the verbal ambiguity present in the universe of discourse. These problems make it impossible to correctly identify and validate the requirements that must be met by the system (software). Likewise, they increase the cost of the final product because errors must be corrected during the system's (software) implementation and production stages.

In order to: (i) improve the results of the requirements elicitation and validation phases, and (ii) reduce the communication gap between the stakeholder and the analyst, a number of conceptual schemes are used. One of such is the KAOS (Knowledge Acquisition in Automated Specification) diagram, which allows users to identify and validate the requirements associated with the system (software). Furthermore, the UML diagrams, which belong to the third phase (design) of the software life cycle and are a part of the analyst's domain, can be obtained from this diagram. However, the following problems have been observed in the various studies which specify or use the KAOS goal diagram: (i) lack of consistency between natural language and the basic components of the KAOS goal diagram, (ii) lack of terminological tools that can be used to minimize the semantic ambiguity

in the universe of discourse, (iii) lack of semi-automatic validation of software requirements performed by the stakeholder during the initial stage, and (iv) the corresponding goal diagram is not obtained semi-automatically.

In order to solve the problems stated in the preceding paragraphs, the following objectives were achieved in this doctoral dissertation: (i) characterize the components of the KAOS goal diagram. (ii) Build an Ontology that represents the components of the KAOS goal diagram more accurately. (iii) Define a set of morphosyntactic forms which make it possible to extract the basic components of the KAOS goal diagram from natural language (goal, entity, operation and agent) to solve the problem of consistency between natural language and the basic components of the KAOS goal diagram. (iv) Define a set of semantic rules that make it possible to minimize the semantic ambiguity found in the universe of discourse that negatively affects the activities associated with software requirements elicitation and validation. (v) Create a terminology processing model for eliciting and validating the software requirements expressed in natural language, with the help of the user and in an interactive manner. (vi) Develop the NL2KAOS application (written in JAVA, JSP and PHP) which automates the proposed model and includes the morphosyntactic forms and semantic rules defined in this dissertation. (vii) Semi-automatically obtain the KAOS goal diagram. (viii) Evaluate the model by experimenting and analyzing the results obtained; such results were compared with a number of case studies appearing in the scientific literature. Additionally, the exhaustive, precision, correctness, completeness and consistency of the basic components (goal, entity, operation and agent).

Keywords: Keywords: Software Engineering, natural language processing, requirements elicitation, requirements validation, goal, KAOS goal diagram.

Contenido

	Pág.
Introducción	21
1.1 Motivación	21
1.2 Preguntas de Investigación.....	24
1.3 Objetivo de la tesis	25
1.4 Metodología propuesta	26
1.4.1 Plan de trabajo.....	27
1.5 Contribuciones principales alcanzadas	31
1.6 Organización del documento	34
1.7 Difusión de resultados	36
Procesamiento del Lenguaje Natural e Ingeniería de Requisitos: Marco Teórico y Estado del Arte.	40
2.1 Procesamiento del Lenguaje Natural	40
2.1.1 Lingüística Computacional	41
2.1.2 Ontología	42
2.1.3 Sentido de las palabras.....	47
2.1.4 Algoritmos de desambiguación	49
2.2 Ingeniería de Software y de Requisitos.....	53
2.2.1 Problemas asociados durante la Educación de Requisitos de Software	54
2.2.2 Definición de objetivos para un sistema (software) futuro	56
2.3 Conclusiones del capítulo	75
Caracterización semántica de los elementos que conforman el diagrama de objetivos de KAOS.....	77
3.1 Metodología KAOS	77
3.2 Caracterización conceptual de los elementos del modelo de KAOS	79
3.2.1 Objetivo	80
3.2.1.1 Objetivo tipo Lograr.....	81
3.2.1.2 Objetivo tipo Evitar.....	82
3.2.1.3 Objetivo tipo Parar	82
3.2.1.4 Objetivo tipo Mantener.....	83
3.2.1.5 Softgoal	83
3.2.1.6 Requisito.....	84
3.2.1.7 Expectativa	85
3.2.1.8 Refinamiento de Objetivo.....	86
3.2.1.9 Conflicto.....	88
3.2.2 Agente	89
3.2.2.1 Agente de Ambiente	89
3.2.2.2 Agente de Software	90

3.2.2.3 Relación de Responsabilidad y de Asignación	90
3.2.3 Operación	91
3.2.3.1 Operacionalización	92
3.2.3.2 Entrada	93
3.2.3.3 Salida.....	94
3.2.4 Evento	94
3.2.5 Objeto	95
3.2.6 Propiedad de Dominio	95
3.3 Ontología del metamodelo KAOS	96
3.4 Conclusiones del capítulo	99

Modelo de procesamiento terminológico interactivo para la definición de requisitos de software bajo la especificación KAOS..... 100

4.1 Vista general del modelo	101
4.1.1 Especificación en lenguaje natural textual	102
4.1.2 Etiquetado morfosintáctico.....	102
4.1.3 Identificación de palabras vacías	103
4.1.4 Caracterización morfosintáctica	104
4.1.4.1 Reglas morfosintácticas para caracterizar objetivos.....	108
4.1.4.2 Reglas Morfosintácticas para identificar agentes	111
4.1.4.3 Reglas morfosintácticas para identificar entidades y atributos	113
4.1.4.4 Reglas morfosintácticas para identificar operaciones.....	114
4.1.5 Proceso de desambiguación	116
4.1.5.1 Regla 1. Clasificación de objetivos.....	117
4.1.5.2 Regla 2. Clasificación de agente.....	119
4.1.5.3 Regla 3. Clasificación de Operaciones.....	120
4.1.5.4 Regla 4. Clasificación de entidad	121
4.1.6 Caracterización semántica	121
4.1.7 Validación elementos KAOS	121
4.1.8 Salida.....	122
4.2 Conclusiones del capítulo	122

Arquitectura y diseño del prototipo propuesto124

5.1 Arquitectura	124
4.1.1. Especificación en Lenguaje Natural Textual.....	126
5.1.2 Etiquetado morfosintáctico.....	127
5.1.3 Identificador de palabras vacías.....	132
5.1.4 Caracterización morfosintáctica	132
5.1.5 Proceso de desambiguación.....	133
5.1.6 Caracterización semántica.....	136
5.1.7 Validación elementos KAOS	137
5.1.8 Salida.....	138
5.2 Conclusiones del capítulo	138

Validación del modelo139

6.1 Variables a medir	139
6.1.1 Exhaustividad	140
6.1.2 Precisión	140
6.2 Experimentación	141
6.2.1 Introducción	141
6.2.2 Resultados y Análisis.....	142
6.2.2.1 Modelo propuesto Vs. Ontología de dominio.....	153

6.2.2.2 Técnica utilizada Vs. Tiempo	156
6.2.3 Validación factores de calidad.....	157
6.2.3.1 Experimento.....	158
6.2.3.1.1 Resultados y Análisis	158
Sistema de la pizzería	160
6.3 Conclusiones del capítulo	163
CONCLUSIONES	164
7.1 Contribuciones principales	164
7.2 Trabajos futuros.....	168
Anexo A. Manual de usuario	170
Anexo B. Casos de estudio adaptados y utilizados en la experimentación	181
Anexo C. Implementación Ontología	229
Anexo D. Casos de uso y diagrama de secuencias.....	235
REFERENCIAS	249

Lista de figuras

Pág.

Figura 2.1. Relaciones semánticas entre conceptos. Fuente: (Rodríguez, 2011).	45
Figura 2.2. Segmento de la taxonomía utilizada en WordNet. Fuente: (Rodríguez, 2011).	46
Figura 2.3. Estructura de una oración con palabras polisémicas Fuente: elaboración propia.	48
Figura 2.4. Formas de expresión de la realidad entre los usuarios y los analistas. Fuente: Elaboración propia.	56
Figura 2.5. Incapacidad de entender el lenguaje del otro. Fuente: Elaboración propia.	56
Figura 2.6. Ejemplo de diagrama de objetivos de I*. Fuente: Adaptada de: (Yu, 1995).	58
Figura 2.7. Elementos básicos del diagrama de objetivos de KAOS. Fuente: Adaptado de (Respect IT, 2007).	60
Figura 2.8. Esquema preconceptual caso Softcom. Fuente: (Zapata, et al., 2011).	63
Figura 2.9. Resumen de las actividades GBRAM. Fuente: (Antón, 1997).	68
Figura 3.1. Metamodelo KAOS. Fuente: (Matulevičius et al., 2007).	79
Figura 3.2. Representación gráfica de un objetivo. Fuente: adaptado de (Matulevičius et al., 2007).	81
Figura 3.3. Representación gráfica de un objetivo de logro. Fuente: adaptado de (Matulevičius et al., 2007).	81
Figura 3.4. Representación gráfica de un objetivo de evitar. Fuente: adaptado de (Matulevičius et al., 2007).	82
Figura 3.5. Representación gráfica de un objetivo de tipo parar. Fuente: adaptado de (Matulevičius et al., 2007).	83
Figura 3.6. Representación gráfica de un objetivo de mantenimiento. Fuente: adaptado de (Matulevičius et al., 2007).	83
Figura 3.7. Representación gráfica de Softgoal en el diagrama de KAOS.	84
Fuente: (Matulevičius et al., 2007).	84
Figura 3.8. Representación gráfica de un requisito. Fuente: adaptado de (Matulevičius et al., 2007).	85
Figura 3.9. Representación gráfica de una expectativa. Fuente: adaptado de (Matulevičius et al., 2007).	86
Figura 3.10. Representación gráfica de refinamiento tipo AND. Fuente: adaptado de (Matulevičius et al., 2007).	87
Figura 3.11. Representación gráfica de refinamiento tipo OR. Fuente: adaptado de (Matulevičius et al., 2007).	87

Figura 3.12. Representación gráfica de conflicto. Fuente: adaptado de (Matulevičius et al., 2007).	88
Figura 3.13. Representación gráfica de frontera. Fuente: (Matulevičius et al., 2007).	88
Figura 3.14. Representación gráfica de un agente. Fuente: adaptado de (Matulevičius et al., 2007).	89
Figura 3.15. Representación gráfica agente de ambiente. Fuente: adaptado de (Matulevičius et al., 2007).	90
Figura 3.16. Representación gráfica de Agentes de software en el diagrama de KAOS. Fuente: adaptado de (Matulevičius et al., 2007).	90
Figura 3.17. Representación gráfica de responsabilidad y asignación. Fuente: adaptado de (Matulevičius et al., 2007).	91
Figura 3.18. Representación gráfica de una operación. Fuente: adaptado de (Matulevičius et al., 2007).	91
Figura 3.19. Representación gráfica de operacionalización tipo AND. Fuente: adaptado de (Matulevičius et al., 2007).	92
Figura 3.20. Representación gráfica de operacionalización tipo OR. Fuente: adaptado de (Matulevičius et al., 2007).	93
Figura 3.21. Representación gráfica de entrada. Fuente: adaptado de (Matulevičius et al., 2007).	93
Figura 3.22. Representación gráfica de salida. Fuente: adaptado de (Matulevičius et al., 2007).	94
Figura 3.23. Representación gráfica de un evento. Fuente: adaptado de (Matulevičius et al., 2007).	94
Figura 3.24. Representación gráfica objeto. Fuente: adaptado de (Matulevičius et al., 2007).	95
Figura 3.25. Representación gráfica de propiedad de dominio. Fuente: adaptado de (Matulevičius et al., 2007).	95
Figura 3.26. Componentes Ontología elementos KAOS.	97
Figura 4.1. Modelo general propuesto. Fuente: elaboración propia.	101
Figura 5.1. Arquitectura general del modelo propuesto. Fuente: elaboración propia.	125
Figura 5.2. Diagrama de clases del modelo propuesto. Fuente: elaboración propia.	126
A continuación se describen los módulos que componen la arquitectura del modelo propuesto:	126
Figura 5.3. Diagrama de clases del Freeling. Fuente: adaptada de (Padró y Stanilovsky, 2012).	128
Figura 5.5. Estructura de la clase abstracta processor. Fuente: (Padró y Stanilovsky, 2012).	129
Figura 5.4. Clases de procesamiento de Freeling. Fuente: adaptada de (Padró y Stanilovsky, 2012).	130
Figura 6.1 Precisión vs. IDPPE diez casos de estudio. Fuente: elaboración propia	150
Figura 6.2. Exhaustividad vs. IDPPE diez casos de estudio. Fuente: elaboración propia.	150
Figura 6.3. Resultados identificación de operaciones del experimento realizado a los 10 casos de estudio. Fuente: elaboración propia	152

Figura A.1. Ingresar U del D y seleccionar algoritmo. Fuente: Elaboración propia.....	171
Figura A.2. Depurar elementos identificados. Fuente elaboración propia.....	172
Figura A.3. Depurar elementos identificados. Fuente elaboración propia.....	172
Figura A.4. Depurar elementos identificados. Fuente elaboración propia.....	173
Figura A.5. Seleccionar los elementos básicos. Fuente elaboración propia	174
Figura A.6. Seleccionar los elementos básicos. Fuente elaboración propia	174
Figura A.7. Seleccionar los elementos básicos. Fuente elaboración propia	175
Figura A.8. Seleccionar objetivo principal. Fuente elaboración propia	176
Figura A.9. Seleccionar el padre para cada objetivo. Fuente elaboración propia	177
Figura A.10. Asignar agentes. Fuente elaboración propia.....	178
Figura A.11. Asignar operación. Fuente elaboración propia.....	179
Figura A.12. Exportar diagrama. Fuente elaboración propia	180
Figura C1. Estructura de las clases del meta-modelo diagramado en la Ontología. Fuente: elaboración propia.	229
Figura C2. Grafo de la Ontología construida. Fuente: elaboración propia.	230
Figura C3. Inferencia incorrecta por parte del razonador, por tener varios elementos en el dominio de una propiedad de datos. Fuente: elaboración propia.	231
Figura C4. Clases anónimas para Object que definen condiciones necesarias a través de la cardinalidad. Fuente elaboración propia.....	234
Figura D1. Caso de uso “Escribir texto”. Fuente: elaboración propia.....	235
Figura D2. Caso de uso “Procesar texto”. Fuente: elaboración propia.	237
Figura D3. Caso de uso “Borrar Cuadro de texto”. Fuente: elaboración propia.	237
Figura D4. Casos de uso “Cargar ejemplo”. Fuente: elaboración propia.	238
Figura D5. Aprobar elementos identificados. Fuente: elaboración propia.....	239
Figura D6. Remover elementos. Fuente: elaboración propia.....	241
Figura D7. Seleccionar el objetivo principal. Fuente: elaboración propia.....	242
Figura D8. Establecer jerarquía entre objetivos. Fuente: elaboración propia.....	243
Figura D9. Establecer asignación de agentes. Fuente: elaboración propia	244
Figura D10. Establecer operacionalización de objetivos. Fuente: elaboración propia...	245
Figura D11. Exportar diagrama. Fuente: elaboración propia.	246
Figura D12. Fragmento 1 del diagrama de secuencias de la aplicación desarrollada. Fuente: elaboración propia.	247
Figura D13. Fragmento 2 del diagrama de secuencias de la aplicación desarrollada. Fuente: elaboración propia.	248

Lista de tablas

Pág.

Tabla 2.1. Fragmento lenguaje controlado UN_Lencep. Fuente: (Zapata, Gelbukh, y Arango, 2006c)	62
Tabla 2.2. Estructuras lingüísticas para caracterizar objetivos. Fuente: (Zapata y Vargas, 2011).	64
Tabla 2.3. Plantilla de entrevistas. Fuente: (Lezcano et al., 2013).....	66
Tabla 2.4. Verbos presentados por Antón. Fuente: (Antón, 1998).	68
Tabla 2.5. Sintaxis de representación de elementos en GBRAM. Fuente: (Antón, 1998). 70	
Tabla 2.6. Resumen de los trabajos que utilizan diferentes métodos orientados a objetivos en la Ingeniería de Software. Fuente: elaboración propia.	74
Tabla 4.1. Ejemplo de categorías gramaticales presentes en un texto	103
Tabla 4.2. Ejemplo de palabras vacías presentes en un texto	103
Tabla 4.3. Macro estructura de las perífrasis verbales. Fuente: (Genta, 2008).	106
Tabla 4.4. Estructura morfosintáctica de perífrasis verbales modales válidas en el modelo. Fuente: (Genta, 2008).....	107
Tabla 4.5. Ejemplo de las formas morfosintácticas definidas para caracterizar objetivos. Fuente: elaboración propia.	110
Tabla 4.6. Ejemplo de forma morfosintáctica para caracterizar agentes. Fuente: elaboración propia.	112
Tabla 4.7. Ejemplo de forma morfosintáctica para caracterizar entidades y atributos. Fuente: elaboración propia.	113
Tabla 4.8. Ejemplo de forma morfosintáctica para caracterizar operaciones. Fuente: elaboración propia.	116
Tabla 4.9. Clasificación verbos de objetivos. Adaptado de (Zapata y Lezcano, 2009)... 117	
Tabla 6.1. Resultados de los objetivos identificados por el modelo para el caso de estudio de la ambulancia. Fuente: elaboración propia.....	142
Tabla 6.2. Resultados de los agentes identificados por el modelo para el caso de estudio de la ambulancia. Fuente: elaboración propia.....	145
Tabla 6.3. Resultados de las operaciones identificadas por el modelo para el caso de estudio de la ambulancia. Fuente: elaboración propia.	146
Tabla 6.4. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio de la ambulancia. Fuente: elaboración propia.	147
Tabla 6.5. Resumen de resultados de objetivos, agentes y entidades, para los 10 casos de estudio. Fuente: elaboración propia.	149
Tabla 6.6. Resumen de resultados de operaciones para los diez casos de estudio. Fuente: elaboración propia.	151

Tabla 6.7. Comparativo de relación semántica asociada a los objetivos. Fuente: elaboración propia.	154
Tabla 6.8. Comparativo de relación semántica asociada a los agentes. Fuente: elaboración propia.	155
Tabla 6.9. Comparativo de relación semántica asociada a las operaciones. Fuente: elaboración propia.	155
Tabla 6.10. Comparativo de relación semántica asociada a las entidades. Fuente: elaboración propia.	155
Tabla 6.11. Tiempo de procesamiento de acuerdo con la técnica de desambiguación utilizada.	156
Tabla 6.12. Evaluación comparativa completitud. Fuente: elaboración propia.	158
Tabla 6.13. Evaluación comparativa completitud. Fuente: elaboración propia.	159
Tabla 6.14. Evaluación comparativa completitud. Fuente: elaboración propia.	159
Tabla 6.15. Evaluación comparativa completitud. Fuente: elaboración propia.	160
Tabla 6.16. Evaluación comparativa completitud. Fuente: elaboración propia.	161
Tabla 6.17. Evaluación comparativa completitud. Fuente: elaboración propia.	161
Tabla B.1. Resultados de los objetivos identificados por el modelo para el caso de estudio del ascensor. Fuente: elaboración propia.	189
Tabla B.2. Resultados de los agentes identificados por el modelo para el caso de estudio del ascensor. Fuente: elaboración propia.	191
Tabla B.3. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio del ascensor. Fuente: elaboración propia.	192
Tabla B.4. Resultados de las operaciones identificadas por el modelo para el caso de estudio del ascensor. Fuente: elaboración propia.	193
Tabla B.5. Resultados de los objetivos identificados por el modelo para el caso de estudio de la pizzería. Fuente: elaboración propia.	194
Tabla B.6. Resultados de los agentes identificados por el modelo para el caso de estudio de la pizzería. Fuente: elaboración propia.	197
Tabla B.7. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio de la pizzería. Fuente: elaboración propia.	198
Tabla B.8. Resultados de las operaciones identificadas por el modelo para el caso de estudio de la pizzería. Fuente: elaboración propia.	199
Tabla B.9. Resultados de los objetivos identificados por el modelo para el caso de estudio Guayacán. Fuente: elaboración propia.	200
Tabla B.10. Resultados de los agentes identificados por el modelo para el caso de estudio Guayacán. Fuente: elaboración propia.	202
Tabla B.11. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio Guayacán. Fuente: elaboración propia.	203
Tabla B.12. Resultados de las operaciones identificadas por el modelo para el caso de estudio Guayacán. Fuente: elaboración propia.	204
Tabla B.13. Resultados de los objetivos identificados por el modelo para el caso de estudio Marinozzi. Fuente: elaboración propia.	205
Tabla B.14. Resultados de los agentes identificados por el modelo para el caso de estudio Marinozzi. Fuente: elaboración propia.	206

Tabla B.15. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio Marinozzi. Fuente: elaboración propia.....	207
Tabla B.16. Resultados de las operaciones identificadas por el modelo para el caso de estudio Marinozzi. Fuente: elaboración propia.....	208
Tabla B.17. Resultados de los objetivos identificados por el modelo para el caso de estudio Carrefur. Fuente: elaboración propia.	209
Tabla B.18. Resultados de los agentes identificados por el modelo para el caso de estudio Carrefur. Fuente: elaboración propia.	210
Tabla B.19. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio Carrefur. Fuente: elaboración propia.....	211
Tabla B.20. Resultados de las operaciones identificadas por el modelo para el caso de estudio Carrefur. Fuente: elaboración propia.	212
Tabla B.21. Resultados de los objetivos identificados por el modelo para el caso de estudio Colácteos. Fuente: elaboración propia.	213
Tabla B.22. Resultados de los agentes identificados por el modelo para el caso de estudio Colácteos. Fuente: elaboración propia.....	214
Tabla B.23. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio Colácteos. Fuente: elaboración propia.	215
Tabla B.24. Resultados de las operaciones identificadas por el modelo para el caso de estudio Colácteos. Fuente: elaboración propia.	216
Tabla B.25. Resultados de los objetivos identificados por el modelo para el caso de estudio Colcerámicas. Fuente: elaboración propia.....	217
Tabla B.26. Resultados de los agentes identificados por el modelo para el caso de estudio Colcerámicas. Fuente: elaboración propia.	218
Tabla B.27. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio Colcerámicas. Fuente: elaboración propia.	219
Tabla B.28. Resultados de las operaciones identificadas por el modelo para el caso de estudio Colcerámicas. Fuente: elaboración propia.....	220
Tabla B.29. Resultados de los objetivos identificados por el modelo para el caso de estudio ENCA. Fuente: elaboración propia.	221
Tabla B.30. Resultados de los agentes identificados por el modelo para el caso de estudio ENCA. Fuente: elaboración propia.....	222
Tabla B.31. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio ENCA. Fuente: elaboración propia.....	223
Tabla B.32. Resultados de las operaciones identificadas por el modelo para el caso de estudio ENCA. Fuente: elaboración propia.	224
Tabla B.33. Resultados de los objetivos identificados por el modelo para el caso de estudio Yamaha. Fuente: elaboración propia.....	225
Tabla B.34. Resultados de los agentes identificados por el modelo para el caso de estudio Yamaha. Fuente: elaboración propia.	226
Tabla B.35. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio Yamaha. Fuente: elaboración propia.....	227
Tabla B.36. Resultados de las operaciones identificadas por el modelo para el caso de estudio Yamaha. Fuente: elaboración propia.....	228

Lista de abreviaciones

KAOS	Knowledge Acquisition in Automated Specification
JSP	Java Server Pages
PHP	Hypertext Pre-Processor
UML	Unified Modeling Language
RUP	Rational Unified Process
PLN	Procesamiento Lenguaje Natural
LC	Lingüística Computacional
IA	Inteligencia Artificial
IR	Ingeniería Requisitos
LN	Lenguaje Natural
IS	Ingeniería de Software
RO	Regla Objetivo
FMOB	Forma morfosintáctica objetivo
FMAG	Forma morfosintáctica agente
FME	Forma morfosintáctica entidad
FMOP	Forma morfosintáctica operación
GBRAM	Goal-Based Requirements Analysis Method
MFS	Most Frequent Sense
UKB	Knowledge Based Unsupervised
OWL	Web Ontology Language
CAD	Sistema de Despacho Computarizado
TDG	Trabajo Dirigido de Grado
MCR	Multilingual Central Repository
DRAE	Diccionario Real Academia Española
SR	Strategic Rationale Model
SD	Strategic Dependency Model
IC	Information Content

Capítulo 1

Introducción

1.1 Motivación

Debido al incremento en el uso del computador se ha generado un crecimiento en la demanda de software; es por ello que, éste se convirtió en un elemento indispensable del computador y el responsable de la mayor parte de los costos económicos debido a su demanda y al mantenimiento correctivo (ocasionado por las falencias presentadas en las fases de definición y análisis de requisitos) que se debe realizar a la pieza de software. A finales de los años 60 se presentó la crisis de software, para esa época, nació una nueva rama en informática llamada Ingeniería de Software, ésta implementa y utiliza los principios de la ingeniería para el desarrollo del software con el propósito de alcanzar sistemas más confiables, los cuales deben mantener consistencia con las necesidades de los interesados, además, ofrece un enfoque metódico, cuidadoso y cuantificable para el proceso de desarrollo, puesta en marcha y mantenimiento del software (Pressman, 1998).

Entre las metodologías de desarrollo que la Ingeniería de Software utiliza, se encuentra el Proceso Unificado de Rational (Rational Unified Process –RUP-), el cual emplea un conjunto de actividades para transformar las especificaciones de los requisitos de un interesado en un sistema informático (Jacobson, Booch y Rumbaugh, 2001). En este proceso se utiliza el Lenguaje Unificado de Modelamiento (UML) para elaborar los modelos conceptuales del sistema informático.

Según (Jacobson, et al., 2001), RUP tiene un ciclo de vida compuesto por las siguientes fases:

- Definición: en esta fase se capturan los requisitos que debe cumplir el sistema; es decir, se debe describir su comportamiento desde el punto de vista del interesado y el entorno.
- Análisis: los requisitos obtenidos en la fase de definición se transforman en modelos conceptuales, explicando los elementos significativos en el dominio del problema. Los requisitos descritos en la fase de definición se refinan y se estructuran para obtener un conocimiento más preciso representado por modelos lógicos que ayuden a comprender el sistema informático.
- Diseño: el objetivo principal de esta fase es encontrar la forma para que el sistema soporte todos los requisitos obtenidos en la definición.
- Construcción: en esta fase se inicia la transformación e integración de los modelos lógicos al físico, lo cual incluye el desarrollo de código, la base de datos y la implementación de las interfaces del sistema.
- Transición: el propósito principal de esta fase, está orientado a realizar el proceso de transición del sistema antiguo (si lo hay) al nuevo. En esta transición se deben realizar las actividades de instalación del sistema informático y el entrenamiento de los usuarios.
- Producción: se consideran en esta fase, las actividades que ocurren cuando el producto de software está operando y ofreciendo sus servicios a los usuarios finales.

Este es un proceso cíclico e iterativo, en donde cada una de las fases requiere la información de las anteriores y un cambio en una de ellas refleja cambios en las demás.

En las fases de definición, análisis y diseño se utilizan diversos esquemas conceptuales que ayudan a cumplir con los objetivos correspondientes; particularmente, en las fases de definición y análisis se emplea el diagrama de objetivos de KAOS (Knowledge Acquisition in Automated Specification) que permite reducir la brecha de comunicación entre el interesado y el analista durante el proceso de educación de requisitos de software (Lamsweerde, 2000).

Sin embargo, de este diagrama de objetivos no se logra identificar: (i) la consistencia que debe existir entre el lenguaje natural y la caracterización de los elementos básicos que conforman dicho diagrama, (ii) el uso de herramientas terminológicas que permitan minimizar la ambigüedad semántica presente en el universo del discurso, y (iii) la obtención semiautomática de los elementos básicos (Objetivo, entidad, agente, operación) que conforman el diagrama de objetivos de KAOS.

Con el propósito de reducir esa brecha de comunicación existente entre analista e interesado, se propuso en esta tesis doctoral, presentar un modelo interactivo que permita identificar y validar los requisitos del software futuro durante el proceso de educación de requisitos y antes de iniciar la construcción de los modelos conceptuales de UML (Casos de Uso, diagramas de clase, diagramas de secuencias entre otros) por parte del analista, modelos que, en la mayoría de los casos no son comprendidos por los interesados (usuarios).

Para la elaboración de este modelo interactivo, se utilizó un lenguaje intermedio entre el lenguaje natural y los modelos conceptuales, dicho modelo está basado en el diagrama de objetivos de KAOS. Durante el desarrollo de este proyecto doctoral se logró: (i) caracterizar los elementos del diagrama de objetivos de KAOS, (ii) crear una ontología para describir los componentes que representan el diagrama de objetivos de KAOS, (iii) realizar la caracterización morfosintáctica de los elementos básicos (objetivo, entidad, operación y agente) del diagrama de objetivos de KAOS, (iv) establecer las reglas semánticas para la identificación de los elementos básicos del diagrama de objetivos de KAOS, (v) utilizar diferentes técnicas de desambiguación, para realizar el análisis terminológico que permitió dar una mejor precisión a las especificaciones obtenidas a través dicho diagrama, (vi) identificar a partir del lenguaje natural, los elementos básicos que conforman el diagrama de objetivos de KAOS, es decir, lograr la consistencia que debe existir entre el lenguaje natural y los elementos básicos que conforman dicho diagrama, y (vii) desarrollar la aplicación NL2KAOS que permite obtener y validar

de manera semiautomática e interactiva los requisitos de software a través de la metodología KAOS.

Se utilizó en esta tesis doctoral el diagrama de objetivos de KAOS porque, de acuerdo con (Lamsweerde, 2000), permite: (i) reducir la brecha de comunicación entre analista e interesado, (ii) observar la jerarquía de los objetivos, lo cual facilita alcanzar los requisitos que deberá cumplir el sistema (software) para satisfacer los objetivos organizacionales, (iii) reflejar los requisitos del dominio que son expresados por parte de los interesados en el universo del discurso.

1.2 Preguntas de Investigación

A partir del análisis de la literatura, presentado en el capítulo 2 de esta tesis, se puede concluir que la automatización del proceso de educación y validación de requisitos de software a través del diagrama de objetivos de KAOS es un tema que demanda continuar en esta línea de investigación, lo que conlleva a las siguientes preguntas de investigación:

Pregunta general: ¿Cómo se puede implementar un modelo de procesamiento terminológico para la educación y validación interactiva de requisitos de software descritos en lenguaje natural?

Preguntas específicas:

1. ¿Cómo caracterizar semánticamente los elementos que componen la definición de requisitos de software bajo el paradigma del diagrama de objetivos de KAOS para que sean comprensibles por una máquina sin ambigüedades?

2. ¿Cómo diseñar un modelo de procesamiento terminológico interactivo que, a partir de requisitos de software inicialmente descritos en lenguaje natural, permita obtener y validar la caracterización semántica definida en la pregunta número dos?
3. ¿Cómo implementar un ambiente de prueba (prototipo) que, haciendo uso del modelo de procesamiento terminológico propuesto permita su valoración?
4. ¿Cómo validar el modelo de procesamiento terminológico propuesto mediante la utilización de casos de prueba con el fin de valorar sus fortalezas y debilidades?

1.3 Objetivo de la tesis

General: Crear un modelo de procesamiento terminológico para la educación y validación interactiva de requisitos de software inicialmente descritos en lenguaje natural.

Específicos:

1. Caracterizar semánticamente los elementos que componen la definición de requisitos de software bajo el paradigma del diagrama de objetivos de KAOS para que sean comprensibles por una máquina sin ambigüedades.
2. Diseñar un modelo de procesamiento terminológico interactivo que, a partir de requisitos de software inicialmente descritos en lenguaje natural, permita obtener y validar la caracterización semántica definida en el objetivo anterior.

3. Implementar un ambiente de prueba (prototipo) que haciendo uso del modelo de procesamiento terminológico propuesto permita su valoración.
4. Validar el modelo de procesamiento terminológico propuesto mediante la utilización de casos de prueba con el fin de valorar sus fortalezas y debilidades.

1.4 Metodología propuesta

La Ingeniería de Software, intenta mejorar la calidad y la satisfacción de los interesados mientras reduce los costos y tiempos de entrega de los sistemas informáticos, además, busca corregir los errores en los que se incurre durante las primeras fases del ciclo de vida del software (definición y validación de requisitos). En este subcapítulo, se propone una metodología de trabajo a seguir para desarrollar una propuesta que permita la automatización de un modelo interactivo para la educación y validación de requisitos de software basados en el diagrama de objetivos de KAOS, de tal forma, que permita superar algunas de las limitaciones planteadas en la etapa de educación y validación de requisitos de software.

La automatización del proceso para obtener y validar los requisitos de software a través de la metodología KAOS, requiere del uso de unas reglas de traducción y unos recursos lingüísticos a partir de un texto de especificaciones en lenguaje natural. Dicho diagrama debe capturar las características sintácticas y semánticas descritas en el universo del discurso, el cual deberá ser presentado a través de un texto en lenguaje natural.

Algunos de los requisitos que debe cumplir el modelo interactivo para la educación de requisitos basado en la metodología KAOS que se propone en esta tesis doctoral son:

- Utilizar la sintaxis y la semántica del lenguaje natural.
- Definir reglas de traducción que conduzcan a la obtención semiautomática de los elementos básicos del diagrama de objetivos de KAOS con la mayor completitud posible, a partir de una especificación textual.
- Utilizar la mayor cantidad posible de palabras incluidas en las descripciones del lenguaje natural, para que contribuyan a facilitar la conversión de los elementos básicos del diagrama de objetivos de KAOS.
- Hacer uso de las ontologías como mecanismo de representación formal para representar los elementos que conforman el diagrama de objetivos de KAOS. Asimismo, se deberá hacer uso técnicas de desambiguación, para mejorar el entendimiento semántico del texto en lenguaje natural que se tendrá como entrada para el modelo propuesto en esta tesis.
- Los elementos obtenidos correspondientes al diagrama de objetivos de KAOS deberán tener una representación que pueda ser entendida por el interesado y el analista para lograr su validación.

Los aspectos metodológicos en el marco de la creación de un modelo de procesamiento terminológico para la educación y validación de requisitos de software, basado en la metodología KAOS, buscan establecer los pasos a seguir para obtener el resultado esperado al final de esta tesis doctoral.

1.4.1 Plan de trabajo

En este proyecto doctoral se manejaron dos frentes de trabajo, el teórico y el aplicado, lo cual implicó que su desarrollo metodológico se realizará a través de las siguientes dos fases:

Teórica. Contempló: (i) la caracterización semántica de los elementos que componen la definición de requisitos de software bajo el paradigma del diagrama de objetivos de KAOS para que sean comprensibles por una máquina sin ambigüedades, y (ii) el diseño de un modelo de procesamiento terminológico interactivo que, a partir de requisitos de software inicialmente descritos en lenguaje natural, permitiera obtener y validar la caracterización semántica planteada en el numeral anterior. El desarrollo de esta fase permitió enfrentar el problema de la ambigüedad semántica de tipo polisémica que se presenta durante la educación de requisitos de software.

Práctica. Está enfocada desde el punto de vista aplicativo, consistió en implementar un ambiente de prueba (prototipo) a través de un modelo de procesamiento terminológico para la obtención y validación de requisitos de software basado en la metodología KAOS. Asimismo, se validó el modelo propuesto mediante la utilización de casos de prueba con el fin de medir sus fortalezas y debilidades.

La realización de este proyecto se dividió en cinco etapas principales, las cuales están agrupadas en dos fases asociadas: (i) la parte teórica, y (ii) la parte práctica respectivamente.

Fase 1. Desarrollo de Investigación Teórica

Etapa 1.1. Caracterización semántica de los elementos básicos que componen la definición de requisitos de software bajo el paradigma del diagrama de objetivos de KAOS para que sean comprensibles por una máquina sin ambigüedades.

Actividad 1.1.1 Identificación y definición de: (i) las técnicas y modelos utilizados para la educación de requisitos de software, y (ii) el problema asociado durante la identificación y validación de requisitos de software.

Actividad 1.1.2. Abordar el tema de la teoría lingüística, para utilizar los recursos necesarios que permitieran determinar las características lingüísticas, para la caracterización y obtención de los elementos que conforman el diagrama de objetivos de KAOS a partir del procesamiento de Lenguaje Natural y la creación de Ontologías.

El resultado de esta etapa se presenta más detalladamente en los capítulos 2 y 3. Las actividades que se desarrollaron en esta etapa aportan directamente al cumplimiento del primer objetivo consistente en caracterizar semánticamente los elementos que componen la definición de requisitos de software bajo el paradigma del diagrama de objetivos de KAOS para que sean comprensibles por una máquina sin ambigüedades.

Etapa 1.2: Diseñar un modelo de procesamiento terminológico interactivo que, a partir de requisitos de software inicialmente descritos en lenguaje natural, permita obtener y validar la caracterización semántica definida en la etapa 1.1.

Actividad 1.2.1. Abordar el tema de la teoría lingüística (análisis morfosintáctico y semántico), para utilizar los recursos necesarios que permitan determinar las características lingüísticas que se deban utilizar para crear, a partir de lenguaje natural, un modelo semiautomático de procesamiento terminológico que permita minimizar la ambigüedad semántica en especificaciones textuales.

Actividad 1.2.2. Definir la estructura semántica y el procesamiento terminológico que se deberá cumplir para minimizar la ambigüedad semántica en especificaciones textuales durante la obtención y validación de los requisitos de software.

Actividad 1.2.3 Diseñar e implementar un modelo de procesamiento terminológico para minimizar la ambigüedad semántica en especificaciones textuales durante la obtención y validación de los requisitos de software.

El resultado de esta etapa es explicado más detalladamente en el capítulo 4. Todas las actividades de esta etapa están orientadas a cumplir con el segundo objetivo consistente en diseñar un modelo de procesamiento terminológico interactivo que, a partir de requisitos de software inicialmente descritos en lenguaje natural, permita obtener y validar la caracterización semántica definida en el primer objetivo.

Fase 2. Desarrollo de investigación aplicada

Etapa 2.1: implementación de un ambiente de prueba (prototipo) a través de un modelo de procesamiento terminológico para la obtención y validación semiautomática de requisitos de software basado en el diagrama de objetivos de KAOS.

Actividad 2.1.1. Identificar los requisitos que deberá cumplir el ambiente de prueba (prototipo).

Actividad 2.1.2 Realizar el diseño del ambiente de prueba (prototipo).

Actividad 2.1.3 Desarrollar el algoritmo que permita obtener el ambiente de prueba (prototipo) para validar los requisitos de una pieza de software a través de la automatización de los elementos básicos que conforman el diagrama de objetivos de KAOS.

Se implementó un ambiente de prueba (prototipo) a través del desarrollo de una aplicación (software) llamada NL2KAOS con el fin de generar un ambiente para valorar el desempeño del modelo. En el capítulo 5, se detallan los aspectos relacionados con el desarrollo del prototipo, además, se presentan las características principales de la respectiva implementación. Esta etapa está orientada a cumplir con el tercer objetivo consistente en implementar un ambiente

de prueba (prototipo) que haciendo uso del modelo de procesamiento terminológico propuesto permita su valoración.

Etapa 2.2: validar el modelo propuesto mediante la utilización de casos de prueba con el fin de medir sus fortalezas y debilidades.

Actividad 2.2.1. Diseñar las pruebas de validación del modelo.

Actividad 2.2.2. Realizar pruebas de validación del modelo.

Actividad 2.2.3. Realizar el análisis de resultados y determinar las fortalezas y debilidades del modelo propuesto.

Esta etapa incluyó la evaluación experimental del modelo de composición propuesto, se utilizaron varios casos de prueba logrando identificar fortalezas y debilidades del modelo.

Los resultados de esta etapa se ven más detalladamente en el capítulo 6. Las actividades desarrolladas están orientadas a cumplir con el cuarto objetivo consistente en validar el modelo de procesamiento terminológico propuesto mediante la utilización de casos de prueba con el fin de valorar sus fortalezas y debilidades.

1.5 Contribuciones principales alcanzadas

Primera contribución:

Desarrollo de una Ontología con base en el metamodelo del diagrama de objetivos de KAOS propuesto por (Matulevičius, Heymans y Opdahl, 2007). La Ontología desarrollada permite la posibilidad de tener un lenguaje propio de las tecnologías

de la web semántica, permitiendo representar con mayor precisión los elementos del diagrama de objetivos KAOS. En el capítulo 3 y en el anexo “C” de esta tesis se detalla el procedimiento utilizado para esta contribución.

Segunda contribución:

Definición de un conjunto de formas morfosintácticas que permiten la caracterización a partir de lenguaje natural de los elementos básicos (objetivo, entidad, operación y agente) del diagrama de objetivos de KAOS. Esta contribución permite brindar una solución a la necesidad de identificar la consistencia que debe existir entre el lenguaje natural y los elementos básicos que conforman diagrama de objetivos de KAOS, es decir, garantizar la coherencia que debe existir entre la especificación textual y los elementos básicos correspondientes al diagrama de objetivos de KAOS. Esta contribución incluye el uso de los siguientes elementos gramaticales: (i) perífrasis verbales, (ii) sintagma nominal, (iii) sustantivo, (iv) verbo, (v) complemento, y (vi) caracterización y definición de las formas morfosintácticas aplicadas. En el capítulo 4 de esta tesis se detalla el procedimiento utilizado para esta contribución.

Tercera contribución:

(i) Diseño general del modelo, (ii) definición de reglas semánticas para la identificación de los elementos básicos del diagrama de objetivos de KAOS, y (iii) definición de un modelo de procesamiento terminológico basado en las técnicas Simplified Lesk, MFS y UKB para minimizar la ambigüedad semántica de tipo polisémica, la cual afecta de forma negativa el proceso utilizado durante la educación de requisitos de software. Esta contribución incluyó el análisis de: (i) la ambigüedad semántica de tipo polisémica, y (ii) las diferentes técnicas utilizadas para minimizar la ambigüedad semántica presente en el universo del discurso. En el capítulo 4 de esta tesis se detalla el procedimiento utilizado para esta contribución.

Cuarta contribución:

Creación de un modelo de procesamiento terminológico para la educación y validación interactiva con el usuario de requisitos de software inicialmente descritos en lenguaje natural, el cual incluye los siguientes aspectos: (i) arquitectura del modelo, (ii) desarrollo de la aplicación NL2KAOS (elaborada en JAVA, JSP y PHP) que permite automatizar las formas morfosintácticas y las reglas semánticas definidas en el capítulo 4 de esta tesis, (iii) procesamiento terminológico para minimizar la ambigüedad semántica de tipo polisémica, (iv) obtener de manera semiautomática los elementos básicos del diagrama de objetivos de KAOS, y (v) construcción del diagrama de objetivos de KAOS con los elementos básicos. En el capítulo 5 de esta tesis se detalla el procedimiento utilizado para esta contribución.

Quinta contribución:

Evaluación del modelo de procesamiento terminológico para la educación y validación interactiva de requisitos de software inicialmente descritos en lenguaje natural. Para la realización de esta contribución, se utilizaron las características de exhaustividad, precisión, completitud, correctitud y consistencia. Además, se realizó un análisis de los resultados y se determinaron las fortalezas y las debilidades del modelo propuesto. En el capítulo 6 de esta tesis se detalla el procedimiento utilizado para esta contribución.

Con las contribuciones alcanzadas, se pretende:

- (i) Reducir la brecha de comunicación en la primera fase del ciclo de vida de software entre el interesado y el analista;
- (ii) Mejorar la comunicación entre el interesado y el analista en la primera fase del ciclo de vida de software a través de la creación de un conjunto

de formas y reglas morfosintácticas que permiten aproximar el lenguaje técnico del analista al lenguaje natural del interesado;

- (iii) Minimizar la ambigüedad semántica de tipo polisémica (al aplicar reglas morfosintácticas y semánticas) presente en el lenguaje natural textual, la cual afecta de forma negativa el ciclo de vida de software;
- (iv) Mejorar la calidad en la obtención de los elementos básicos correspondientes al diagrama de objetivos de KAOS, al construir una Ontología que permite identificar, describir y caracterizar semánticamente los elementos propuestos por la metodología KAOS para que sean comprensibles por una máquina sin ambigüedades
- (v) Posibilitar la validación de las especificaciones que debe cumplir el software futuro, en dicha validación, se ofrece un proceso interactivo para mejorar la especificación de requisitos de software con el apoyo de la metodología KAOS.

1.6 Organización del documento

Este documento de tesis está organizado de la siguiente manera:

El capítulo 2 proporciona una visión general de la literatura relacionada con las metodologías y las técnicas utilizadas para la educación de requisitos en el área de la Ingeniería de Software. Además, presenta una comparación de los diferentes métodos utilizados en esta área.

El capítulo 3 presenta la caracterización de los elementos que componen el diagrama de objetivos de KAOS. Además, se realiza: (i) la identificación de los elementos más relevantes que serán utilizados durante el desarrollo de esta tesis, y (ii) detalla el desarrollo de la Ontología construida en el marco de esta tesis que representa el metamodelo del diagrama de objetivos de KAOS.

En el capítulo 4 se presenta el modelo general de la propuesta que permite, a partir de Lenguaje Natural, realizar la caracterización morfosintáctica de los elementos básicos (objetivo, entidad, operación y agente) que conforman el diagrama de objetivos de KAOS. Asimismo, presenta diversas técnicas de desambiguación semántica de tipo polisémica, aplicadas a la lingüística computacional y que pueden ser usadas en la Ingeniería de Software. Además, se definen reglas de identificación y clasificación semántica de los elementos básicos que conforman el diagrama de objetivos de KAOS, las cuales tienen como propósito principal, realizar la traza y el proceso completo de identificación morfosintáctica + semántica.

En el capítulo 5, se presentan los componentes que conforman la arquitectura del sistema (software) NL2KAOS desarrollado en el marco de esta tesis, el cual permite la validación del modelo propuesto en esta tesis doctoral.

El capítulo 6 presenta la evaluación y experimentación del modelo propuesto mediante el uso del sistema (software) NL2KAOS desarrollado en el marco de esta tesis y un ambiente de prueba diseñado e implementado. Específicamente se analiza el comportamiento del modelo, frente a los diferentes problemas identificados en esta tesis. Para la evaluación del modelo propuesto se utilizaron varios casos de estudio registrados en la literatura científica. En esta validación se utilizaron las características de exhaustividad, precisión, completitud, correctitud y consistencia.

El capítulo 7 presenta las conclusiones generales y el trabajo futuro que se deriva de esta tesis doctoral.

Anexo A: Manual de Usuario NL2KAOS

En este anexo se presentan detalladamente los pasos a seguir para hacer uso del sistema (software) NL2KAOS desarrollado en el marco de esta tesis.

Anexo B: Casos de estudio adaptados

En este anexo se presentan los 10 casos de estudio utilizados en la experimentación del modelo propuesto.

Anexo C: Implementación Ontología

En este anexo se detallan algunos aspectos asociados a la implementación de la Ontología.

Anexo D: Diagrama casos de uso y secuencias

En este anexo se presentan los casos de uso y los diagramas de secuencias diseñados para la arquitectura del modelo propuesto.

1.7 Difusión de resultados

Los resultados de esta tesis se presentaron en diferentes revistas nacionales e internacionales, capítulos de libro y conferencias. Asimismo, de este trabajo de tesis doctoral, se desprendieron los siguientes trabajos relacionados: un trabajo dirigido de grado (TDG), una tesis de maestría y dos proyectos de investigación. En esta sección se resumen los diferentes resultados en publicaciones y proyectos asociados que soportan esta tesis.

Revistas Indexadas Internacionales:

- “Obtaining agents and entities from natural language”. En: Springer. Aceptado para publicar (noviembre de 2013).

- “Extraction of Goals and their Classification in the KAOS Model Using Natural Language Processing”. En Ingeniare, Revista Chilena de Ingeniería. ISSN 0718-3305. Aceptado para publicar (junio de 2014).

Revistas Indexadas Nacionales:

- "The application of and unresolved problems regarding the use of objectives in software engineering". En: Revista Ingeniería e Investigación ISSN: 0120-5609. Universidad Nacional de Colombia. Volumen 32 pp.63 – 67. Publicado (agosto de 2012).
- “Definition of a method to perform software requirements elicitation from the joint of cause effect diagram with KAOS goal diagram”. Revista Politécnica ISSN 1900-2351. Número 16 pp. 37-49. Publicado (junio de 2013).
- “Characterization of the elements the goal diagram KAOS from natural language”. Revista Colombiana de Tecnologías de Avanzada. ISSN: 1692-7257. Número 21 pp. 18-144. Publicado (agosto de 2013).
- Method for obtaining KAOS diagram entities and attributes from natural language. Revista Ingenierías. ISSN 1692-3324. Sometido (agosto de 2013).

Capítulos de libro:

- “Sistema de Gestión Documental Web para Requisitos de Software Bajo Especificaciones KAOS”. Libro: 16 Convención Científica de Ingeniería y Arquitectura Cujae, Editorial Cujae, pp119 – 127. La Habana-Cuba. 2012.

Congresos Internacionales:

- CIM: Coloquio de Investigación Multidisciplinaria. Evento Internacional. Ponencia: “Desarrollo y construcción de una Ontología basada en el Metamodelo del Diagrama de Objetivos KAOS”. Instituto Tecnológico de Orizaba. Orizaba, Veracruz, México. 2013.
- V Congreso Internacional de Computación y Telecomunicaciones. Ponencia: “Extracción y clasificación de agentes KAOS desde requisitos textuales usando procesamiento de lenguaje natural. Universidad Inca Garcilaso de la Vega. Lima-Perú. 2013.
- 16 Convención Científica de Ingeniería y Arquitectura. I Congreso Internacional de Ingeniería Informática y Sistemas de Información. Ponencia: “Sistema de Gestión Documental Web para Requisitos de Software Bajo Especificaciones KAOS”. Instituto Superior Politécnico José Antonio Echeverría. La Habana-Cuba. 2012.
- IX Congreso Internacional Electrónica y tecnologías de avanzada CIETA 2012. Ponencia: “Caracterización de los elementos del diagrama de objetivos de KAOS a partir de lenguaje natural”. Universidad de Pamplona. Cúcuta-Colombia. 2012.

Tesis de Maestría:

- “Un modelo de resolución de ambigüedad de sentidos de palabras para mejorar la calidad de resultados en una arquitectura de educación de requisitos de software”. Estado: propuesta aprobada y tesis en desarrollo.

Trabajo de Grado:

- “Modelo para la identificación de agentes KAOS mediante Procesamiento de Lenguaje Natural”. Estado: terminado

Proyectos de Investigación:

- Modelo de procesamiento terminológico basado en ontologías para la desambiguación verbal en la educación de requisitos de software. Código Hermes 18717. Financiado a través de la “Convocatoria del programa nacional de proyectos para el fortalecimiento de la investigación, la creación y la innovación en posgrados de la Universidad Nacional de Colombia 2013-2015”. Estado: en desarrollo.
- Un modelo de procesamiento terminológico para la obtención de requisitos de software basado en el diagrama de objetivos de KAOS. Código Hermes 15534. Financiado por la Universidad Nacional de Colombia Sede Medellín a través de la “Convocatoria DIME 2012 - financiación de proyectos de investigación modalidad 1”. Estado: en proceso de cierre.

Capítulo 2

Procesamiento del Lenguaje Natural e Ingeniería de Requisitos: Marco Teórico y Estado del Arte.

En este capítulo se suministra el marco teórico y se revisa el estado del arte de las áreas de investigación abarcadas. En la primera sección se presentan los principios básicos del procesamiento del lenguaje natural, en la segunda se analiza el tema de la Ingeniería de Software y más específicamente el área de Ingeniería de Requisitos. Asimismo, se describen las diferentes aproximaciones de las técnicas empleadas para llevar a cabo el proceso de educación y validación de requisitos de software.

2.1 Procesamiento del Lenguaje Natural

El Procesamiento del Lenguaje Natural (PLN) consiste en el estudio y el análisis de los aspectos lingüísticos de un texto (presentado en lenguaje natural) mediante la utilización de programas informáticos. El PLN es uno de los recursos utilizados en la Lingüística Computacional para realizar gran cantidad de tareas, las cuales representan mayores costos en términos de tiempo y dinero si son realizadas manualmente. Según (Verdejo, 1994), el lenguaje natural se caracteriza por: (i) su riqueza en vocabulario, (ii) flexibilidad, (iii) ambigüedad, y (iv) posibles interpretaciones del sentido literal de acuerdo con la situación que lo origina.

(Moreno y Molina, 1999), indican que: “el Procesamiento del Lenguaje Natural es una parte esencial de la Inteligencia Artificial que investiga y formula mecanismos computacionalmente efectivos que faciliten la interrelación hombre/máquina y permitan una comunicación mucho más fluida y menos rígida que los lenguajes formales”.

2.1.1 Lingüística Computacional

(Grishman, 1991), señala que la Lingüística Computacional (LC) es una disciplina que involucra componentes de la lingüística y la informática para tratar los aspectos computacionales del lenguaje natural. El objetivo principal de la LC está orientado a lograr la comprensión de los procesos lingüísticos. Los productos obtenidos mediante esta disciplina están encaminados a facilitar el procesamiento automático de textos y la recuperación de información. Algunos de los análisis que se deben realizar al lenguaje natural a través de la LC son: (i) el morfológico, (ii) el sintáctico, y (iii) el semántico.

(Montoyo, 2002) indica que, “para diseñar un sistema de PLN se requiere abundante conocimiento sobre las estructuras del lenguaje, como son el conocimiento morfológico, sintáctico, semántico y pragmático”. Además, se deben conseguir categorías gramaticales, roles sintácticos, dependencias, palabras compuestas, entre otros. Una forma de estructurar los problemas interdependientes del PLN podría incluir las siguientes fases y áreas del conocimiento:

- Palabras: utilización de las propias palabras, diferenciar si son palabras llenas (con contenido léxico; para ello se requiere de su categoría sintáctica) o no, y la influencia de ciertas composiciones de palabras en el aprendizaje de sentidos concretos.

- Categorías gramaticales: en principio, determinar si son sustantivos, verbos, adjetivos o adverbios.
- Análisis de la frase: se deben considerar las dependencias, roles sintácticos, unidades gramaticales, detección de palabras compuestas, etc. Las dependencias se representan mediante un árbol, es decir, si la palabra actúa como sujeto o está dentro de un sintagma nominal que interviene como raíz, será un núcleo verbal principal de la frase, en el que se van desarrollando los distintos sintagmas hasta llegar a las hojas terminales.
- Información de dominio: en este punto se abandona la cercanía con la palabra objetivo y se abarca una ventana más amplia, intentando involucrar el tema del discurso o del dominio. (Gale, Kenneth, Church y David, 1992b), señalan que las palabras dentro de un mismo texto o discurso han de estar estrechamente relacionadas semánticamente y, además, sería lógico que varias apariciones de la misma palabra se refieran, en todos o casi todos los casos, a un único sentido. Es el principio de “un sentido por discurso”.
- Corpus: Según el Diccionario de la Real Academia de la Lengua Española se define como “conjunto lo más extenso y ordenado posible de datos o textos científicos, literarios, etc., que pueden servir de base a una investigación”. Para el caso de esta tesis en particular, se hace referencia a: (i) texto, (ii) conjunto de frases, y (iii) párrafos o documentos textuales.

2.1.2 Ontología

Las ontologías son utilizadas para la conceptualización del conocimiento, definiendo los conceptos de diferentes dominios, sus relaciones y sus atributos. (Gómez, Corcho y Fernández, 2003), definen la Ontología como un término

clásico en filosofía que significa “tratado del ser”, o también, “tratado de todo lo que existe”.

(Gruber, 1993), definió la Ontología como: “especificación explícita de una conceptualización del conocimiento”. Las Ontologías en IA, pueden ser utilizadas tanto por humanos como por ordenadores. Asimismo, son definidas como una descripción explícita de conceptos o clases en un dominio, que se utiliza para el análisis conceptual de los lenguajes de modelado de datos. Las propiedades de las clases que describen varios aspectos y atributos del concepto son considerados como campos o propiedades a los que se les puede definir restricciones. Las clases son los elementos básicos de las Ontologías y pueden contener subclases que representan conceptos más específicos.

(Snasel, Moravec, y Pokorny, 2005), indican que las ontologías son especificaciones de una conceptualización abstracta y simplificada del mundo, que se representa para algún propósito.

(Eriksson, 2007), Señala que “desafortunadamente, existe una sorprendente brecha entre la modelación del conocimiento en ontologías y la documentación textual del conocimiento”. Además, indica que un acercamiento que permita combinar documentos y ontologías, permitirá a los usuarios acceder al conocimiento de múltiples maneras.

(Zapata, Giraldo y Urrego, 2010), indican que, "Las ontologías, como una forma de integrar y hacer explícitos los conocimientos correspondientes a cada una de las fases del ciclo de vida de desarrollo del software, constituyen una contribución muy importante en la profundización del conocimiento requerido en IS, para mantenerlo, adaptarlo, reutilizarlo y producir mejores aplicaciones". Asimismo, señalan que, "En general, las fases de definición, análisis y diseño emplean

ontologías existentes del dominio y construyen ontologías como productos intermedios para la transformación de modelos".

Por otra parte, (Samper, 2005), propone y valida una Ontología para el manejo de información de tráfico vial, para luego desarrollar herramientas informáticas, agentes inteligentes u otras aplicaciones que procesen la información y conocimiento representados para realizar diversas tareas relacionadas con la distribución de información de tráfico vial por parte de las autoridades o administraciones pertinentes.

Dado lo anterior, las ontologías constituyen un campo de investigación en diferentes áreas como la Ingeniería del Conocimiento, el Procesamiento de Lenguaje Natural, Sistemas Colaborativos, entre otros. En sí, las ontologías brindan un entendimiento compartido y común de un dominio que puede ser comunicado entre personas y sistemas de información de manera amplia y heterogénea. Además, proveen una conceptualización explícita (meta-información) que describe la semántica de los datos (Fensel, 2001).

Una de las ontologías más grandes y conocidas es WordNet, posee un tesoro con más de 100.000 términos del inglés, estableciendo jerarquías y relaciones semánticas entre ellos (p.e. sinonimia, hiperonimia, hiponimia, etc.), agrupándolos en verbos, adjetivos y adverbios, en donde cada término representa un concepto léxico subyacente (Fellbaum, 1998).

Aunque WordNet se encuentra en idioma inglés, existen repositorios como el Multilingual Central Repository (MCR) (Gonzalez, Agirre, Laparra, y Rigau, 2012), que permiten obtener el término equivalente en otro idioma, para el propósito de esta tesis, se dispone de éste en idioma español. El MCR, es una base de datos léxica en idioma español de relaciones semánticas entre términos, tales como hiperónimos, hipónimos y sinónimos (González, et al. 2012).

Hiperónimo: expresión cuyo significado incluye al de otra u otras expresiones; es decir, la expresión E1 es un hiperónimo de la expresión E2 si la actividad representada por E2 es un tipo de E1. Ejemplos: (i) viajar es hiperónimo de moverse, y (ii) pájaro es hiperónimo de gorrión y paloma. (DRAE, 2013).

Hipónimo: expresión cuyo significado está incluido en el de otra; es decir, X es un hipónimo de Y si todo X es un Y. Ejemplos: (i) Bagre es hipónimo de pescado, y (ii) gorrión es hipónimo de pájaro. (DRAE, 2013).

Sinónimo: expresión que tiene un mismo significado o un parecido con otra. Ejemplos: (i) boda es sinónimo de matrimonio, (ii) ebrio es sinónimo de borracho. (DRAE, 2013).

Es importante resaltar, que las ontologías permiten precisar la definición de un concepto con base en su propio dominio y, la semántica, se logra obtener a través de las relaciones existentes entre conceptos (Ver figura 2.1).

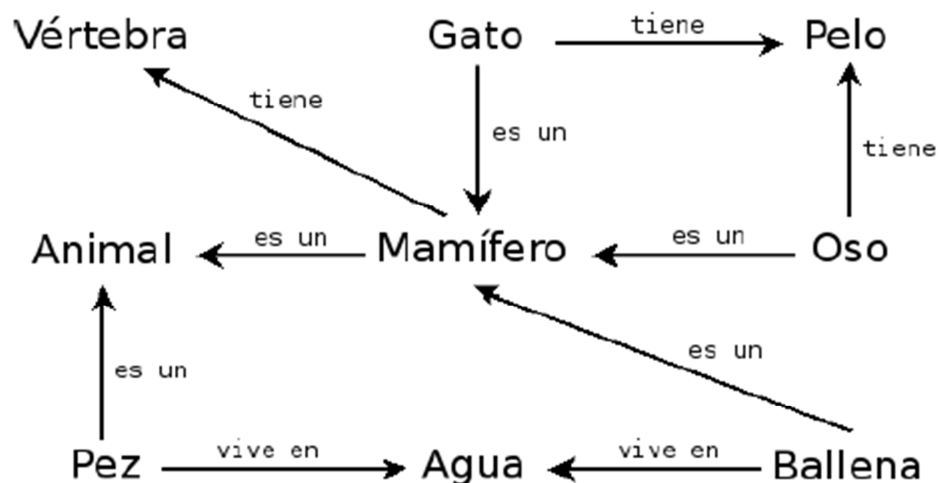


Figura 2.1. Relaciones semánticas entre conceptos. Fuente: (Rodríguez, 2011).

Además, es necesario diferenciar la similitud semántica existente entre conceptos y palabras en un contexto dado, por ejemplo, la palabra "naranja" concierne a dos

conceptos diferentes: (i) color, y (ii) fruta. Nótese en la figura 2.2 (segmento de la taxonomía utilizada en WordNet) que "NIKEL" es un concepto diferente para cada caso, al igual que sus respectivos ascendentes.

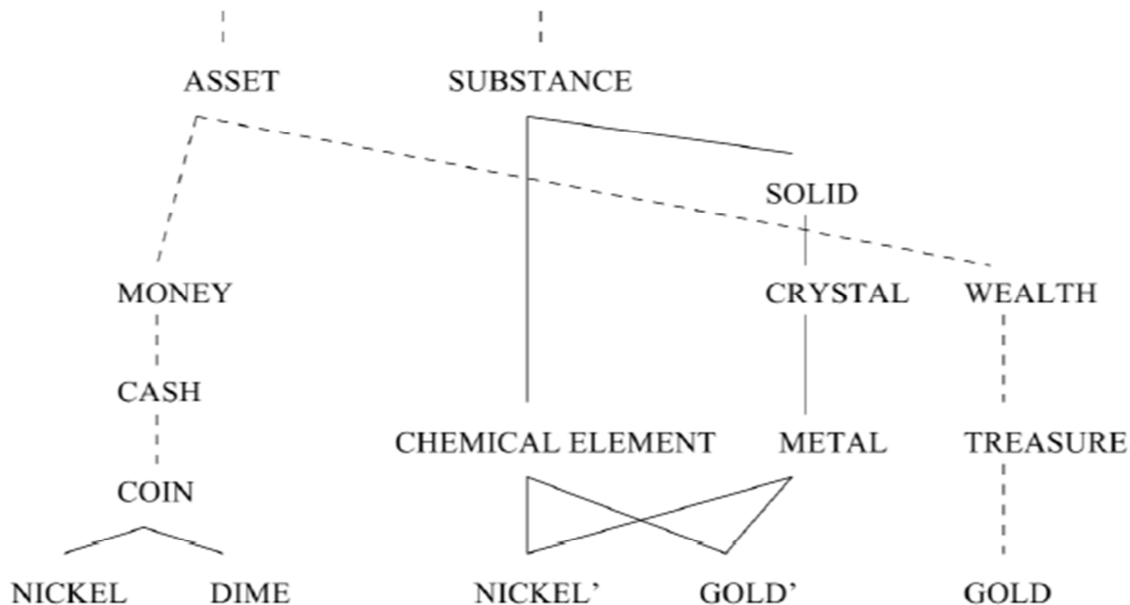


Figura 2.2. Segmento de la taxonomía utilizada en WordNet. Fuente: (Rodríguez, 2011).

El propósito general del uso de la semántica, es lograr entender el sentido de los conceptos, es decir, entender con claridad lo que se quiere expresar.

Actualmente, el lenguaje natural se ve afectado por ambigüedades de tipo: (i) léxico, (ii) sintáctico, (iii) semántico, y (iv) fonético. Este tipo de ambigüedades son ocasionadas por la vaguedad misma del lenguaje y por la poca precisión con la que es usado cotidianamente, además, por la gran cantidad de sentidos posibles asociados a una sola palabra en un idioma específico (Cueto, 2004). Una palabra que posee más de un sentido posible se le denomina palabra polisémica, es decir, es una palabra que tiene muchos sentidos pero que bajo un contexto determinado es posible intuir el sentido más correcto, aunque en muchos casos, es incluso difícil para el humano mismo descubrir el sentido más correcto (Rigau, 2002). Las

palabras más susceptibles a este tipo de ambigüedad son los: (i) verbos, (ii) adjetivos, y (iii) sustantivos.

La Ingeniería de Software es una de las ramas de las ciencias de la computación que intenta definir de manera clara y no ambigua los requisitos que un software debe cumplir. En este proceso de definición de requisitos intervienen diferentes actores tales como: (i) interesados (usuarios), (ii) ingenieros, y (iii) analistas de requisitos, entre otros. Pero, aun así, este proceso se torna difícil debido a las ambigüedades presentes en el lenguaje utilizado por los interesados, dado lo anterior, los analistas se ven obligados a crear mecanismos de representación intermedios entre los interesados y las máquinas. Sin embargo, estos mecanismos de representación siguen careciendo de claridad debido a la ambigüedad semántica presente en la definición de requisitos de software (Booch, Rumbaugh, Molina, Martínez, Jacobson, 2000).

Diversas técnicas de desambiguación polisémica han sido creadas a lo largo de la historia por diferentes grupos u organizaciones científicas debido a que este problema es transversal a diversas áreas del conocimiento (Navigli, 2009).

2.1.3 Sentido de las palabras

Una palabra polisémica cumple con una estructura computacional básica que puede ser vista como una etiqueta <label> que apunta a 1... <n> sentidos (<synsets>), pero a su vez un sentido (<synset>) puede ser apuntado por 1.... <n> etiquetas (<label>), por tanto el problema de desambiguación de este tipo de ambigüedad, se remite a seleccionar el mejor <synset> correspondiente a un <label> en un contexto dado (Navigli y Vannella, 2013).

Un sentido o synset, es una definición no ambigua de una abstracción en el mundo real, normalmente puede ser identificada por muchas palabras (<label>), además, tiene asociado un número único conocido y aceptado mundialmente por

las agrupaciones académicas que trabajan en esta área del conocimiento (Artiles, Gonzalo, y Sekine, 2007). En la figura 2.3, se presenta la estructura de una oración con palabras polisémicas en donde se puede observar la asociación de éstas con sus correspondientes synset.

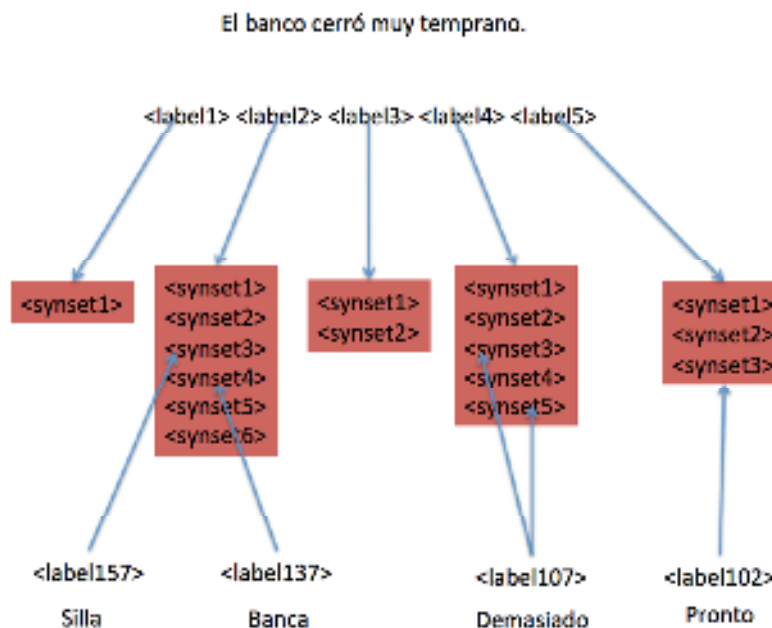


Figura 2.3. Estructura de una oración con palabras polisémicas

Fuente: elaboración propia.

La representación más común de un synset son tres letras mayúsculas al inicio, las cuales representan el idioma SPA (Español), ENG (Inglés), POR (portugués), etc. Sin embargo, son más usadas las letras ILI que significa que el concepto es el mismo independiente del idioma, posteriormente a las 3 letras iniciales el synset se presenta un número entero compuesto entre 4 a 11 números, finalmente, el synset posee la inicial del tipo de palabra, V (verbo), N (sustantivo), A (Adjetivo), etc... Un ejemplo de un synset es: ILI-8881283-V que representa que el synset hace referencia a un verbo y que es transversal a todos los idiomas.

2.1. 4 Algoritmos de desambiguación

En esta sección se presentan algunos algoritmos y métodos utilizados en desambiguación semántica que se basan en una representación ontológica de las palabras en español, permitiendo consultar las definiciones jerarquías semánticas (hiperónimos, hipónimos, sinónimos), de los sentidos (conceptos) de palabras en idioma español. Para ello, utilizan el repositorio MCR, que, como se indicó en la sección 2.1.2, se basa en WordNet

Lesk

El algoritmo Lesk se presentó por primera vez en 1986, éste tiene como propósito principal determinar el mejor synset (sentido) posible para una palabra en una oración dada (Lesk, 1986). Este algoritmo se basa en la suposición de que las palabras de un vecindario de una porción de texto tenderán a compartir un tema en común y, por tanto, es posible identificar el sentido más correcto con base en las definiciones del diccionario.

Una versión simplificada del algoritmo Lesk, es comparar la definición o los ejemplos del diccionario de una palabra ambigua con los demás términos contenidos en su vecindario. Su representación formal se presenta a continuación:

$$\text{scoreLesk}(S) = |\text{context}(w) \cap \text{gloss}(S)| \quad (1)$$

En donde: $\text{context}(w)$ representa el conjunto de expresiones de la frase donde está contenida la palabra a desambiguar; $\text{gloss}(S)$ representa la definición de la palabra.

El algoritmo Lesk debe cumplir con las siguientes reglas básicas: (i) para todos los sentidos de la palabra que se desea desambiguar, se debe contar la cantidad de

palabras que se encuentran en el vecindario de la palabra ambigua y en la definición de cada sentido de la palabra en el diccionario seleccionado, (ii) el sentido que ha de ser elegido, será aquel que tenga el mayor número en el conteo realizado, y (iii) no participan en el análisis las palabras vacías o “stop words” que son aquellas palabras diferentes a verbos, adjetivos o sustantivos.

Un ejemplo de operación del algoritmo Lesk se presenta a continuación:

Desambiguar la palabra “Pino y cono” en la oración: “Los conos de pino cuelgan en un árbol”.

Lesk([Pino], [Conos] [Cuelgan][Arbol]);

Sentidos posibles de “Pino”:

1. tipo de árbol de hoja perenne con hojas en forma de aguja
2. consumirse a través de la tristeza o enfermedad.

Sentidos posibles de "Cono"

1. cuerpo sólido que se estrecha a un punto
2. algo de esta forma sea sólida o hueca
3. fruto de ciertos árboles de hoja perenne

El algoritmo debe seleccionar el sentido #1 para el pino debido a que la palabra árbol se encuentra en la definición de este sentido, y a su vez se encuentra la frase o vecindario de la palabra a desambiguar, mientras que en el sentido 2 no hay coincidencia con ninguna de las palabras de la frase o vecindario de la palabra a desambiguar.

Para la palabra como el algoritmo debe seleccionar el sentido 3 debido a que posee 1 palabra presente en el vecindario (árbol).

Diversas mejoras pueden ser añadidas a este algoritmo, por ejemplo, realizar la búsqueda recursivamente, realizar el cálculo de similitud semántica entre conceptos para hallar el sentido más probable o extender la búsqueda a más de una frase o documento entero (Ríos, 2012).

Algoritmo MFS (Most Frequent Sense)

Consiste en una o múltiples bases de conocimiento que contienen un puntaje asociado a cada sentido de cada palabra, la manera de desambiguar consiste en utilizar el mejor sentido (con el puntaje más alto) para una palabra (Ng, Lee, 1996). Esta es la técnica de desambiguación más utilizada actualmente, debido a que diversas organizaciones académicas han venido trabajando en esta técnica y tienen en común múltiples bases de conocimiento, por ejemplo, Wordnet (Leacock, Chodorow, 1998) y (Miller, 1995).

Diversas mejoras se han ido incorporando a esta técnica de desambiguación, por ejemplo, el sentido más frecuente de cada palabra es ordenado de acuerdo a una temática o dominio particular, o el sentido más frecuente se adapta a modismos culturales que, así como las bases de conocimiento, se encuentran disponibles en la nube posibilitando determinar con mayor exactitud el mejor sentido.

Anualmente se realiza la competencia de SENSEVAL (Artiles, et al., 2007) donde diversas técnicas de desambiguación obtienen el sentido más adecuado para cada palabra de acuerdo a un contexto, y es allí donde se define el mejor sentido para cada palabra, esta competencia es aceptada mundialmente por todas las organizaciones académicas, lo cual constituye una gran base de conocimiento que permite remitir la tarea de desambiguación a una base de datos en la nube.

Distancia conceptual

Es una técnica usada para determinar qué tan cerca en la Ontología de Wordnet se encuentran dos conceptos, esto se mide contando los conceptos que se encuentra en el camino más corto entre los dos nodos a comparar (Navigli, 2009). Este cálculo de distancia conceptual fue adaptado por (Agirre y Rigau, 1996) para realizar tareas de desambiguación debido a que en una frase con palabras polisémicas, es posible desambiguar completamente la frase identificando los sentidos que hacen parte del camino más corto que une todos los nodos de la oración.

Puntos de grafo

Esta técnica permite crear un coeficiente para cada sentido probable en cada <label>, está orientada para determinar qué tan similar es su comportamiento en los párrafos o frases anteriores, es decir, esta técnica crea memoria estadística con base en un corpus entrenado, sugiere un puntaje a cada sentido que puede ser representado como un punto en el grafo, luego, y de acuerdo con la menor suma de los coeficientes en un camino, se puede determinar el mejor sentido (synset) de las palabras que generan ambigüedad en la oración (Navigli, Lapata, 2007).

UKB

Según (Agirre y Soroa, 2009), UKB es una colección de programas utilizados para desambiguación semántica basada en similitud léxica en relación con una base de conocimiento pre-existente. Combina las técnicas de puntos de grafo y del sentido más frecuente debido a que utiliza el page Rank como medida y una base de conocimiento existente en la web.

2.2 Ingeniería de Software y de Requisitos

El proceso de Ingeniería de Software involucra la aplicación de principios científicos que permiten realizar la representación ordenada de un problema o una necesidad en una solución elaborada de software, además, incluye el mantenimiento de éste hasta el final de su vida útil. Las tareas que involucra la Ingeniería de Software inician antes de la construcción y/o elaboración del código y continúa después de la finalización de la versión inicial de la pieza de software (Davis, 1993).

Los actores que utilizan un enfoque ingenieril para el desarrollo de software, deben cumplir con una serie de fases, éstas son conocidas comúnmente como el ciclo de vida del desarrollo de software que en su fase inicial comprende la Ingeniería de Requisitos (IR). La IR se encarga de entender las necesidades y/o requisitos de los interesados con relación a un Sistema (Software) y traducir sus necesidades a un conjunto de sentencias precisas, no ambiguas, que serán usadas para el desarrollo de la pieza de software (Loucopoulos, y KaraKostas, 1995).

El ciclo de vida de desarrollo de software se considera como una secuencia de procesos que contemplan una serie de pasos en donde se incluyen actividades, restricciones y recursos para producir una salida, es decir, un resultado esperado. Los procesos son importantes porque aportan consistencia y estructura sobre las actividades, manteniendo un nivel de calidad en los productos o servicios producidos (Pfleeger, 2002).

En un proyecto de software, se deben establecer previa y claramente su dominio de aplicación y sus objetivos. El analista de software y el interesado se reúnen para definir los objetivos del proyecto y su ámbito. Esta actividad es el inicio del proceso de la ingeniería que se le realiza al software futuro y se debe considerar

como el primer paso en el análisis de los requisitos del software. Una vez se han validado los objetivos del proyecto de software entre analista e interesado, se continúa con las diferentes alternativas de solución (Pressman, 1998). Los objetivos establecidos y validados entre analista e interesado permiten formular los requisitos que debe cumplir la pieza de Software.

2.2.1 Problemas asociados durante la Educción de Requisitos de Software

La Educción de Requisitos es una actividad de la Ingeniería de Requisitos que consiste en descubrir y adquirir el conocimiento necesario de las necesidades del interesado en relación con el problema o necesidad que se desea sistematizar. Una de las técnicas utilizadas en la educación y validación de requisitos son las entrevistas, en éstas, analista e interesado estudian conjuntamente los diferentes aspectos del área del dominio, algunos de estos aspectos son: (i) los objetivos del área del dominio, (ii) los procesos utilizados para lograr los objetivos, y (iii) los actores involucrados en los procesos.

La actividad de la Educción de Requisitos se basa generalmente en un diálogo realizado entre analista e interesado. En este diálogo, el analista es quien aporta su conocimiento para la elaboración de software y, el interesado, aporta su conocimiento sobre el área del dominio. Una vez finaliza el diálogo, queda como elemento de salida un texto en lenguaje natural que consigna el conocimiento del interesado, y un conjunto de modelos que representan la interpretación que realiza el analista de dicho conocimiento.

El interesado es experto en el dominio del problema (procesos, reglas del negocio, funciones, entre otros) y el analista es experto en modelos conceptuales propios de la Ingeniería del Software (Casos de uso, diagramas de clases, diagramas de secuencias, entre otros).

El analista debe realizar la identificación de requisitos y validarlos con el interesado una vez comprenda la naturaleza, características y límites del problema o la necesidad. Sin embargo, las diferencias que existen entre las especialidades de éstos, hace que dicha validación, por lo general, no sea interpretada de la misma manera entre las partes, lo que finalmente se traduce en la entrega de una pieza de software que, en la mayoría de los casos, no cumple con las necesidades que quiso plantear el interesado. Lo anterior obedece a la falta de medidas de verificación que puedan ser interpretadas del mismo modo entre el interesado y el analista.

A continuación, se relacionan algunos problemas vigentes durante la educación y validación de requisitos de software:

- El analista en la mayoría de los casos trata de interpretar el lenguaje utilizado por los interesados, pues, éstos manejan su propio lenguaje (frases y términos que son propios del área del dominio), además, la ambigüedad verbal que es muy común en el lenguaje natural, evita que el analista tenga mayor claridad de los requisitos.
- Por lo general, el interesado no está en capacidad de interpretar los modelos conceptuales propios de la Ingeniería del Software presentados por el analista, esta situación hace que el futuro sistema (software) pierda consistencia, calidad y no cumpla con los requisitos del interesado. Las figuras 2.4 y 2.5 ejemplifican la problemática vigente en la educación y validación de requisitos.

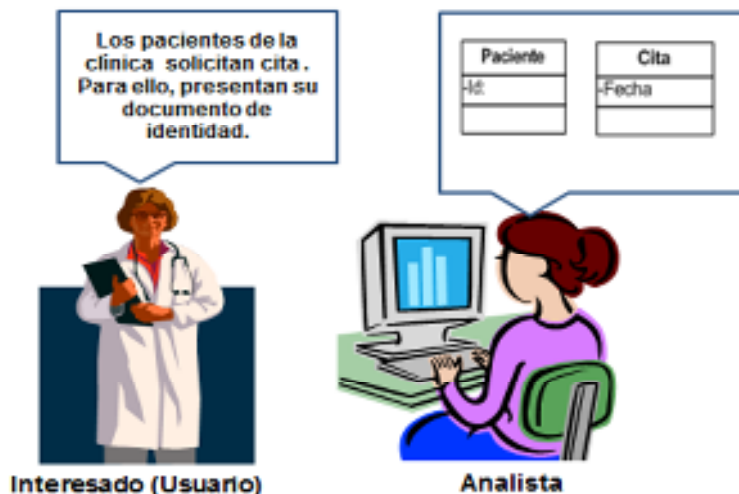


Figura 2.4. Formas de expresión de la realidad entre los usuarios y los analistas.

Fuente: Elaboración propia.

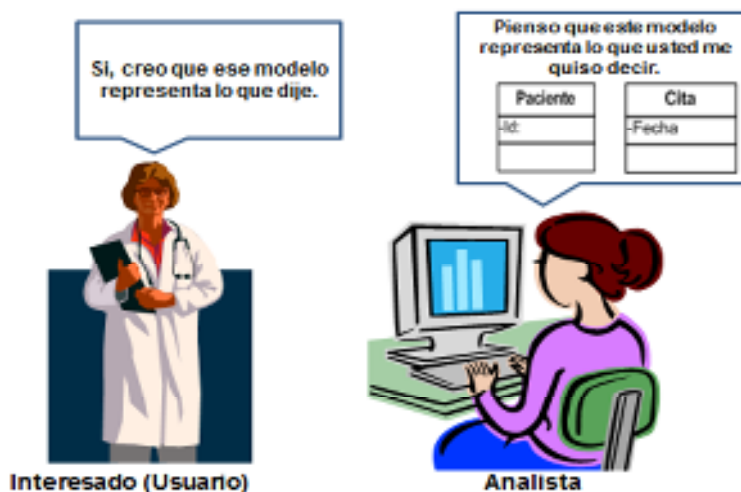


Figura 2.5. Incapacidad de entender el lenguaje del otro. Fuente: Elaboración propia.

2.2.2 Definición de objetivos para un sistema (software) futuro

En los proyectos que se formulan para las diferentes áreas del conocimiento, uno de los primeros elementos que se deben identificar son los objetivos, los cuales

permiten determinar los resultados que se deberán alcanzar, a partir de éstos, se identifican las metas (fines de corto alcance, consideradas como los pequeños logros que acercan al cumplimiento de un objetivo), los procesos, las tareas y los actores responsables. Además, con base en los objetivos planteados se puede dar seguimiento a las actividades definidas para su respectivo logro y, finalmente, evaluar los resultados obtenidos.

Diferentes autores (citados más adelante en esta sección) que realizan investigaciones en el área de Ingeniería de Requisitos, consideran importante disponer de una buena definición de objetivos para el sistema (software) futuro. Asimismo, indican que la definición de objetivos permite una comunicación más entendible entre analista y usuario durante el proceso de identificación y validación de requisitos.

Según (Byars, 1984), un objetivo es “un resultado que se desea lograr”; el análisis y la especificación de los objetivos pueden parecer una tarea relativamente sencilla, sin embargo, ésta requiere de un conocimiento básico que permita establecer con claridad los diferentes elementos que se articulan para alcanzar el resultado que se desea lograr.

Existen dos propuestas que durante el proceso de educación de requisitos de software utilizan metodologías orientadas a objetivos, éstas son: (i) I*, y (ii) KAOS.

En (Yu, 1995), se define i^* como un lenguaje orientado a objetivos, además, propone el uso de dos modelos, cada uno correspondiente a un nivel de abstracción: (i) Nivel intencional, representado por el Strategic Dependency Model (SD), consta de un conjunto de nodos que representan actores y las relaciones entre éstos, (ii) Nivel racional representado por el Strategic Rationale Model (SR). Permite visualizar objetivos, tareas, recursos, entre otros, dentro de los límites de un actor.

En i^* se utiliza un diagrama de objetivos que no representa en su elaboración la jerarquía entre objetivos, la cual es necesaria para determinar el objetivo de alto nivel y los demás objetivos que lo subrogan hasta identificar los requisitos que deberá cumplir el sistema (software). Además, en esta propuesta, el diagrama de objetivos lo construye el analista a partir de la información entregada por el interesado, es decir, no existe una interacción permanente del interesado ni un proceso terminológico durante este proceso. En la figura 2.6 se puede observar el diagrama de objetivos utilizado en i^* .

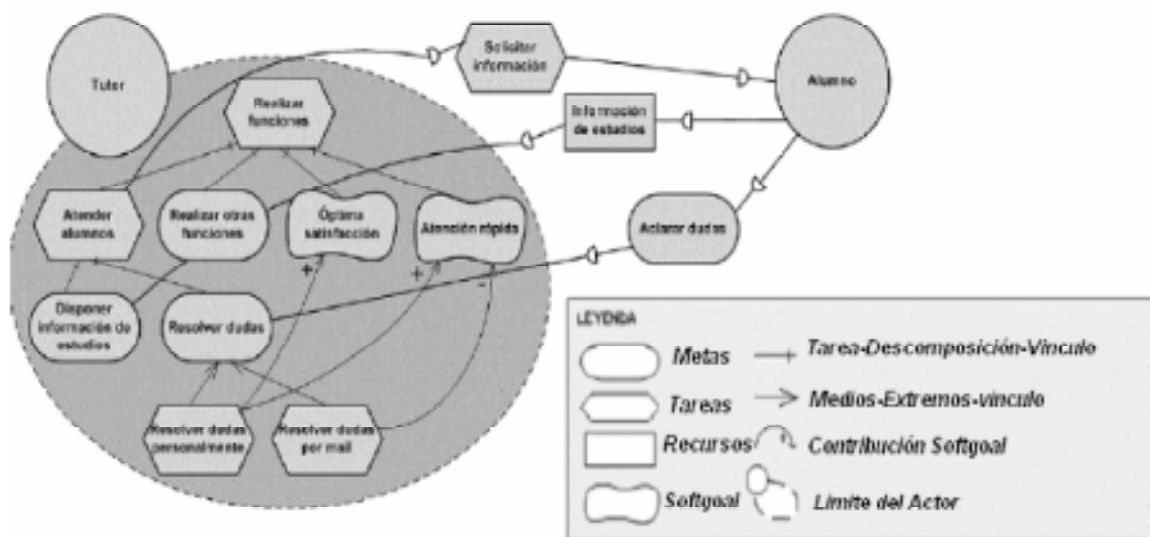


Figura 2.6. Ejemplo de diagrama de objetivos de i^* . Fuente: Adaptada de: (Yu, 1995).

Según (Lezcano, 2007), en la figura 2.6 se logran observar algunas confusiones con relación al uso de los elementos que conforman el diagrama. Por ejemplo, “resolver dudas” y “realizar funciones”, figuran simultáneamente como objetivo y como tarea. Dado lo anterior, se genera un contrasentido, puesto que, las tareas definen cómo se realiza un proceso y el objetivo describe el para qué se realiza. Asimismo, en esta propuesta no se logran identificar estructuras lingüísticas que

permitan: (i) caracterizar los elementos que conforman el diagrama de objetivos, y (ii) minimizar la ambigüedad semántica de los elementos que conforman dicho diagrama.

La segunda propuesta la presenta (Lamsweerde, 1993, 2000), en ésta, se propone la búsqueda de palabras clave en los documentos de información disponibles para la identificación de objetivos. Además, establece la representación de metas, objetos, agentes y operacionalización de objetivos en requisitos. Asimismo, propone una representación del diagrama de objetivos, que se basa en la metodología KAOS (Knowledge Acquisition in Automated Specification). En este diagrama, se parte de los objetivos de más alto nivel de la organización, los cuales se van subrogando en otros objetivos hasta alcanzar los requisitos que deberá cumplir una aplicación de software para satisfacer los objetivos organizacionales. El proceso para trazar el diagrama de objetivos de KAOS, exige la definición de objetivos secundarios, los cuales subrogan los objetivos generales, para presentar finalmente los objetivos más elementales o atómicos. Los elementos básicos que conforman el diagrama de objetivos de la metodología KAOS se pueden observar en la figura 2.7. Asimismo, se puede observar la jerarquización de este diagrama en donde se logra determinar el “cómo” de un objetivo desplazándose hacia abajo en el diagrama y la explicación del “por qué” desplazándose hacia arriba en el diagrama (Letier 2001). Dicha figura ilustra el modelado de un proceso para el servicio de ambulancia.

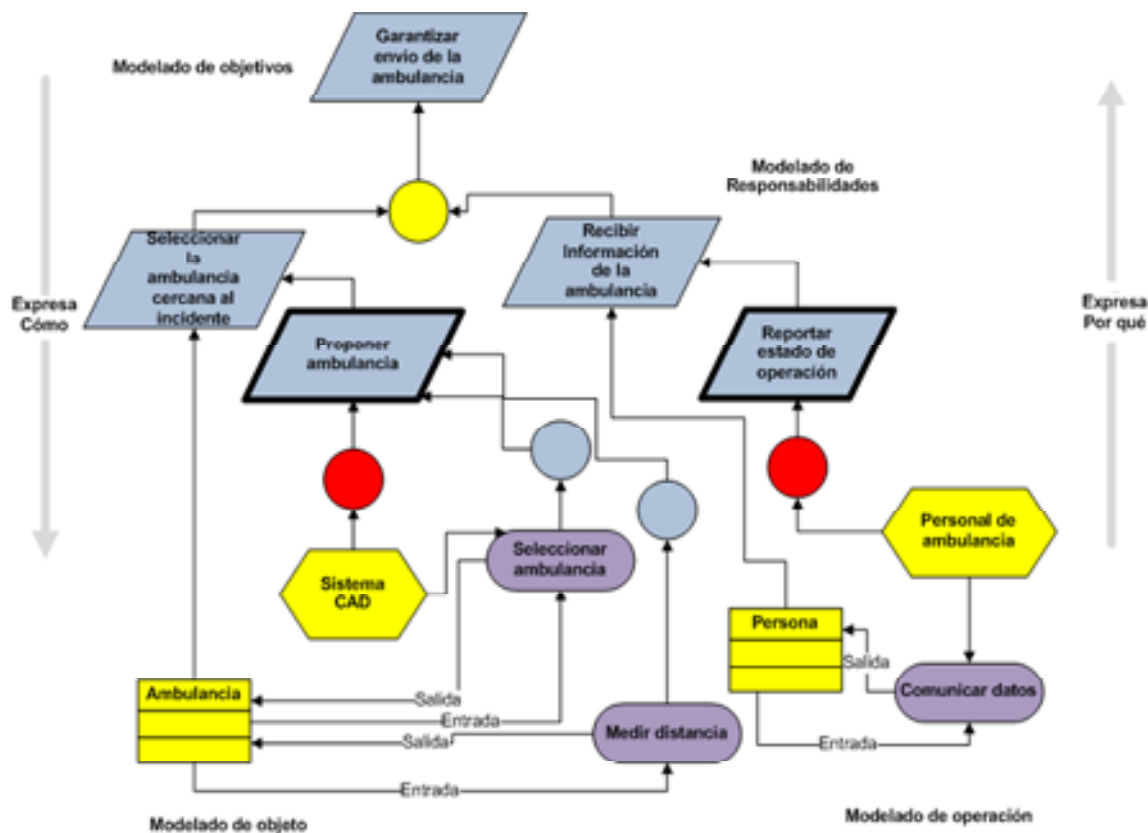


Figura 2.7. Elementos básicos del diagrama de objetivos de KAOS. Fuente: Adaptado de (Respect IT, 2007).

La explicación de los elementos que conforman la figura 2.7, se presenta a continuación:

Objetivo: Resultado que se espera lograr con la realización de procesos o actividades. Su representación gráfica es mediante un paralelogramo con fondo de color azul.

Relación AND: permite subrogar los objetivos mediante la relación lógica AND. Su representación gráfica es mediante un círculo sombreado de color amarillo.

Requisito: es un objetivo de bajo nivel, alcanzable por un agente. Se representa con la figura de paralelogramos con sus bordes resaltados y fondo del paralelogramo azul.

Agente: puede ser un componente humano o automático (software), responsable de un requisito o una expectativa. Los agentes se representan con cajas amarillas con ángulos en las esquinas.

Relación de responsabilidad: conecta un agente a un requisito que está bajo su responsabilidad. Su representación gráfica es un círculo sombreado de color rojo.

Relación de operacionalización: conecta la operación con un requisito. Su representación es un círculo sombreado de color azul.

Operación: elemento encargado de operacionalizar los requisitos. Su representación gráfica es un ovalo sombreado de color morado.

Entidad: se utiliza para representar objetos de manera independiente. Su representación gráfica es un rectángulo sombreado de color amarillo.

En la sección 3.2 de esta tesis, se amplía la definición de los demás elementos que conforman la metodología KAOS.

Según (Thomas y Oliveros, 2005), la metodología KAOS no dispone de un proceso lo suficientemente claro que permita obtener el modelo de objetivos. Asimismo, (Zapata, Lezcano, y Tamayo, 2011), indican que, actualmente, no es posible obtener los elementos que conforman el diagrama de objetivos de KAOS a partir del lenguaje natural, o de una representación de éste. Finalmente, en esta propuesta no se logra identificar un procedimiento que permita minimizar la ambigüedad semántica de los elementos que conforman diagrama de objetivos de KAOS.

Por otra parte, (Zapata, Villegas, y Arango, 2006), utilizan el diagrama de objetivos durante el proceso de desarrollo de software, aunque no lo enfocan

directamente para validar con el interesado los requisitos que deberá cumplir el sistema (software). Además, durante el proceso utilizan el lenguaje controlado (UN-Lencep) para identificar los requisitos del dominio del problema. Sostienen que, “con este lenguaje controlado, se puede obtener una especificación de UN-Lencep básico, que corresponde de manera inambigua a un Esquema Preconceptual único, con el cual se pueden generar tres diagramas UML: Clases, Comunicación y Máquina de Estados”. En la tabla 2.1 se puede observar un fragmento del lenguaje controlado utilizado en esta propuesta, el cual se representa a través de triadas (concepto, verbo, concepto).

Tabla 2.1. Fragmento lenguaje controlado UN_Lencep. Fuente: (Zapata, Gelbukh, y Arango, 2006c)

Concepto	Verbo	Concepto
Estudiante	es	persona
Profesor	es	persona
Profesor	tiene	curso
Curso	tiene	estudiante

Las dificultades que presenta el lenguaje utilizado en esta propuesta son: (i) para su utilización, se requiere entrenar previamente al interesado, y (ii) al ser un lenguaje restringido o controlado, minimiza ostensiblemente la expresividad del interesado, situación que podría impedir que éste pueda formular todas sus necesidades y/o requisitos relacionados con el área del problema. Además, no dispone de un proceso lo suficientemente claro que permita obtener el diagrama de objetivos, prueba de ello, es que el diagrama de objetivos obtenido no es consistente con el universo del discurso presentado en lenguaje restringido. Además, no se logra identificar un procedimiento que permita minimizar la ambigüedad semántica de los elementos que conforman diagrama de objetivos de KAOS.

Asimismo, (Zapata, et al., 2011), presentan una propuesta para obtener el diagrama de objetivos KAOS a partir de los esquemas preconceptuales presentados en (Zapata et al., 2006). En la figuras 2.8 se pueden observar el

esquema preconceptual obtenido para el caso de estudio de SoftCom propuesto por (Rumbaugh, Blaha, Premerlany, Eddy, y Lorensen, 1991).

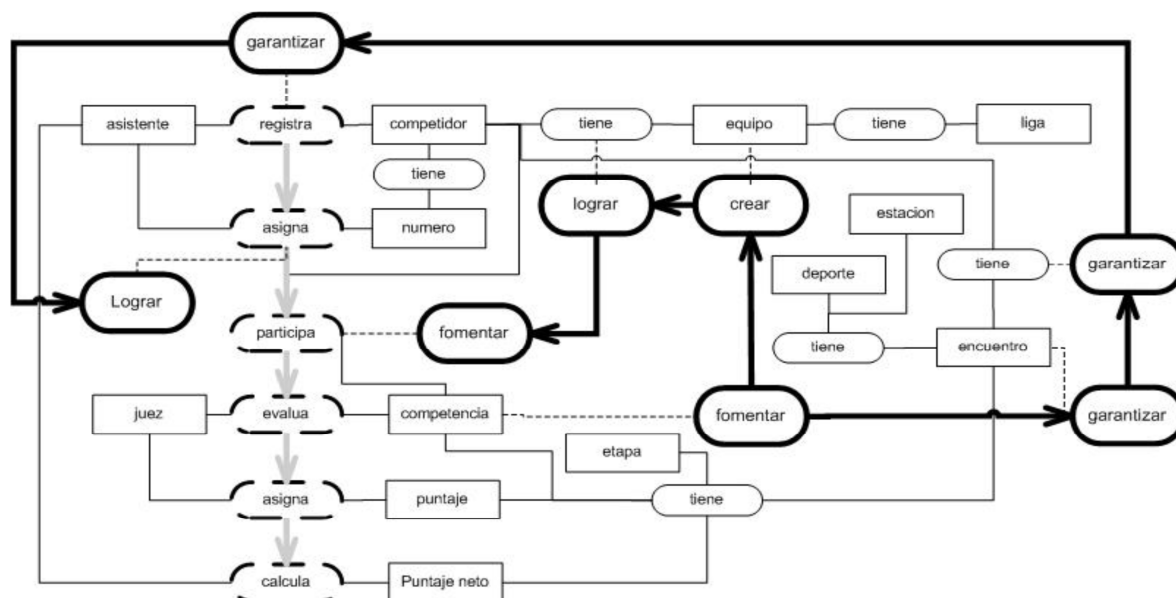


Figura 2.8. Esquema preconceptual caso Softcom. Fuente: (Zapata, et al., 2011)

Las principales falencias encontradas en este trabajo son: (i) el esquema preconceptual se obtiene a partir del lenguaje controlado UN-Lencep, el cual minimiza ostensiblemente la expresividad del interesado, y (ii) no identifica los elementos básicos (entidad, atributos, operaciones, entre otros) del diagrama de objetivos de KAOS. Además, no se logra identificar un procedimiento que permita minimizar la ambigüedad semántica de los elementos que conforman diagrama de objetivos de KAOS.

(Zapata y Vargas, 2011), presentan algunas estructuras lingüísticas para caracterizar objetivos, las cuales presentan un alto grado de restricción. En la tabla 2.2 se pueden observar dichas estructuras.

Tabla 2.2. Estructuras lingüísticas para caracterizar objetivos. Fuente: (Zapata y Vargas, 2011).

Descripción	Restricciones
O→V1+Ad+SN	V1→{Verbo de logro}; Ad→ {Connotación positiva. Por ejemplo: “Alta”; “buena”, “Adecuada”}
O→V1+C+V2+ SN+Adv	V1→{Verbo de logro}; C→ {“que”}; V2 → {forma reflexiva impersonal o voz pasiva refleja; se sugieren verbos como: “presentar”, “cometer”, “realizar”, “generar”, “utilizar”.}; Adv→ {Connotación positiva, ejemplos: “a tiempo”, “correctamente”, etc.}
O→V1+C+V2+ SN	V1→{Verbo de logro}; C→ {“que”}; V2 → {Verbo conjugado}
O→V+SN	V → {Verbo de Mejoramiento con connotación negativa, por ejemplo “reducir”, “disminuir”, “decrementar”, etc.}; SN → {Connotación negativa; por ejemplo: “demora”, “retraso”, “deserción”, “accidentalidad”, etc.}
O→V1+C+SN1+ V2+SN2+Adv	V1→{Verbo de logro}; C→ {“que”}; V2→ {verbo conjugado}; Adv→ {Connotación positiva; se sugieren palabras como: “a tiempo”, “correctamente”, etc.}
O→V1+C+SN+ V2+Ad+SN	V1→{Verbo de logro}; C→ {“que”}; V2→ {verbo conjugado}; Ad→{Connotación positiva; por ejemplo: “alto”, “bien”, etc.}
O→V1+C+SN+ Adv+V2+SN	V1→{Verbo de logro}; C→ {conjunción “que”}; Adv→{“no”}; V2→{Verbo conjugado que exprese una connotación negativa; por ejemplo: “subvalorar”, “perder”, “fallar”, etc.}
O→V1+C+SN1+Adv+V2+S N2	V1→{Verbo de logro}; C→ {“que”}; Adv→{“no”} V2 → {Verbo conjugado}; SN2 → {Connotación negativa; por ejemplo: “demoras”, “retrasos”, “errores”, etc.}
O→V1+C+V2+Ad+SN	V1→{Verbo de logro}; C→ {“que”}; V2 → {“haber” o “existir” en forma conjugada.}; Ad → {Connotación positiva; por ejemplo: “mucho”, “demasiado”, etc.} SN → {Connotación positiva}

Las siguientes son las falencias encontradas en esta propuesta: (i) la estructura lingüística para identificar objetivos presenta un alto grado de restricción, y (ii) no presenta otras estructuras lingüísticas que permitan caracterizar los demás elementos (entidad, operaciones, agentes, entre otros) que conforman el diagrama de objetivos de KAOS. Asimismo, no se logra identificar un procedimiento que permita minimizar la ambigüedad semántica de los elementos que conforman diagrama de objetivos de KAOS.

(Lezcano, Herrera y Londoño, 2013), presentan un método para realizar la educación de requisitos de software, en éste, utilizan el diagrama causa efecto y el diagrama de objetivos de KAOS, además, elaboran una plantilla para obtener el universo del discurso (ver tabla 2.3), y definen las siguientes reglas gramaticales para obtener algunos elementos (objetivo, requisito y actor) del diagrama de objetivos de KAOS:

RO1: Sujeto + perífrasis de obligación + infinitivo + complemento

RO2: Sujeto + verbo + complemento

Con la primera regla (RO1), intentan caracterizar el objetivo de alto nivel concerniente al diagrama de objetivo de KAOS y con la segunda regla (RO2), intentan caracterizar el requisito y el agente perteneciente al mismo diagrama.

Las falencias encontradas en esta propuesta son: (i) la estructura lingüística utilizada para identificar el objetivo, el requisito y el agente es bastante incipiente, y (ii) no presenta otras estructuras lingüísticas que permitan caracterizar los demás elementos (entidad, atributos, operaciones, entre otros) que conforman el diagrama de objetivos de KAOS. Además, no se logra identificar un procedimiento que permita minimizar la ambigüedad semántica de los elementos que conforman diagrama de objetivos de KAOS.

Tabla 2.3. Plantilla de entrevistas. Fuente: (Lezcano et al., 2013)

ENTREVISTA ANALISTA – USUARIO	
Objetivo: esta entrevista tiene como propósito lograr que el interesado exprese con sus propias palabras las necesidades y/o problemas que tiene la organización, de tal forma, que permita entregar los primeros insumos para capturar los requisitos.	
Nombre del entrevistado (nombre completo)	
Cargo (cargo que desempeña el usuario)	
Nombre de la empresa (nombre completo de la empresa)	
Misión (colocar la misión de la empresa)	
Visión (colocar la visión de la empresa)	
Actividad económica (a que se dedica la empresa)	
Área del problema (describa el área en el que se presenta el problema)	
Funciones del área del problema (colocar cada una de las actividades que se desarrollan en el área del problema)	
Actores (las personas que intervienen en el área del problema con sus respectivas funciones)	
Problema (describir con claridad cada uno de los problemas que se están presentando en el área)	
Terceros (coloque si estos problemas afectan otras áreas de la empresa)	

(Guzmán, Lezcano y Gómez, 2013), presentan una propuesta para caracterizar algunos de los elementos (objetivo, objetivo subrogado, requisito y agente) que conforman el diagrama de objetivos de KAOS, para ello, utilizan los siguientes elementos gramaticales: sujeto, perífrasis con infinitivo mediato, verbo y complemento. Asimismo, definen las siguientes reglas gramaticales:

R1: Sujeto+ Perífrasis con infinitivo mediato + complemento + verbo en infinitivo + complemento.

R2: Sujeto + Perífrasis con infinitivo mediato + complemento.

R3: Sujeto+ verbo+ objeto (o) sujeto + complemento.

La regla (R1), la utilizan para caracterizar a partir de lenguaje natural, el objetivo de alto nivel propio del diagrama de objetivos de KAOS.

La regla (R2), es utilizada para caracterizar el subobjetivo u objetivo subrogado del diagrama de objetivo de KAOS.

La regla (R3), es utilizada para caracterizar el requisito y el agente ambos elementos propios del diagrama de objetivo de KAOS.

Las siguientes son las falencias encontradas en esta propuesta: (i) la estructura lingüística utilizada para identificar el objetivo, el requisito y el agente es bastante incipiente, y (ii) no presenta otras estructuras lingüísticas que permitan caracterizar los demás elementos (entidad, atributos, operaciones, entre otros) que conforman el diagrama de objetivos de KAOS. Asimismo, no se logra identificar en esta propuesta un procedimiento que permita minimizar la ambigüedad semántica de los elementos que conforman diagrama de objetivos de KAOS.

(Antón, 1996, 1997, 1998), plantea el método GBRAM (Goal-Based Requirements Analysis Method), el cual está basado en heurísticas para identificar, clasificar, refinar y elaborar: objetivos, interesados, agentes y restricciones. Con dicho método, propone atacar los desacuerdos entre el interesado y el analista originados durante el proceso de educación y especificación de requisitos de software. Para resolver estos problemas, propone la utilización de un procedimiento que permita identificar, clasificar y refinar los objetivos que la aplicación de software debe alcanzar, con la subsiguiente transformación de objetivos en requisitos operacionales. En la figura 2.9, se observan las actividades que debe realizar el analista al emplear esta propuesta, utilizando el método GBRAM (*Goal-Based Requirements Analysis Method*), las cuales se integran en un modelo de: Entrada, Proceso y Salida, en donde el Proceso se divide en Análisis y Refinamiento. Además, presenta un conjunto de nueve verbos para la

identificación y/o definición de objetivos de software, los cuales clasifica en tres tipos: (i) mantenimiento, (ii) mejoramiento, y (iii) logro. Ver tabla 2.4.

Tabla 2.4. Verbos presentados por Antón. Fuente: (Antón, 1998).

Objetivo	Tipo de objetivo
Evitar información obsoleta	Mantenimiento
Garantizar transacciones seguras	Mantenimiento
Mantener solicitudes de participación	Mantenimiento
Mantener dos servidores	Mantenimiento
Mejorar el mantenimiento y la administración de contenido	Mejoramiento
Reducir el tiempo necesario para lograr la aprobación de modificación de las actualizaciones.	Mejoramiento
Incrementar las ganancias de seminarios	Mejoramiento
Conocer los privilegios de acceso de los socios	Logro
Hacer que el socio se registre	Logro

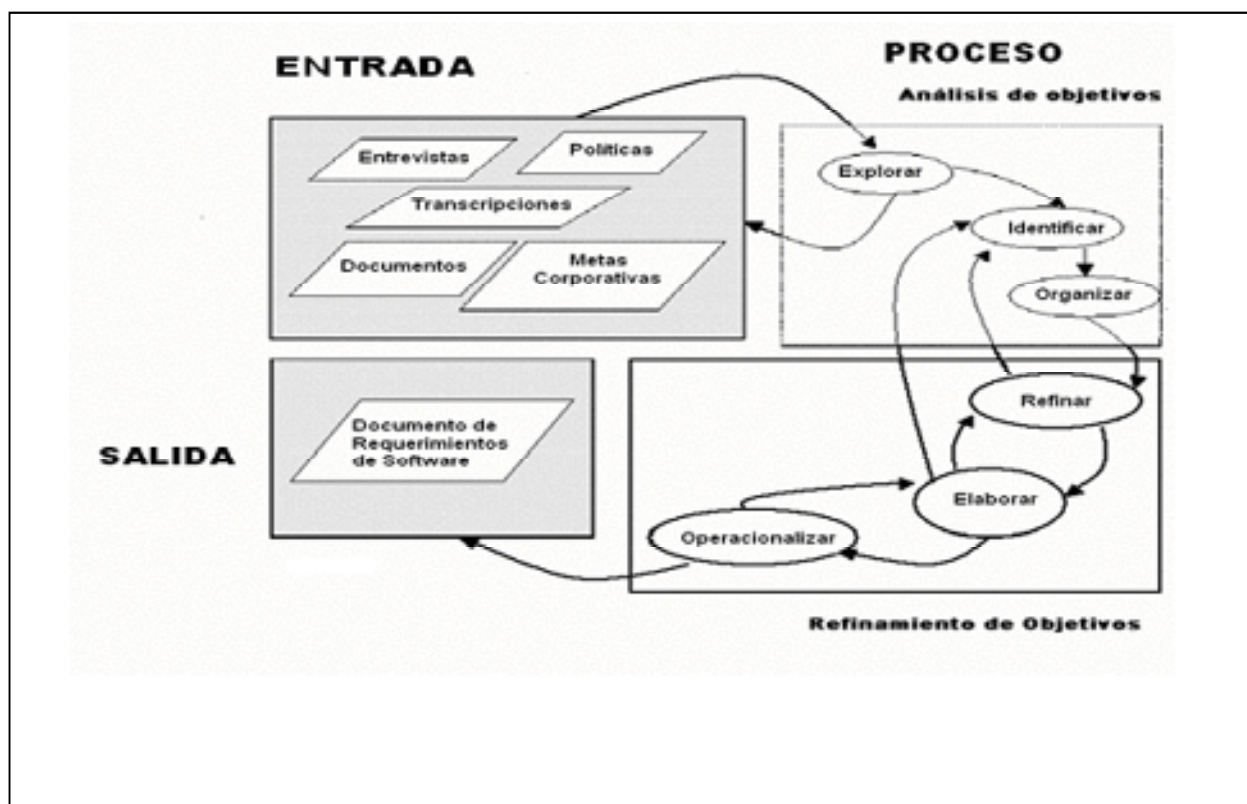


Figura 2.9. Resumen de las actividades GBRAM. Fuente: (Antón, 1997).

(Thomas y Oliveros, 2005), indican que en GBRAM, el analista está involucrado en las siguientes actividades:

- Identificar objetivos: explora la documentación existente, identifica los interesados e identifica agentes y sus responsabilidades.
- Organizar y clasificar objetivos: elimina redundancias, clasifica los objetivos, especifica dependencias entre los objetivos y finalmente construye una jerarquía de objetivos.
- Refinar y elaborar objetivos: especifica los obstáculos para cada objetivo, construye escenarios e identifica restricciones.
- Determinar las operaciones que satisfacen los objetivos y luego las transforma en requisitos: traduce los objetivos en requisitos operacionales, utilizando una sintaxis de representación que se puede observar en la Tabla 2.5.

Según (Thomas y Oliveros, 2005), algunas de las principales falencias encontradas en este trabajo son: (i) provee indicaciones limitadas sin criterio metodológico, (ii) no dispone de pautas precisas para obtener el modelo de objetivos. Además, no se logran identificar en esta propuesta estructuras lingüísticas que permitan caracterizar los elementos que conforman el diagrama de objetivos. Asimismo, no se logra identificar un procedimiento que permita minimizar la ambigüedad semántica de los elementos que conforman diagrama de objetivos.

Tabla 2.5. Sintaxis de representación de elementos en GBRAM. Fuente: (Antón, 1998).

Componente	Descripción
Nombre del Objetivo	Es el identificador único para cada objetivo. Indica un estado deseado o alcanzado.
Tipo	Los objetivos se clasifican de acuerdo con el comportamiento requerido: alcanzar algún estado (“Achievement”), mantener (“Maintenance”) o mejorar (“Improvement”) alguna condición o estado.
Descripción	Es un texto informal que describe una meta u objetivo.
Acción	Es el nombre que se le otorga a la operacionalización de una meta u objetivo. Describe el comportamiento necesario para satisfacer el objetivo.
Agente	Es el responsable de completar o cumplir un objetivo.
Interesados	Son las personas interesadas en que una meta u objetivo sea cumplido.
Restricciones	Son las limitaciones bajo las cuales un objetivo se debe cumplir. Una restricción, especifica algún requisito o condición que se debe cumplir para lograr un objetivo. Tiene vigencia hasta que se logre el objetivo.
Obstáculos	Representan las circunstancias que puedan impedir que un objetivo sea cumplido.
Precondiciones	Son condiciones que deben existir para posibilitar el logro de un objetivo.
Postcondiciones	Son condiciones a las que se arriba luego de obtener o completar un objetivo.
Submetas	Cada submeta debe mapear a una acción. Si mapea a varias acciones, se debería descomponer y refinar.

(Mylopoulos, Chung, Liao, Wang, Yu, 2001), señalan que el análisis de requisitos orientado por objetivos permite complementar y fortalecer las técnicas utilizadas tradicionalmente durante el proceso de educación y validación de requisitos. Además, presentan una propuesta orientada a objetivos que consta de las siguientes entradas y salidas:

Entrada. Consta de un conjunto de metas u objetivos funcionales, a los cuales les aplican el siguiente procedimiento: (i) descomponen cada meta u objetivo en una jerarquía, (ii) descomponen cada característica de calidad en una jerarquía de Softgoals, (iii) identifican correlaciones entre los Softgoals, (iv) identifican correlaciones entre los Goals y los Softgoals, y (v) seleccionan los Goals y Softgoals que permitan satisfacer las metas funcionales.

Salida. Se compone de un conjunto de funciones que deberán ser ejecutadas por el sistema (software) para cumplir con las metas funcionales.

De acuerdo con (Thomas y Oliveros, 2005), esta propuesta presenta las siguientes falencias: (i) sólo presenta indicaciones generales; (ii) no dispone de pautas precisas para obtener el modelo de objetivos. Además, no presenta estructuras lingüísticas que permitan caracterizar los elementos que conforman el diagrama de objetivos. Asimismo, no se logra identificar en esta propuesta un procedimiento que permita minimizar la ambigüedad semántica de los elementos que conforman diagrama de objetivos de KAOS.

(Oshiro, Watahiki y Saeki, 2003), presentan un método orientado a objetivos, en éste, el usuario participa en las reuniones con los actores involucrados en el desarrollo del proyecto, además, se nombra un analista experimentado para controlar el avance de las reuniones. Este método utiliza los siguientes pasos: (i) incluye las necesidades de los usuarios a modo de objetivos iniciales, (ii) selecciona un objetivo para ser descompuesto y refinado, (iii) genera ideas para ser vinculadas al objetivo elegido, (iv) agrupa las ideas formadas, (v) busca relaciones entre las agrupaciones de ideas, y (vi) agrega las agrupaciones de ideas con las asociaciones al grafo que representa los objetivos.

Sin embargo, en esta propuesta no se logra: (i) identificar estructuras lingüísticas que permitan caracterizar los elementos que conforman el diagrama de objetivos, y (ii) identificar un procedimiento que permita minimizar la ambigüedad semántica de los elementos que conforman diagrama de objetivos.

(Thomas y Oliveros, 2005), establecen las técnicas utilizadas en los escenarios (Leite et al., 1997, Leite et al., 2000) como insumo para la educación de objetivos. La técnica consiste en emplear un conjunto de reglas sobre los componentes que pertenecen a un escenario, para posteriormente obtener los objetivos que

corresponden al escenario seleccionado. Las reglas utilizadas en esta propuesta, se logran agrupar en los siguientes tipos:

- (i) Reglas para identificar escenarios: consisten en formar el nombre del objetivo, a través del estado alcanzado al cumplirse un escenario. Por ejemplo: en el escenario “Cancelación de la reserva”, se obtiene el objetivo: “reserva cancelada”.
- (ii) Reglas para los componentes de escenarios: utiliza los elementos de un escenario para transformarlos en los elementos (descripción de los objetivos, precondiciones, interesados y agentes) presentados en las plantillas definidas en (Antón, 1998).
- (iii) Reglas para los episodios de los escenarios: con la aplicación de las reglas asociadas a episodios obtiene submetas y postcondiciones.
- (iv) Reglas para bifurcaciones de escenarios: permiten formar los obstáculos o las restricciones de un objetivo.

En esta propuesta no se logran identificar estructuras lingüísticas que permitan caracterizar los elementos que conforman el diagrama de objetivos. Las reglas están orientadas en transformar los escenarios en objetivos que no expresan metas u objetivos a cumplir, sólo estados ya ejecutados (p.e. Escenario “Cancelación de la reserva”, objetivo: “reserva cancelada”). Finalmente, no se logra identificar un procedimiento que permita minimizar la ambigüedad semántica de los elementos que conforman diagrama de objetivos de KAOS.

(Nicolás y Toval, 2009), presentan una revisión de la literatura relacionada con la generación de especificaciones de requisitos textuales en la Ingeniería de Software, en ésta, logran demostrar que las especificaciones textuales de

requisitos de software, es un tema interés que amerita continuidad en los trabajos de investigación que realiza la comunidad científica en la Ingeniería de Requisitos. Algunos de los trabajos orientados a objetivos analizados en esta revisión fueron los presentados por: (Alrajeh, Russo y Uchitel, 2006), (Maiden, Manning, Jones y Greenwood, 2005), (Lamsweerde y Willemet, 1998), (Letier y Lamsweerde, 2002), (De Landtsheer, Letier y Lamsweerde, 2004), (Lamsweerd, 2004), (Yu, Dubois, Dubois, y Mylopoulos, 1995), (Antón y Potts, 1998).

Por otra parte, (Loucopoulos y Karakostas, 1995), indican que las especificaciones textuales de requisitos de software son presentados comúnmente en Lenguaje Natural (LN), además, sostienen que el LN es uno de los lenguajes más utilizados para la educación de requisitos de software, puesto que, su mayor atractivo radica en su vocabulario, su informalidad y su sintaxis. Es un lenguaje familiar para el interesado y el analista, además, no requiere tiempo de aprendizaje. Este lenguaje se divide en dos categorías: (i) interacción directa con el usuario utilizando lenguaje natural verbal, y (ii) educación de requisitos desde un documento escrito en lenguaje natural. Esta técnica presenta dos tipos de problemas al momento de validar entre el analista y el interesado los requisitos que deberá cumplir la pieza de software futuro, el lenguaje natural es: (i) complejo, y (ii) ambiguo.

Una síntesis de la información relacionada con la revisión de la literatura de algunos de los autores que realizan investigaciones en Ingeniería de Software se compendian en la tabla 2.6. Los criterios que se analizan para los trabajos presentados en este capítulo son los siguientes: (i) utiliza algún tipo de método orientado a objetivos, (ii) presenta estructura lingüística (sintáctica, semántica o una combinación de éstas) para identificar objetivos, (iii) plantea algún tipo estructura lingüística (sintáctica, semántica o una combinación de éstas) para identificar entidades, (iv) presenta estructura lingüística (sintáctica, semántica o una combinación de éstas) para identificar operaciones, (v) plantea algún tipo estructura lingüística (sintáctica, semántica o una combinación de éstas) para

[illegible]

2.3 Conclusiones del capítulo

En este capítulo se realizó una revisión de la literatura de las principales metodologías y técnicas utilizadas en la Ingeniería de Software para la educación de requisitos orientada a objetivos.

La revisión previa permitió identificar el crecimiento y la importancia de la Ingeniería de Requisitos orientada a objetivos. Asimismo, permitió esclarecer que los trabajos relacionados presentan diferentes limitaciones (estructura lingüística incipiente, ambigüedad semántica, falta de caracterización de los elementos básicos del diagrama de objetivos, problemas de comunicación entre usuario y analista, entre otras). Asimismo, se presentaron diversas técnicas de desambiguación semántica de tipo polisémica aplicadas a la Lingüística Computacional y que pueden ser usadas en la Ingeniería de Software. Se hizo énfasis en las técnicas del sentido más frecuente (MFS), Lesk y UKB, debido a que éstas son utilizadas con mayor frecuencia a nivel mundial, además, se pueden personalizar fácilmente y aplicar al campo de la Ingeniería de Software, estas serán las técnicas que se implantarán durante el desarrollo de esta tesis. Igualmente, se logró determinar que actualmente no existe una aproximación que pueda resolver las limitaciones identificadas de forma conjunta, lo que deja abierto espacios que son atacados en esta tesis doctoral.

Dado lo anterior, se propone en esta tesis: (i) definir un proceso que permita obtener el modelo de objetivos, (ii) plantear una estructura lingüística menos incipiente que la registrada en la literatura, para identificar los elementos básicos (objetivo, entidad, operación y agente) del diagrama de objetivos de KAOS, (iii) definir un procedimiento que permita minimizar la ambigüedad semántica de tipo polisémica de los elementos básicos que conforman diagrama de objetivos de KAOS, y (iv) obtener los elementos básicos del diagrama de objetivos de KAOS a partir del lenguaje natural escrito (que no sea tan controlado como los es UN_lencep).

La naturaleza jerárquica del diagrama de objetivos de KAOS, la manera de ligar los objetivos con los requisitos de los interesados, la posibilidad que ofrece de validar entre analista e interesado los requisitos del sistema (software) futuro y la obtención de artefactos UML propios de la tercera fase (diseño) del ciclo de vida de software, son características deseables durante el proceso de educación de requisitos de software, es por ello que, en esta tesis doctoral, se propone una forma de obtener los elementos básicos que conforman dicho diagrama, de tal manera que: (i) se aproxime al discurso del interesado, y (ii) permita la interacción del usuario durante todo el proceso.

Capítulo 3

Caracterización semántica de los elementos que conforman el diagrama de objetivos de KAOS

En este capítulo se identifican, describen y se caracterizan semánticamente los elementos propuestos por la metodología KAOS con el fin de que sean comprensibles por una máquina sin ambigüedades. Para tal fin, se toma como modelo de referencia el metamodelo propuesto por (Matulevičius, et al., 2007) y, a partir de éste, se propone una Ontología que caracteriza este metamodelo que abarca los elementos del diagrama de objetivos KAOS. El desarrollo de este capítulo permite el cumplimiento del primer objetivo propuesto en esta tesis, es decir, caracterizar semánticamente los elementos que componen la definición de requisitos de software bajo el paradigma del diagrama de objetivos de KAOS para que sean comprensibles por una máquina sin ambigüedades.

3.1 Metodología KAOS

KAOS es una metodología utilizada en la Ingeniería de Requisitos que permite al analista construir modelos de requisitos. Además, facilita la identificación de propiedades y diseños alternativos. Este enfoque puede ser usado para cualquier sistema de información (Respect IT, 2007). La ventaja principal de esta metodología, radica en la capacidad de alinear los requisitos con los objetivos y las metas organizacionales, aumentando las posibilidades de nuevos desarrollos

que pueden generar valor agregado a las dinámicas del negocio. Asimismo, ofrece la posibilidad de validar entre analista e interesado los requisitos del sistema (software) futuro, además, permite la obtención de artefactos UML propios de la fase de diseño (Almisned y Keppens, 2010).

En esencia, la metodología KAOS es un enfoque de la Ingeniería de Requisitos orientada a objetivos. (Lamsweerde y Letier, 2004), presentan en este enfoque los siguientes niveles de expresividad y razonamiento: (i) semiformal. Utilizado para modelar y estructurar objetivos, (ii) cualitativo. Permite seleccionar entre diferentes alternativas, y (iii) formal. Utilizado (si es necesario) para un razonamiento más preciso. Cada constructo en el lenguaje de modelado de KAOS tiene una estructura de dos niveles: la capa exterior semántica y gráfica donde el concepto se declara junto con sus atributos y relaciones con otros conceptos, y la capa interna para la definición formal del concepto (Lapouchnian, 2005).

(Matulevičius et al., 2007), proponen el metamodelo de KAOS siguiendo el enfoque UEML (Unified Enterprise Modelling Language), este metamodelo, representa los constructos correspondientes al diagrama de objetivos de KAOS (ver figura 3.1). Al final de este capítulo se explicarán cuáles constructos fueron seleccionados para el desarrollo de esta tesis y por qué.

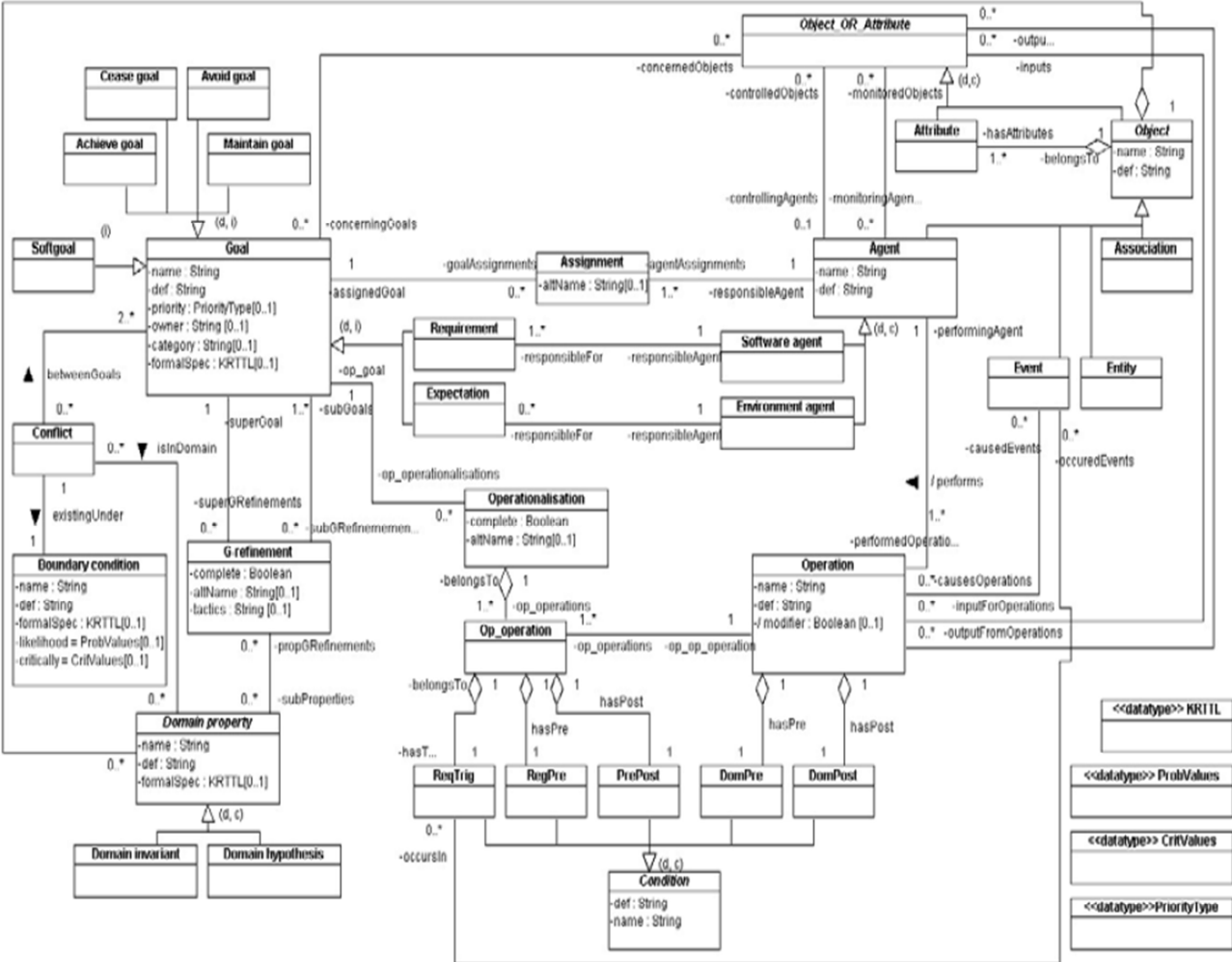


Figura 3.1. Metamodelo KAOS. Fuente: (Matulevičius et al., 2007).

3.2 Caracterización conceptual de los elementos del modelo de KAOS

En esta sección se presentan conceptualmente los elementos que conforman la metodología KAOS. Dicha conceptualización se realizó con base en el trabajo presentado por (Letier, 2001). Cabe aclarar que, estas definiciones harán parte del marco de trabajo propuesto en esta tesis.

La siguiente es la convención de los elementos utilizados para la definición formal de los componentes de la metodología KAOS que se presentan en esta sección.

\Diamond : algún tiempo en el futuro

$\Diamond_{\leq d}$: algún tiempo en el futuro dentro de un plazo d

\Box : siempre en el futuro

\Rightarrow : implica

\neg : negación

\neq : Diferente

\emptyset : vacío

P: objetivo en su estado inicial

Q: objetivo en su estado final

3.2.1 Objetivo

Resultado que se espera lograr con la realización de procesos o actividades. Es una afirmación prescriptiva, conjunto de exigencias y/o de comportamientos deseables que un sistema esperado debe cumplir. Algunas de las propiedades que especifican un objetivo son: nombre, definición, categoría, entre otros (Matulevičius et al., 2007). Asimismo, para determinar el cumplimiento de un objetivo, se requiere de un indicador (cualitativo o cuantitativo) que permita realizar su respectivo seguimiento.

Entidad Objetivo

Tiene nombre, definición, tipo, categoría, prioridad, especFormal, indicador, satisface, subObjs: ListaDe(Objetivo)

Invariable $\text{nombre} \neq \emptyset \wedge \text{definición} \neq \emptyset \wedge \text{indicador} \neq \emptyset$

Fin Objetivo

En la figura 3.2, se puede observar la representación gráfica de un objetivo

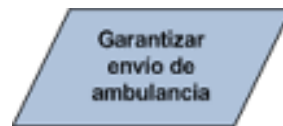


Figura 3.2. Representación gráfica de un objetivo. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.1.1 Objetivo tipo Lograr

Permite representar situaciones en una corta duración. Además, establece que alguna condición de un objetivo se deba cumplir en un estado futuro. Es decir, requiere que una propiedad en algún tiempo futuro eventualmente se cumpla.

Lograr: $P \Rightarrow \Diamond_{\leq d} Q$

Este tipo de objetivo es etiquetado, en NL2KAOS, con los verbos de logro (ver tabla 4.7 del capítulo 4). En la figura 3.3, se puede observar la representación gráfica de un objetivo de logro.



Figura 3.3. Representación gráfica de un objetivo de logro. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.1.2 Objetivo tipo Evitar

Permite que una posible situación nunca se dé, es decir, requiere que una propiedad nunca se cumpla.

Evitar: $P \Rightarrow \Box \neg Q$

Este tipo de objetivo es etiquetado, en NL2KAOS, con los verbos de evitar (ver tabla 4.7 del capítulo 4). Su representación gráfica se puede observar en la figura 3.4.



Figura 3.4. Representación gráfica de un objetivo de evitar. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.1.3 Objetivo tipo Parar

Permite que una situación deje de cumplirse en el tiempo, es decir, debe garantizar que unas condiciones en los estados del sistema se suspendan en un momento dado.

Parar: $P \Rightarrow \Diamond_{\leq d} \neg Q$

Este tipo de objetivo es etiquetado, en NL2KAOS, con los verbos de parar (ver tabla 4.7 del capítulo 4). Su representación se muestra en la figura 3.5.



Figura 3.5. Representación gráfica de un objetivo de tipo parar. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.1.4 Objetivo tipo Mantener

Permite que una propiedad se cumpla y se mantenga durante un determinado periodo, es decir, requiere que una propiedad siempre se mantenga.

Mantener: $P \Rightarrow \Box Q$

Este tipo de objetivo es etiquetado, en NL2KAOS, con los verbos de mantenimiento (ver tabla 4.7 del capítulo 4). Su representación gráfica se puede observar en la figura 3.6.

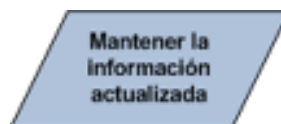


Figura 3.6. Representación gráfica de un objetivo de mantenimiento. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.1.5 Softgoal

Permite definir objetivos que no tienen un criterio claro para su satisfacción. Un ejemplo de éstos, son los objetivos "no funcionales". Su representación gráfica se puede observar en la figura 3.7.



Figura 3.7. Representación gráfica de Softgoal en el diagrama de KAOS.

Fuente: (Matulevičius et al., 2007).

De los constructos presentados en esta sección, se seleccionaron para la realización de esta tesis doctoral los siguientes: (i) Objetivo, (ii) Objetivo de evitar, (iii) Objetivo de mantenimiento, (iv) Objetivo de logro, (v) Objetivo de parar, (vi) Objeto (entidad), (vii) Agente, (viii) Agente de Software, (ix) Agente de ambiente, y (x) Operación. Éstos fueron elegidos porque el enfoque de esta tesis está orientado a la utilización de los elementos básicos del diagrama de objetivos de KAOS, por tanto, los demás constructos, están por fuera del alcance del presente estudio.

3.2.1.6 Requisito

Es un objetivo de bajo nivel, es decir, es un objetivo hoja que no se puede descomponer (atomizar) más, además, la responsabilidad de éste, corresponde a un agente de software.

$\forall o: \text{Objetivo}$

$$o.\text{subObjs} = \emptyset \wedge (\exists a: \text{AgenteSoftware}) \text{Responsabilidad}(a, o) \\ \rightarrow \text{InstanciaDe}(o, \text{Requisito})$$

Este tipo de objetivo puede ser etiquetado, en NL2KAOS, con los verbos de objetivos presentados en el capítulo 4 (ver tabla 4.7). Su representación se puede observar en la figura 3.8.



Figura 3.8. Representación gráfica de un requisito. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.1.7 Expectativa

Es un objetivo de bajo nivel, es decir, es un objetivo hoja que no se puede descomponer (atomizar) más, además, la responsabilidad de éste, corresponde a un agente de ambiente.

$\forall o: \text{Objetivo}$

$$o.\text{subObjs} = \emptyset \wedge (\exists a: \text{AgenteAmbiente}) \text{Responsabilidad}(a, o) \\ \rightarrow \text{InstanciaDe}(o, \text{Expectativa})$$

Este tipo de objetivo puede ser etiquetado, en NL2KAOS, con los verbos de objetivos presentados en el capítulo 4 (ver tabla 4.7). Su representación gráfica se puede observar en la figura 3.9.



Figura 3.9. Representación gráfica de una expectativa. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.1.8 Refinamiento de Objetivo

Elemento que permite obtener la representación jerárquica del diagrama de objetivos, es decir, permite refinar un objetivo padre a través de la subrogación de objetivos, los cuales deberán ser suficientes para satisfacer al objetivo padre. La subrogación se realiza hasta alcanzar los requisitos o expectativas (objetivos hojas que no se puede descomponer más). Es llamado también enlace de subrogación entre los elementos (objetivo, expectativa, requisito) del diagrama de objetivos de KAOS. Existe el refinamiento de tipo “AND”, se utiliza cuando se requiere de la consecución simultánea de dos o más objetivos, y de tipo “OR”, se utiliza cuando es suficiente con la consecución de uno de los objetivos unidos mediante este tipo de refinamiento. En la figuras 3.10 y 3.11, se puede observar su representación gráfica.

Definición tipo “AND”

$\forall o: \text{Objetivo}$

$$o.\text{subObjs} \neq \emptyset \wedge \forall x \in o.\text{subObjs}: \text{Satisface}(x) \rightarrow \text{Satisface}(o)$$

Definición tipo “OR”

$\forall o: \text{Objetivo}$

$$o.\text{subObjs} \neq \emptyset \wedge \exists x \in o.\text{subObjs}: \text{Satisface}(x) \rightarrow \text{Satisface}(o)$$

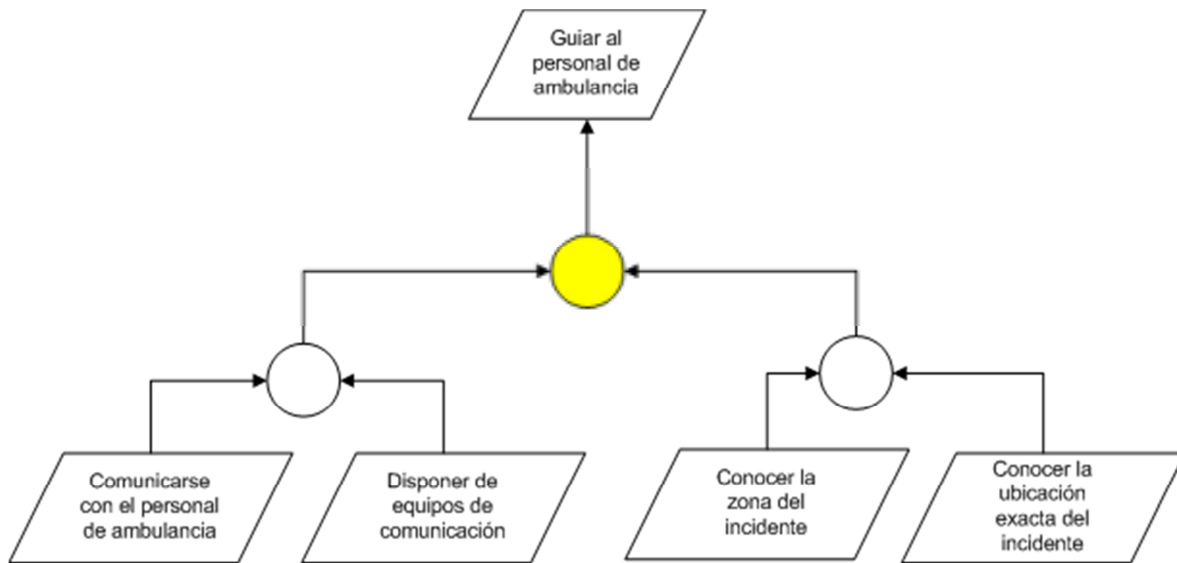


Figura 3.10. Representación gráfica de refinamiento tipo AND. Fuente: adaptado de (Matulevičius et al., 2007).

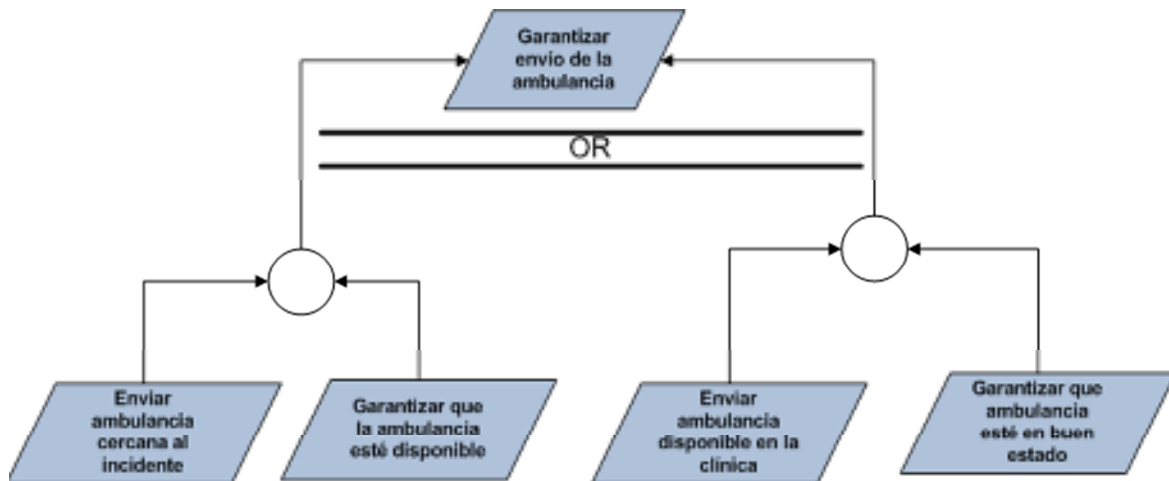


Figura 3.11. Representación gráfica de refinamiento tipo OR. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.1.9 Conflicto

Permite definir que dos o más objetivos están en conflicto y se vuelven lógicamente inconsistentes en un dominio de interés, es decir, que ambos objetivos no se pueden alcanzar de forma conjunta. En la figura 3.12, se puede observar su representación gráfica.



Figura 3.12. Representación gráfica de conflicto. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.1.10 Condición de Frontera

Permite describir las inconsistencias en un dominio determinado, cuando dos o más objetivos diferentes no pueden ser alcanzados simultáneamente, permitiendo minimizar el conflicto existente entre los objetivos implicados. Este elemento hace parte del modelo de objetivos, su representación se puede apreciar en la figura 3.13.

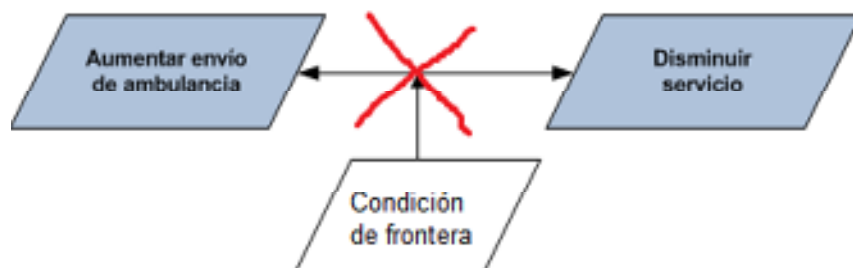


Figura 3.13. Representación gráfica de frontera. Fuente: (Matulevičius et al., 2007).

3.2.2 Agente

Puede ser un componente humano o automático (software) con capacidad de realizar diversas operaciones, es el responsable de un requisito o una expectativa. Existen agentes de ambiente (Environment Agent) y de software (Software Agent).

Entidad Agente

Tiene nombre, definición

Invariable $\text{nombre} \neq \emptyset \wedge \text{definición} \neq \emptyset$

Fin Agente

Su representación gráfica se puede observar en la figura 3.14.



Figura 3.14. Representación gráfica de un agente. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.2.1 Agente de Ambiente

Será un agente de ambiente sólo si su representación está dada a través de: (i) componentes de software pre-existente, (ii) sensores, (iii) actuadores, (iv) humanos, y (v) unidades organizacionales, es el responsable de alcanzar un objetivo que se denominará expectativa. En la figura 3.15, se observa la forma en que se debe presentar este constructo dentro de una representación gráfica del diagrama de objetivos.



Figura 3.15. Representación gráfica agente de ambiente. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.2.2 Agente de Software

Será un agente de software sólo si su representación está dada a través de componentes automáticos (software) responsables de alcanzar un objetivo que se denominará requisito. En la figura 3.16, se puede observar su representación gráfica.



Figura 3.16. Representación gráfica de Agentes de software en el diagrama de KAOS. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.2.3 Relación de Responsabilidad y de Asignación

(i) Responsabilidad: es un elemento que permite asignar un agente a un requisito que está bajo su responsabilidad, se representa a través de un círculo sombreado de color rojo, y (ii) Asignación: es un elemento que permite asignar un agente a una expectativa, se representa a través de un círculo sombreado de color rosado. En la figura 3.17, se puede observar su representación gráfica.

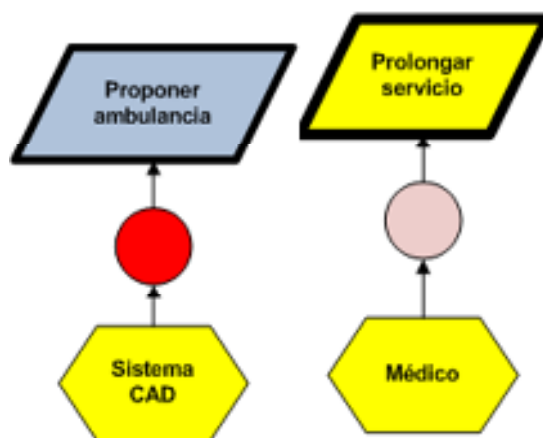


Figura 3.17. Representación gráfica de responsabilidad y asignación. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.3 Operación

Es el elemento encargado de operacionalizar los requisitos. Luego de ser realizada una operación, ésta deja como salida su(s) respectiva(s) post-condición(es), es decir, los resultados o cambios generados en los datos.

Entidad Operacion

Tiene nombre, definicion, preCondicion, postCondicion, categoria, entradas, salidas

Invariable $nombre \neq \emptyset \wedge definicion \neq \emptyset \wedge preCondicion \neq \emptyset \wedge postCondicion \neq \emptyset$

Fin Operacion

En la figura 3.18, se puede observar su representación gráfica en el diagrama de objetivos de KAOS.

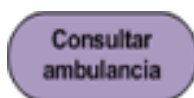


Figura 3.18. Representación gráfica de una operación. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.3.1 Operacionalización

Permite la asignación de operaciones a un objetivo de bajo nivel (requisito). Esta asignación puede ser de tipo “AND” si se requiere de la participación simultánea de dos o más operaciones, o de tipo “OR”, si solamente se requiere una operación para satisfacer el requisito. Su representación gráfica se puede observar en la figuras 3.19 y 3.20.

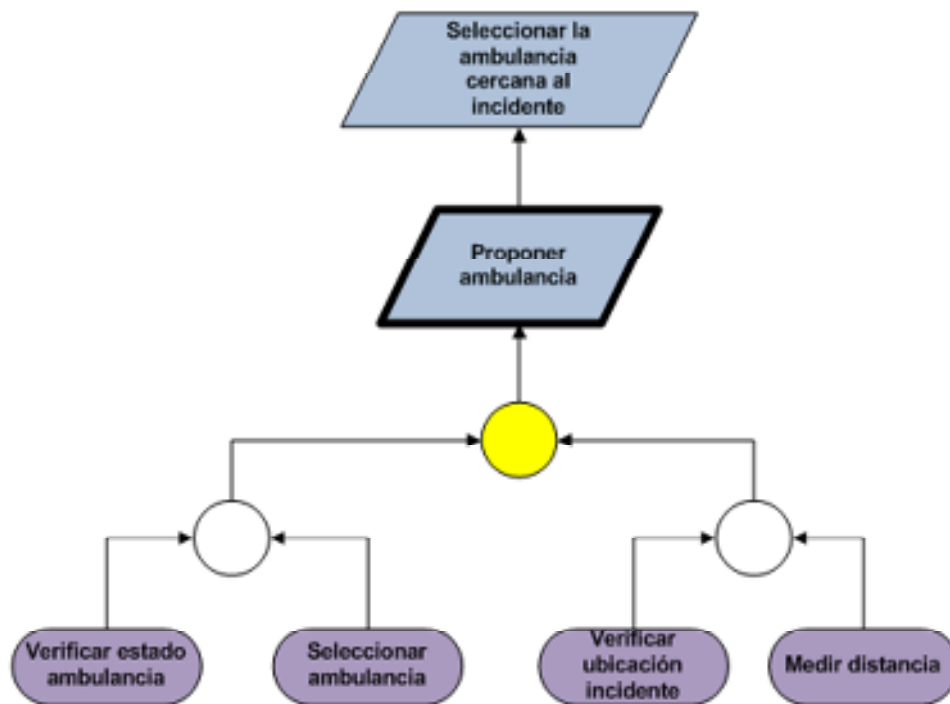


Figura 3.19. Representación gráfica de operacionalización tipo AND. Fuente: adaptado de (Matulevičius et al., 2007).



Figura 3.20. Representación gráfica de operacionalización tipo OR. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.3.2 Entrada

Define la fuente de información que requiere una operación en el marco de un agente para poder ejecutarse. Esta fuente de información apuntará a un atributo de un objeto que conforma el sistema. En la figura 3.21 se puede observar su representación gráfica.



Figura 3.21. Representación gráfica de entrada. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.3.3 Salida

Permite el retorno del dato que realiza una operación. Su representación gráfica se puede observar en la figura 3.22.

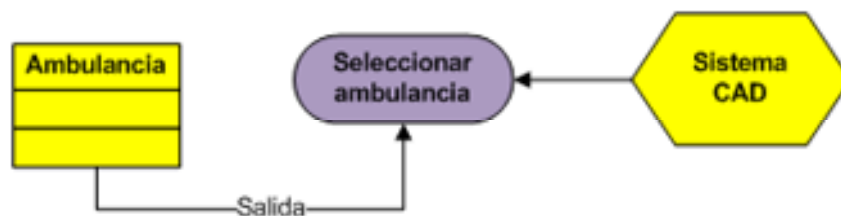


Figura 3.22. Representación gráfica de salida. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.4 Evento

Es el elemento encargado de activar las operaciones. Hace parte del modelo de objetos y de operación.

Entidad Evento

Tiene nombre, definición,

Invariable $\text{nombre} \neq \emptyset \wedge \text{definición} \neq \emptyset$

Fin Evento

Su representación se puede observar en la figura 3.23.

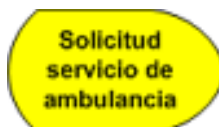


Figura 3.23. Representación gráfica de un evento. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.5 Objeto

Es un elemento que permite representar una entidad de interés en el sistema que se está modelando.

Entidad Objeto

Tiene nombre, definición

Invariable $\text{nombre} \neq \emptyset \wedge \text{definición} \neq \emptyset$

Fin Objeto

Su representación gráfica se puede observar en la figura 3.24.

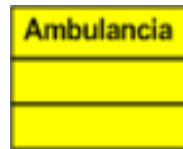


Figura 3.24. Representación gráfica objeto. Fuente: adaptado de (Matulevičius et al., 2007).

3.2.6 Propiedad de Dominio

Una propiedad de dominio es una propiedad que es naturalmente verdadera en el sistema. Las propiedades de dominio se declaran como invariantes de dominio conectadas a los objetos del modelo de objetos. Su representación gráfica se puede observar en la figura 3.25.



Figura 3.25. Representación gráfica de propiedad de dominio. Fuente: adaptado de (Matulevičius et al., 2007).

3.3 Ontología del metamodelo KAOS

Introducción

Un metamodelo es una definición precisa de los conceptos y reglas necesarias para la creación de modelos semánticos (UML, 2004). Los metamodelos pueden servir para múltiples propósitos (Sousa, 2008): (i) definir un esquema para datos semánticos que necesitan ser almacenados, (ii) precisar un lenguaje que soporte una metodología particular o proceso, (iii) definir un lenguaje para expresar semántica adicional de la información existente, (iv) permitir a un analista capturar, analizar y entender mejor nuevos enfoques, y (v) servir como base para definir una herramienta automática de soporte.

Con el propósito de caracterizar semánticamente los elementos que conforman el diagrama de objetivos de KAOS definidos conceptualmente en la sección anterior, se desarrolló en esta tesis una Ontología a partir del metamodelo presentado por, (Matulevičius et al., 2007). Ver figura 3.1.

En la figura 3.26 se presentan los componentes de la Ontología construida que será utilizada para la caracterización semántica de los elementos que conforman la metodología KAOS. En la construcción de la ontología se utilizó el lenguaje OWL. En el anexo “C” se pueden observar los detalles de la implementación de la Ontología.

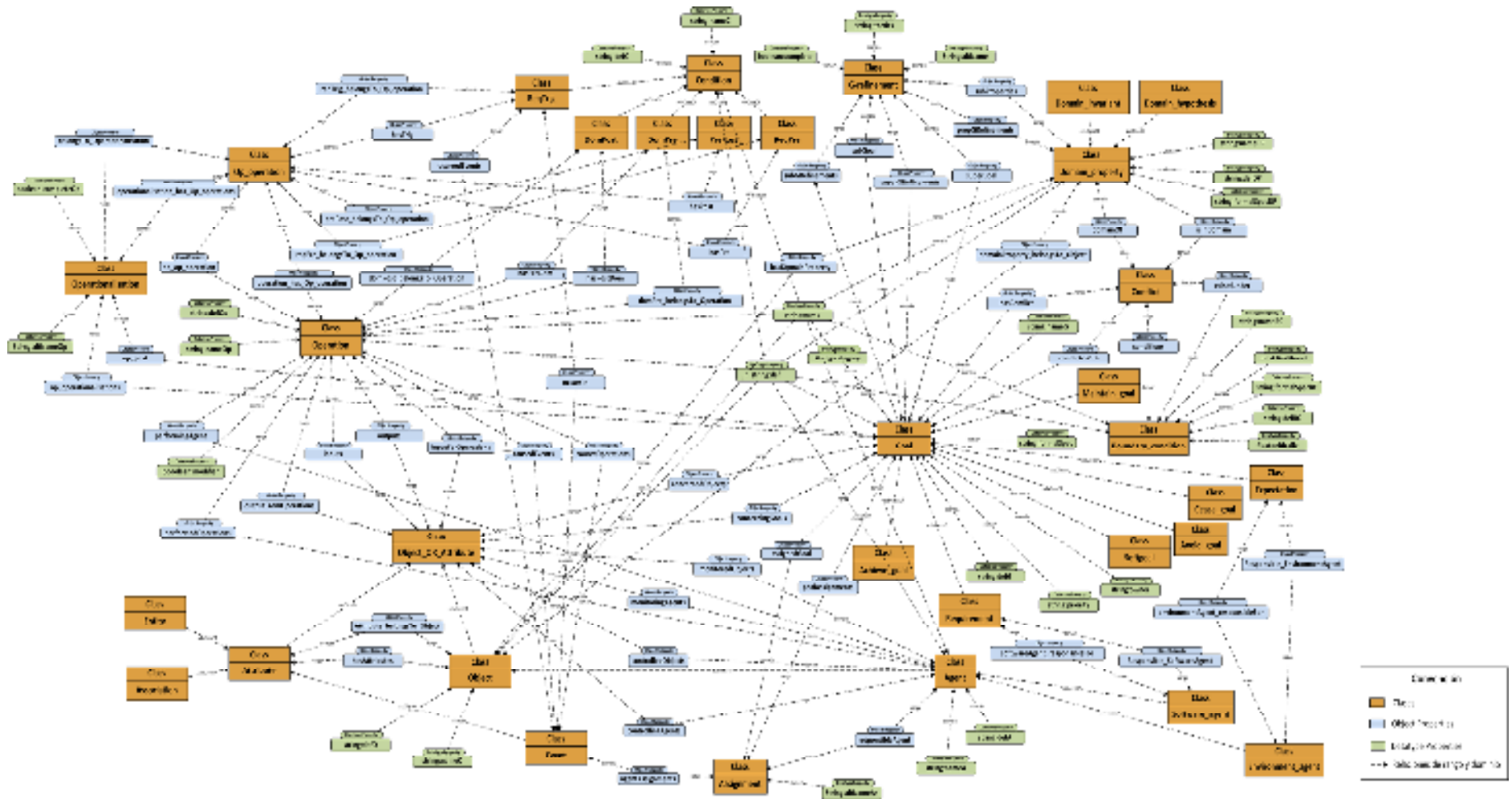


Figura 3.26. Componentes Ontología elementos KAOS.

3.4 Conclusiones del capítulo

En este capítulo se realizó la caracterización conceptual de los elementos utilizados en el diagrama de objetivos de KAOS. Dicha caracterización permitió identificar los elementos que serán utilizados en el marco de esta tesis (ver sección 3.2). Además, se presentó: (i) el metamodelo (Matulevičius et al., 2007), utilizado como punto de partida para la creación de la Ontología propuesta en esta tesis, (ii) la formalización de los elementos caracterizados, y (iii) los componentes de la Ontología construida, la cual será utilizada para la caracterización semántica de los elementos que conforman el diagrama de objetivos de KAOS.

El aporte de este capítulo, está orientado a mejorar la calidad de los elementos del diagrama de objetivos de KAOS obtenidos, al construir una Ontología que permite la identificación, la descripción y la caracterización semántica de los elementos propuestos en la metodología KAOS para que sean comprensibles por una máquina sin ambigüedades.

Capítulo 4

Modelo de procesamiento terminológico interactivo para la definición de requisitos de software bajo la especificación KAOS

El Procesamiento de Lenguaje Natural es una tarea que realizan de manera conjunta los componentes de software y hardware en un sistema de cómputo, éste analiza y/o procesa el lenguaje natural hablado o escrito (Jackson y Moulinier, 2002).

Con base en la definición presentada en la sección 2.1.1 (Montoyo, 2002) y la expresada en el párrafo anterior, se presenta en este capítulo una propuesta para diseñar un modelo de procesamiento terminológico que permita extraer y procesar los términos necesarios que faciliten la caracterización de los elementos básicos del diagrama de objetivos de KAOS a partir de especificaciones textuales.

Este modelo consta de un conjunto de formas morfosintácticas y reglas semánticas para la extracción de los elementos básicos (objetivo, entidad, operación y agente) que conforman el diagrama de objetivos de KAOS, bajo el enfoque semántico caracterizado en el capítulo 3. Con este capítulo se pretende satisfacer el objetivo número dos de esta tesis: diseñar un modelo de procesamiento terminológico interactivo que, a partir de requisitos de software inicialmente descritos en lenguaje natural, permita obtener y validar la caracterización semántica definida en el objetivo anterior (objetivo número uno).

4.1 Vista general del modelo

Con el propósito de atacar los problemas identificados en esta tesis, se procede a definir un modelo de procesamiento terminológico interactivo para la definición de requisitos de software bajo la especificación KAOS. En la figura 4.1 se presenta una visión a nivel general del modelo propuesto.

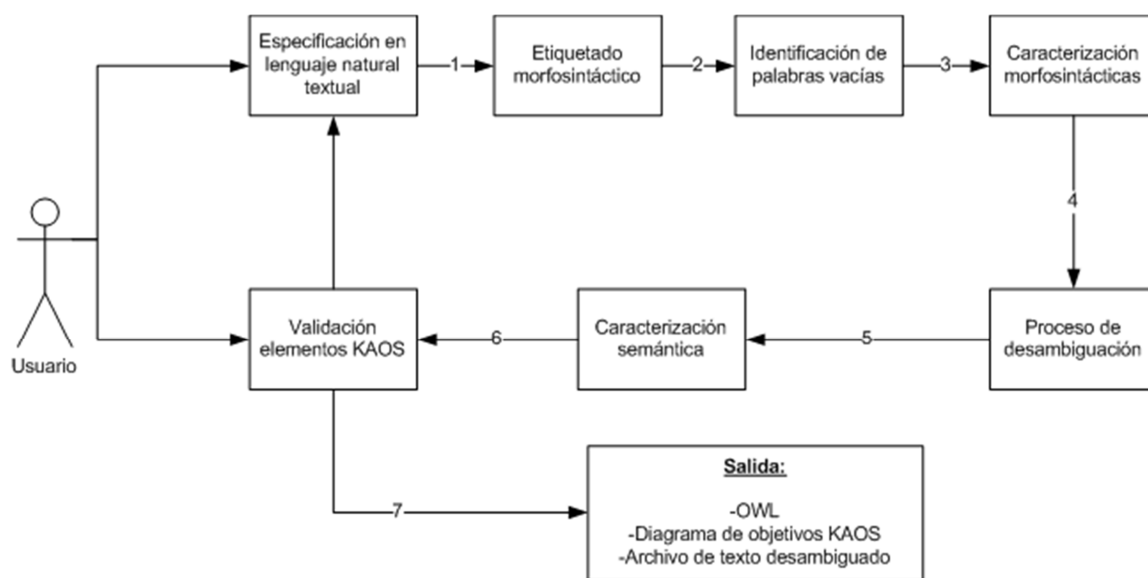


Figura 4.1. Modelo general propuesto. Fuente: elaboración propia

Información de entrada y salida:

1. Especificación en lenguaje natural textual
2. Documento etiquetado
3. Lista de palabras vacías y documento etiquetado
4. Listado de objetivos, agentes, operaciones, entidades y atributos
5. Lista de expresiones con su respectivo sentido
6. Listado objetivos, agentes, operaciones, entidades y atributos caracterizados
7. Listado de objetivos, agentes, operaciones, entidades y atributos caracterizados y validados.

Durante este proceso, todas las frases deben contener explícitamente el sujeto de la oración. Esta es una simplificación que se hace con el fin de evitar trabajar la resolución de anáfora, el cual es un problema adicional y que está por fuera del alcance de esta tesis.

A continuación se describen los módulos que componen el modelo propuesto:

4.1.1 Especificación en lenguaje natural textual

Permite el ingreso de la especificación de los requisitos expresados por el usuario a través de un documento escrito en lenguaje natural del idioma español. Este primer módulo, será el insumo de entrada para la ejecución del proceso planteado en el modelo propuesto. La especificación deberá ser expresada utilizando las reglas morfosintácticas propuestas en esta tesis.

4.1.2 Etiquetado morfosintáctico

Una vez ingresada la especificación textual de los requisitos, este módulo realiza un análisis morfosintáctico con el propósito de etiquetar cada palabra del texto ingresado de acuerdo con su categoría gramatical, es decir, la función (verbo, sustantivo, artículo, entre otros) que desempeña en su contexto, de tal forma, que permita la ejecución de los módulos restantes.

En la tabla 4.1 se puede observar un ejemplo de las categorías gramaticales que podrían ser identificadas y etiquetadas en este proceso.

Tabla 4.1. Ejemplo de categorías gramaticales presentes en un texto

Lenguaje Natural	Categoría Gramatical
El sistema permite validar los datos	El: <i>artículo</i> ; sistema: <i>sustantivo</i> ; permite: <i>verbo</i> ; validar: <i>verbo</i> ; los: <i>artículo</i> ; datos: <i>sustantivo</i>

4.1.3 Identificación de palabras vacías

Este módulo realiza el proceso que permite capturar las palabras vacías expresadas en la especificación textual de los requisitos, es decir, aquellas palabras sin significado en sí mismo, como lo son los artículos, pronombres, preposiciones, entre otros. Dado lo anterior, se compara cada una con la lista de palabras vacías del español que aparecen en Tartarus (2013). Las palabras que estén en ambos conjuntos se almacenan en una lista para excluirlas del proceso de desambiguación semántica.

En la tabla 4.2 se puede observar un ejemplo de las palabras vacías que podrían ser capturadas en este proceso.

Tabla 4.2. Ejemplo de palabras vacías presentes en un texto

Lenguaje Natural	Palabras vacías
El sistema permite validar los datos	El, los

4.1.4 Caracterización morfosintáctica

El módulo para la caracterización morfosintáctica de los elementos básicos del diagrama de objetivos de KAOS se ejecuta una vez finalizada la identificación de palabras vacías, para ello, se utiliza el Procesamiento del Lenguaje Natural (PLN).

El PLN es una de las ramas de la Inteligencia Artificial (IA) que permite obtener información de un dominio a partir de un discurso, esta información puede ser formalizada por algún mecanismo de inferencia sobre un texto o un discurso en particular (Cvitas, 2011). Según la gramática de casos de (Fillmore, 1968), a partir de un conjunto concreto de casos se puede modelar un comportamiento esperado en el idioma, es por ello que, el método que aquí se propone hará uso de este enfoque, es decir, se modelará un conjunto de casos mediante diferentes formas con el fin de extenderlas completamente al Lenguaje Natural en Español.

A nivel general, en este modelo una oración en español puede describir objetivos, entidades, agentes y operaciones, sigue una estructura morfosintáctica específica que servirá como referencia para identificar y clasificar estos elementos. Dado que el insumo de entrada para esta caracterización morfosintáctica es el lenguaje natural en español, se hace necesario definir los siguientes componentes gramaticales que se contemplarán como expresiones válidas.

Oración o frase: conjunto de palabras ordenadas sintácticamente que expresan un sentido completo, son usadas habitualmente por las personas para indicar hechos (Saporiti, 2006).

Verbo: Clase de palabra que puede tener variación de persona, número, tiempo y modo (DRAE, 2013).

Verbo infinitivo: es todo aquel que aparece en forma no personal y terminado en ar, er o ir (p.e. Garantizar, ofrecer, permitir) (Saporiti, 2006).

Verbo Nominalizado: es aquel sustantivo proveniente de un verbo, (ejemplo: la realización, proveniente de realizar) (Saporiti, 2006).

Verbo auxiliar: es todo aquel que se utiliza como componente en la perífrasis verbal (p.e. Deber, haber, poder) (Saporiti, 2006).

Verbo auxiliado: es todo aquel que aparece en forma no personal, es decir, en: infinitivo (permitir), gerundio (permitiendo), participio (permitido) (Saporiti, 2006).

Verbo no copulativo: son verbos predicativos, es decir, dan indicaciones activas de lo que hace el sujeto (p.e. envía, consulta, guarda, entre otros). Mientras que, el copulativo sólo entrega descripción nominal, es decir, información de un estado o atributo (p.e. Ser, estar, parecer, entre otros) (DRAE, 2013).

Sustantivo: es la expresión que funciona como sujeto en una oración. Tiene género (masculino o femenino) y número (singular o plural) (Saporiti, 2006).

Sintagma Nominal: es un grupo de expresiones que se articulan alrededor (antes y/o después) del sustantivo (también llamado nombre), el cual corresponde al núcleo en este tipo de sintagma (Demonte y Bosque, 1999).

Expresión: palabra o conjunto de palabras utilizadas para especificar o declarar algo para darlo a entender (DRAE, 2013).

Complemento: es el conjunto de expresiones que acompañan la oración con el propósito de darle sentido o significado a ésta (DRAE, 2013).

Sujeto: en una oración, corresponde a la persona animal o cosa que realiza una acción del verbo (DRAE, 2013).

Objeto directo: “Nombre, pronombre, sintagma o proposición en función nominal, que completa el significado de un verbo transitivo”. Ejemplo: El profesor califica el examen. (DRAE, 2013).

Objeto indirecto: “Nombre, pronombre, sintagma o proposición en función nominal, que completa el significado de un verbo transitivo o intransitivo, expresando el destinatario o beneficiario de la acción”. Ejemplo: El profesor entrega el examen al estudiante. (DRAE, 2013).

Preposición: “Palabra invariable (a, sobre, durante, entre otras) que introduce elementos nominales u oraciones subordinadas sustantivas haciéndolos depender de alguna palabra anterior” (DRAE, 2013).

Perífrasis verbales: son construcciones sintácticas de dos o más verbos que funcionan como núcleo del predicado. Sirven para expresar las características de la acción verbal que no pueden señalarse mediante el uso de las formas simples o compuestas (Genta, 2008). Su macro estructura se puede observar en la tabla 4.3.

Tabla 4.3. Macro estructura de las perífrasis verbales. Fuente: (Genta, 2008).

Verbo auxiliar	Nexo	Verbo Auxiliado
Verbo que aparece en forma personal y aporta los contenidos gramaticales.	Pueden ser conjunciones, preposiciones o puede no haber nexo.	Verbo que aparece en forma no personal, es decir, en infinitivo, gerundio, participio.

Asimismo, las perífrasis se pueden concentrar en dos grandes grupos, las perífrasis verbales aspectuales y las perífrasis verbales modales. En el marco de

este modelo sólo se contemplarán las perífrasis verbales modales. En la tabla 4.4, se detalla la estructura morfosintáctica de una perífrasis de este tipo.

Tabla 4.4. Estructura morfosintáctica de perífrasis verbales modales válidas en el modelo. Fuente: (Genta, 2008).

Perífrasis verbales modales válidas			
Tipo	Verbo auxiliar	Verbo Auxiliado	Nexo (opcional)
De obligación	Tener que/ Deber/ Haber de / Haber que	+ Infinitivo	Conjunciones o preposiciones
De probabilidad o suposición	Venir a / Deber de	+ Infinitivo	
Posibilidad	Poder	+ Infinitivo	

La entrada de este proceso, al igual que para los otros elementos, consiste en un documento redactado en idioma español, sin errores ortográficos, gramaticales o ambigüedades, que contiene las especificaciones de los requisitos funcionales del sistema, en el cual se describen los agentes que hacen parte del sistema a desarrollar. La estructura morfosintáctica del texto que se procesa en este módulo deberá tener oraciones en voz activa, pues son las que indican claramente los sujetos de la acción. Además, el verbo principal de la oración debe ser no copulativo, de tal forma, que indiquen un cambio de estado, y no una descripción nominal del mismo.

A continuación, se definirán las reglas morfosintácticas, las cuales serán utilizadas para: (i) realizar la especificación textual de los requisitos, y (ii) caracterizar los elementos básicos (objetivo, operación, agente, entidad) del diagrama de objetivos de KAOS.

4.1.4.1 Reglas morfosintácticas para caracterizar objetivos

Las siguientes formas morfosintácticas que se proponen permitirán caracterizar en el contexto del idioma español un objetivo independiente del tipo al que pertenece (según la clasificación KAOS):

FMOB1: Sintagma Nominal + Perífrasis verbal modal + Verbo auxiliado + Verbo nominalizado + Complemento.

Ejemplo: El sistema + debe/tiene que + permitir + El registro + de los clientes.

Explicación. Es importante resaltar que, la perífrasis verbal modal para este caso corresponde al *debe o tiene que* + verbo auxiliado. Complemento: en este modelo el complemento se define como una agrupación de palabras que no contienen ningún verbo, pero por el contrario es permitido todo tipo de adverbio, adjetivo o stop word, y no se limita el número de palabras presentes en el complemento, la única restricción es que el complemento no contenga ninguna forma verbal.

FMOB2: Que + Sintagma Nominal + Verbo no copulativo + Complemento.

Ejemplo: Que + la secretaria + registre + los clientes.

Explicación. Complemento: será restringido sólo a sintagmas nominales con o sin adjetivos, por ejemplo que la secretaria registre “los clientes rápido” es un complemento válido, así como que la secretaria registre “los archivos” también es un complemento válido.

FMOB3: Verbo en infinitivo + que + Sintagma Nominal + Verbo no copulativo + Complemento.

Ejemplo: “Garantizar/lograr/permitir” + que + el sistema/la secretaria + registre + los clientes.

Explicación. Bajo el modelo propuesto en esta tesis, el complemento se define como una agrupación de palabras, las cuales no contienen ningún verbo, pero por el contrario es permitido todo tipo de adverbio, adjetivo o stop word, y no se limita el número de palabras presentes en complemento.

FMOB4: Sintagma Nominal + Perífrasis verbal modal + Verbo auxiliado + complemento.

Ejemplo: El sistema + debe/tiene que + registrar + los clientes

Explicación. Es una variante de FMOB1 que permite expresar objetivos con un poco más de rigor, sin embargo, el complemento también debe estar restringido. Complemento: bajo este modelo el complemento se define como una agrupación de expresiones, las cuales no contienen ningún verbo, pero por el contrario, es permitido todo tipo de adverbio, adjetivo o stop word, y no se limita el número de expresiones presentes en el complemento, la única restricción es que el complemento no contenga ninguna forma verbal.

En la tabla 4.5, se puede observar un ejemplo de las formas morfosintácticas definidas para caracterizar objetivos.

Tabla 4.5. Ejemplo de las formas morfosintácticas definidas para caracterizar objetivos. Fuente: elaboración propia.

Lenguaje natural	Forma utilizada	Elementos morfosintácticos identificados
El sistema debe permitir el registro de los clientes.	FMOB1	Sintagma nominal: “el sistema” Perífrasis verbal modal: “debe permitir” Verbo nominalizado: “el registro” Complemento: “de los clientes”
Que la secretaria obtenga los teléfonos.	FMOB2	Expresión: “que” Sintagma nominal: “la secretaria” Verbo no copulativo: “obtenga” Complemento: “los teléfonos”
Garantizar que el director guarde la información de los clientes.	FMOB3	Verbo en infinitivo: “garantizar” Expresión: “que” Sintagma nominal: “el director” Verbo no copulativo: “guarde” Complemento: “la información de los clientes”
El sistema debe validar los datos.	FMOB4	Sintagma Nominal: “El sistema”; Perífrasis verbal modal: “debe validar”; Complemento: “los datos”.

De acuerdo con la tabla anterior, las reglas morfosintácticas para la identificación de objetivos queda determinada de la siguiente manera:

$$\text{Si } FMOB1(s) \vee FMOB2(s) \vee FMOB3(s) \vee FMOB4(s) \rightarrow \text{objetivo}(\text{nombre: } \{inf(s_{vp}) + s_{vp+}\})$$

Lo anterior significa que, si una oración cumple con alguna de las formas morfosintácticas definidas para los objetivos, se crea una instancia de objetivo, representado en la Ontología presentada en el capítulo 3. El parámetro que se pasa es el nombre que se le da al objetivo, siendo, para esta regla, el verbo principal (s_{vp}), más el conjunto de expresiones que se encuentren después del verbo principal (s_{vp+}).

4.1.4.2 Reglas Morfosintácticas para identificar agentes

FMAG1: Sustantivo o sintagma nominal + Verbo no copulativo + Complemento.

Explicación: para el complemento, aplican las mismas condiciones definidas en las formas FMOB1, FMOB2, FMO3 y FMO4. Adicionalmente, si bien es posible que haya sustantivos en el complemento, éstos se tomarán como objetos (directos o indirectos) y no como sujetos. De esta forma, el complemento no puede conformarse por una oración diferente a la principal, es decir, dos ideas (dos sujetos y dos verbos principales) siempre deberán separarse por punto final de oración. Asimismo, es claro que cada oración debe tener explícito el sujeto que la realiza, el cual debe nombrarse de forma única para evitar ambigüedades al tener el mismo agente con diferentes nombres, pues el sistema no logra desambiguar esta problemática.

Algunos ejemplos pueden ser: “El personal de ambulancia envía información al operador de radio”, “El operador de radio recibe información de la ambulancia seleccionada”, donde claramente el agente es el conjunto de palabras que hay antes del verbo. Sin embargo, no todas las frases que cumplen con esta forma se refieren a agentes, por ejemplo: “El tiempo necesario para atender la llamada y llenar el formulario del incidente no deben tomar más de cierto tiempo”. Este tipo de oraciones que no se refieren a agentes, son las que se eliminan a través de la metodología que se presenta en la implementación, ya que es menester eliminar aquellos agentes poco frecuentes.

En la tabla 4.6, se puede observar un ejemplo de la forma morfosintáctica definida para caracterizar agentes.

Tabla 4.6. Ejemplo de forma morfosintáctica para caracterizar agentes. Fuente: elaboración propia.

Lenguaje natural	Elementos morfosintácticos identificados	Agente identificado
El personal de ambulancia envía información	Sintagma nominal: “ <i>El personal de ambulancia</i> ” Verbo principal: “ <i>envía</i> ” (no copulativo) Complemento: “ <i>información</i> ”	“ <i>El personal de ambulancia</i> ”
El sistema CAD puede proponer otra ambulancia	Sintagma nominal: “ <i>El sistema CAD</i> ” Verbo principal: “ <i>puede</i> ” (no copulativo) Complemento: “ <i>proponer otra ambulancia</i> ”	“ <i>El Sistema CAD</i> ”
El controlador es de la empresa CX.	Sintagma nominal: “ <i>El controlador</i> ” Verbo principal: “ <i>es</i> ” (copulativo) Complemento: “ <i>proponer otra ambulancia</i> ”	No hay agente (el verbo principal es copulativo)

De acuerdo con lo presentado en la tabla anterior, las reglas morfosintácticas para la identificación de agentes queda determinada de la siguiente manera:

Si $FMOB1(s) \vee FMOB2(s) \vee FMOB3(s) \vee FMOB4(s) \rightarrow agente(nombre: SN)$

Es decir, si una oración cumple con alguna de las formas morfosintácticas definidas para los objetivos, se crea una instancia del agente, el parámetro que se pasa es el nombre que se le da al agente, siendo, para esta regla, el sintagma nominal (SN).

4.1.4.3 Reglas morfosintácticas para identificar entidades y atributos

FME1: Sustantivo + preposición (de/del) + sustantivo

Si en una frase se encuentran dos sustantivos unidos por la preposición “de” o “del” el segundo sustantivo será una entidad, y el primer sustantivo será candidato a ser atributo de la entidad identificada.

FME2: Sustantivo + verbo (tiene) + sustantivo

Si en una frase se encuentran dos sustantivos unidos por el verbo “tiene” el primer sustantivo será una entidad, y el segundo sustantivo será candidato a ser atributo de la entidad identificada.

En la tabla 4.7, se puede observar un ejemplo de la forma morfosintáctica definida para caracterizar entidades y atributos.

Tabla 4.7. Ejemplo de forma morfosintáctica para caracterizar entidades y atributos. Fuente: elaboración propia.

Lenguaje natural	Forma utilizada	Elementos morfosintácticos identificados	Entidad	Atributo
El radio operador recibe <i>información del personal</i> de la ambulancia	FME1	Sustantivo: “información” y “personal” Preposición: “del”	Persona	Información
Ambulancia tiene placa	FME2	Sustantivo: “ambulancia” y “placa” Verbo: “tiene”	Ambulancia	Placa

De acuerdo con lo presentado en la tabla anterior, las reglas morfosintácticas para la identificación de entidades quedan determinadas de la siguiente manera:

Si $FME1(s) \rightarrow entidad(nombre: S2) \wedge atributo(nombre: S1)$

Es decir, si una oración cumple con la forma FME1 para caracterizar entidades, se crea una instancia de entidad representada en la Ontología presentada en el capítulo 3 de esta tesis. El parámetro que se pasa es el nombre que se le da a la entidad siendo para esta regla, el segundo sustantivo S2, también se crea un atributo cuyo parámetro es nombre que corresponde al primer sustantivo S1.

Si $FME2(s) \rightarrow entidad(nombre: S1) \wedge atributo(nombre: S2)$

Es decir, si una oración cumple con la forma FME2 para caracterizar entidades, se crea una instancia de entidad representada en la Ontología presentada en el capítulo 3 de esta tesis. El parámetro que se pasa es el nombre que se le da a la entidad siendo para esta regla, el primer sustantivo S1, también se crea un atributo cuyo parámetro es el nombre que corresponde al segundo sustantivo S2.

Si un sustantivo o sintagma nominal es identificado como entidad y como posible atributo tiene prelación la entidad.

$\forall (entidad(nombre: S1) \wedge atributo(nombre: S1)) \rightarrow entidad(nombre: S1)$

4.1.4.4 Reglas morfosintácticas para identificar operaciones

Para aplicar las reglas morfosintácticas que permitan identificar operaciones, es necesario extraer el verbo principal (vp) de cada oración y el objeto directo (dobj) asociado a él.

FMOP1: Sustantivo o sintagma nominal + Verbo no copulativo + Complemento.

Explicación: Para el complemento, aplican las mismas condiciones definidas en las formas FMOB1, FMOB2, FMO3 y FMO4. Adicionalmente, los sustantivos en el complemento se tomarán como objetos directos y no como sujetos. Asimismo, para esta primera forma morfosintáctica el complemento deberá contar con un objeto preposicional.

Si: $(\exists \text{dobj}) \wedge (\exists \text{obj-prep}) \rightarrow \text{input}(\text{dobj}, \text{obj-prep}), \text{output}(\text{dobj}, \text{SUSTANTIVAR}(\text{vp}))$.

La forma FMOP1 establece que, dada una operación de la forma indicada, se identifica la variable de entrada que consta del objeto directo y el preposicional, y la variable de salida que se compone del objeto directo y la sustantivación del verbo principal.

FMOP2: Sustantivo o sintagma nominal + Verbo no copulativo + Complemento.

Explicación: Para el complemento, aplican las mismas condiciones definidas en las formas FMOB1, FMOB2, FMO3 y FMO4. Adicionalmente, los sustantivos en el complemento se tomarán como objetos directos y no como sujetos. Para esta segunda forma morfosintáctica el complemento no requiere contar con un objeto preposicional.

Si: $(\exists \text{dobj}) \wedge (\nexists \text{obj-prep}) \rightarrow \text{input}(\text{dobj}), \text{output}(\text{dobj}, \text{SUSTANTIVAR}(\text{vp}))$

La forma FMOP2 indica que, si una frase con un verbo principal catalogado como de operación, posee en su complemento un objeto directo, pero no preposicional, la variable de entrada será este objeto, y la de salida, el mismo objeto pero acompañado de la sustantivación del verbo principal.

En la tabla 4.8, se puede observar un ejemplo de la forma morfosintáctica definida para caracterizar operaciones.

Tabla 4.8. Ejemplo de forma morfosintáctica para caracterizar operaciones.

Fuente: elaboración propia.

Lenguaje natural	Forma utilizada	Verbo de operación	Objeto directo	Operación	Input	Output
El operario consulta la localización de la ambulancia.	FMOP1	Consultar	Ambulancia	Consultar localización	Localización de la ambulancia	La consulta de la localización
El sistema registra a los usuarios	FMOP2	Registrar	Usuario	Registrar usuario	USUARIO	el registro de usuario

4.1.5 Proceso de desambiguación

De acuerdo con lo presentado en el capítulo dos de esta tesis, el modelo propuesto atacará solamente el problema de la ambigüedad de tipo polisémica. Dado lo anterior, en esta sección se definirán un conjunto de reglas que permitan la identificación y la clasificación semántica de los elementos básicos (objetivo, entidad, operación y agente) del diagrama de objetivos de KAOS. En la especificación KAOS, un objetivo de alto nivel de una solución informática, debe estar clasificado dentro de uno de los siguientes tipos de objetivo: Lograr, Mantener, Parar, Evitar, y éstos a su vez, tienen asociado en su definición un tipo de verbo característico agrupado en alguno o algunos de los tipos anteriormente descritos (Lamsweerde, Dardenne, Delcourt, y Dubisy, 1991). Ver tabla 4.9. Sin embargo, el modelo propuesto en esta tesis, también podrá encontrar objetivos que no estén enmarcados en los cuatro tipos de objetivos antes señalados.

Tabla 4.9. Clasificación verbos de objetivos. Adaptado de (Zapata y Lezcano, 2009).

Verbos para Objetivos tipo Lograr	Verbos para Objetivos tipo Mantener	Verbos para Objetivos tipo Parar	Verbos para Objetivos tipo Evitar
Formar	Administrar	Parar	Evitar
Mejorar	Garantizar	Finalizar	Obstaculizar
Aumentar	Guardar	Suspender	Prevenir
Fomentar	Ofrecer	Detener	Esquivar
Registrar	Prolongar	Concluir	Eludir
Elaborar	Avalar	Interrumpir	Sortear
Formular	Gestionar	Cesar	
Promover	Obtener	Terminar	
Hacer	Preservar		
Meter	Perdurar		
Obrar	Conservar		
Preparar	Mantener		
Reducir	Reconocer		
Decrecer	Dar		
Incrementar			
Desarrollar			

En la siguiente sección, se utilizará la clasificación de los verbos presentados en la tabal 4.9, para realizar la clasificación de los objetivos.

4.1.5.1 Regla 1. Clasificación de objetivos

Una vez que se tiene la caracterización morfosintáctica de los objetivos (presentada en la primera parte de este capítulo), se realiza el proceso de desambiguación del verbo principal del objetivo (V_m). Para encontrar el sentido correcto del verbo se utilizaron en esta tesis las técnicas Simplified Lesk, MFS y UKB presentados en el capítulo dos de esta tesis. Su estructura se compone de los siguientes elementos: (i) un identificador para determinar que el concepto es el mismo independiente del idioma, y (ii) identificador universal del sentido.

El sentido seleccionado mediante los métodos elegidos permitirá obtener un identificador único para el sentido del verbo y una definición asociada a éste.

Una vez obtenido el sentido del verbo se procede a realizar la clasificación del tipo de verbo que permitirá a su vez clasificar el objetivo completo.

Se considera “M” como el conjunto de sentidos de los verbos tipo Mantener en donde cada sentido estará en el conjunto $M = \{M_1, \dots, M_n\}$. Análogamente esta misma especificación seguirá para los demás grupos de sentidos de verbos:

Lograr: $L = \{L_1, \dots, L_n\}$

Parar: $P = \{P_1, \dots, P_n\}$

Evitar: $E = \{E_1, \dots, E_n\}$

Dado lo anterior, la clasificación de los objetivos en cualquiera de los tipos será determinado por el siguiente conjunto de ecuaciones.

$$\varphi_M = \frac{\sum_{i=1}^n |Similitud(M_i, V_m)|}{n} \quad (1)$$

$$\varphi_L = \frac{\sum_{i=1}^n |Similitud(L_i, V_m)|}{n} \quad (2)$$

$$\varphi_P = \frac{\sum_{i=1}^n |Similitud(P_i, V_m)|}{n} \quad (3)$$

$$\varphi_E = \frac{\sum_{i=1}^n |Similitud(E_i, V_m)|}{n} \quad (4)$$

En donde similitud es la medida de la interrelación existente entre dos expresiones cualesquiera en un texto, dada por la siguiente ecuación propuesta por (Wu y Palmer, 1994):

$$similitud(concept_1, concept_2) = \frac{2 * depth(LCS)}{depth(concept_1) + depth(concept_2)} \quad (5)$$

Donde depth (LCS) es la distancia completa del árbol de taxonomía usado y depth(concept) representa la profundidad del concepto en el árbol de taxonomía usado.

Una vez se tiene la medida del promedio de similaridad (φ) de cada grupo de objetivos, se realiza el comparativo entre los coeficientes y el mayor coeficiente permitirá realizar la clasificación del verbo en el grupo correspondiente de acuerdo con los tipos objetivos definidos en la metodología KAOS.

4.1.5.2 Regla 2. Clasificación de agente

Considerada $A=\{A_1, A_2, A_3, A_4, \dots, A_n\}$ como una lista ordenada de posibles agentes identificados por medio de reglas morfosintácticas, se realiza el proceso de desambiguación de los sustantivos del agente ($S_{p1}, S_{p2}, \dots, S_{pn}$), para encontrar el sentido correcto de cada sustantivo se utilizaron los métodos Simplified Lesk, MFS y UKB, dado lo anterior, un agente queda definido de la siguiente manera:

$$A_i = \{ S_{p1}, S_{p2}, \dots, S_{pn} \}$$

Dónde:

$$S_{p1} \rightarrow \text{ILI-XXXXXX-N}$$

$$S_{p2} \rightarrow \text{ILI-XXXXXX-N}$$

$$S_{pn} \rightarrow \text{ILI-XXXXXX-N}$$

XXXXXX representa el identificador universal del synset.

Una vez se tiene identificado el sentido correcto para cada sustantivo (articulado a cada agente), se aplica el siguiente procedimiento:

Un agente se considera de software si y sólo si:

$$\exists S_{pi} \in A_j : \maxhyp(S_{pi}) = \text{"Machine"}$$

Un agente se considera de ambiente si y sólo si:

$$\exists S_{pi} \in A_j : \maxhyp(S_{pi}) = \text{"Human"}$$

Donde $maxhyp(S_{pi})$ corresponde al sentido del último hiperónimo en el árbol de taxonomías del MCR o de Wordnet.

4.1.5.3 Regla 3. Clasificación de Operaciones

Se considera un conjunto ordenado de posibles operaciones $O = \{O_1, O_2, \dots, O_n\}$ que cumplen el proceso de identificación morfosintáctico, para la identificación semántica de operaciones se realiza el proceso de desambiguación del verbo principal de la operación (s_{vp}), para encontrar el sentido correcto del verbo se utilizaron los métodos Simplified Lesk, MFS y UKB, por tanto, una operación queda definida por un: (i) identificador que permite determinar que el concepto es el mismo independiente del idioma, y (ii) identificador universal del sentido.

Adicionalmente, se deberá cumplir la siguiente regla:

Sea $P = \{ P_1, P_2, \dots, P_n \}$ el conjunto de sentidos de los verbos de operación, clasificados por (Jaramillo, Zapata y Arango, 2005), por tanto, una operación será clasificada si y sólo si:

$$\mu > \frac{\sum_{i=1}^n |Similitud(P_i, s_{vp})|}{n} \quad (6)$$

Donde μ corresponde a un umbral definido por el usuario con el fin de determinar qué tan estricta es la condición de ser operación o no, y similitud hace referencia a la ecuación (5).

4.1.5.4 Regla 4. Clasificación de entidad

Una vez que se tiene la caracterización morfosintáctica de las entidades, se realiza el proceso de desambiguación del sustantivo que compone la entidad (E_m) para encontrar el sentido correcto del sustantivo se utilizaron igualmente, los métodos Simplified Lesk, MFS y UKB, por tanto, una entidad queda definida por un: (i) identificador que permite determinar que el concepto es el mismo independiente del idioma, y (ii) identificador universal del sentido. Dado lo anterior, queda asignado el sentido al sustantivo que representa la entidad.

4.1.6 Caracterización semántica

El módulo de caracterización semántica recibe el listado de objetivos, agentes, operaciones, entidades desambiguados por el módulo “Proceso de desambiguación”, luego realiza la caracterización semántica utilizando técnicas computacionales que permiten medir la similitud semántica entre conceptos.

4.1.7 Validación elementos KAOS

Para lograr la validación de los elementos básicos del diagrama de objetivos de KAOS, el usuario deberá haber tenido una primera interacción con el sistema (NL2KAOS) para ingresar la especificación (utilizando las reglas morfosintácticas propuestas en esta tesis) de requisitos de software en lenguaje natural textual, asimismo, éste (el usuario) deberá seleccionar entre las técnicas que NL2KAOS le ofrece para realizar el proceso de desambiguación y la caracterización semántica, el sistema procesa el texto y lo evalúa aplicando las reglas sintácticas y semánticas. Una vez realizada la caracterización semántica, el módulo de validación recibe el listado de objetivos, agentes, operaciones, entidades y

atributos caracterizados, en este módulo, el usuario podrá interactuar nuevamente con el sistema, puesto que, éste (el sistema) le despliega un formulario que lista los elementos caracterizados y le ofrece la posibilidad al usuario de aprobar y/o eliminar aquellos que considere no relevantes o no necesarios para el modelo a desarrollar.

4.1.8 Salida

El módulo denominado salida recibirá los elementos validados y permitirá: (i) exportar el diagrama de objetivos de KAOS en términos de la jerarquía definida por el usuario, (ii) realizar instancias de la ontología KAOS, en OWL, desarrollada en el marco de esta tesis, o (iii) exportar un archivo de texto con los resultados, incluyendo el sentido de cada palabra que se obtuvo en la desambiguación.

4.2 Conclusiones del capítulo

El alcance del modelo propuesto tiene como propósito lograr la identificación de los elementos básicos (objetivo, entidad, agente y operación) del diagrama de objetivos de KAOS, minimizando la ambigüedad semántica de tipo polisémica presente en la especificación presentada a través de lenguaje natural textual. Para ello, se presentó en este capítulo el modelo general de la solución propuesta.

Dado lo anterior, y con el objetivo de brindar una solución a la necesidad de identificar la consistencia que debe existir entre el lenguaje natural y los elementos básicos del diagrama de objetivos de KAOS, se definieron en este capítulo un conjunto de formas morfosintácticas que permiten definir una serie de reglas morfosintácticas para identificar los descriptores de los elementos básicos del diagrama de objetivos de KAOS a partir de una especificación (utilizando las

reglas morfosintácticas propuestas en esta tesis) de requisitos de software en lenguaje natural textual.

Adicionalmente, se presentaron las reglas que permiten la identificación y clasificación semántica de los elementos básicos que conforman el diagrama de objetivos de KAOS, las cuales tienen como propósito principal, realizar la traza y el proceso completo de identificación morfosintáctico + semántico.

El aporte generado en este capítulo, permite mejorar la comunicación entre el interesado y el analista en la primera fase del ciclo de vida de software a través de la creación de un conjunto de formas y reglas morfosintácticas que permiten aproximar el lenguaje técnico del analista con el lenguaje natural del interesado. Asimismo, logra minimizar la ambigüedad semántica de tipo polisémica (al aplicar reglas morfosintácticas y semánticas) presente en el lenguaje natural textual, la cual afecta de forma negativa el ciclo de vida de software;

Capítulo 5

Arquitectura y diseño del prototipo propuesto

Este capítulo presenta los componentes que conforman la arquitectura del sistema (NL2KAOS), desarrollado en el marco de esta tesis, el cual implementa el modelo propuesto en el capítulo 4 orientado a la obtención de los elementos básicos (objetivo, entidad, agentes y operación) del diagrama de objetivos KAOS a partir de especificaciones textuales en idioma español. Adicionalmente, se especifican los elementos de diseño que se contemplaron en la programación de dicho sistema (software), donde se destacan los diagramas de casos de uso, clases y secuencias. Este capítulo pretende satisfacer el tercer objetivo de esta tesis, el cual indica: implementar un ambiente de prueba (prototipo) que haciendo uso del modelo de procesamiento terminológico propuesto permita su valoración.

5.1 Arquitectura

La arquitectura general del sistema desarrollado para la validación del modelo propuesto en el capítulo 4, se presenta en la figura 5.1. Asimismo, en la figura 5.2 se puede observar el diagrama de clases de la solución propuesta.

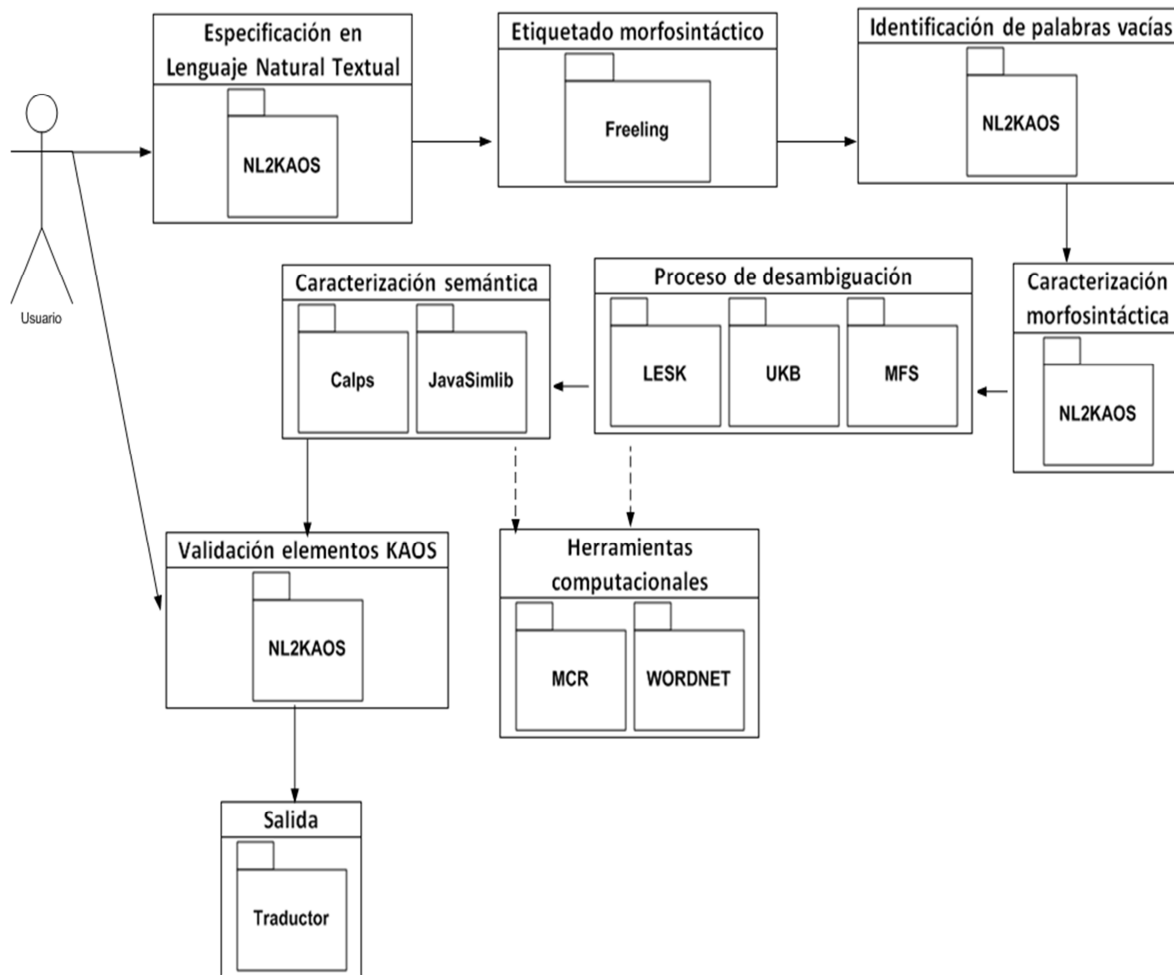


Figura 5.1. Arquitectura general del modelo propuesto. Fuente: elaboración propia.

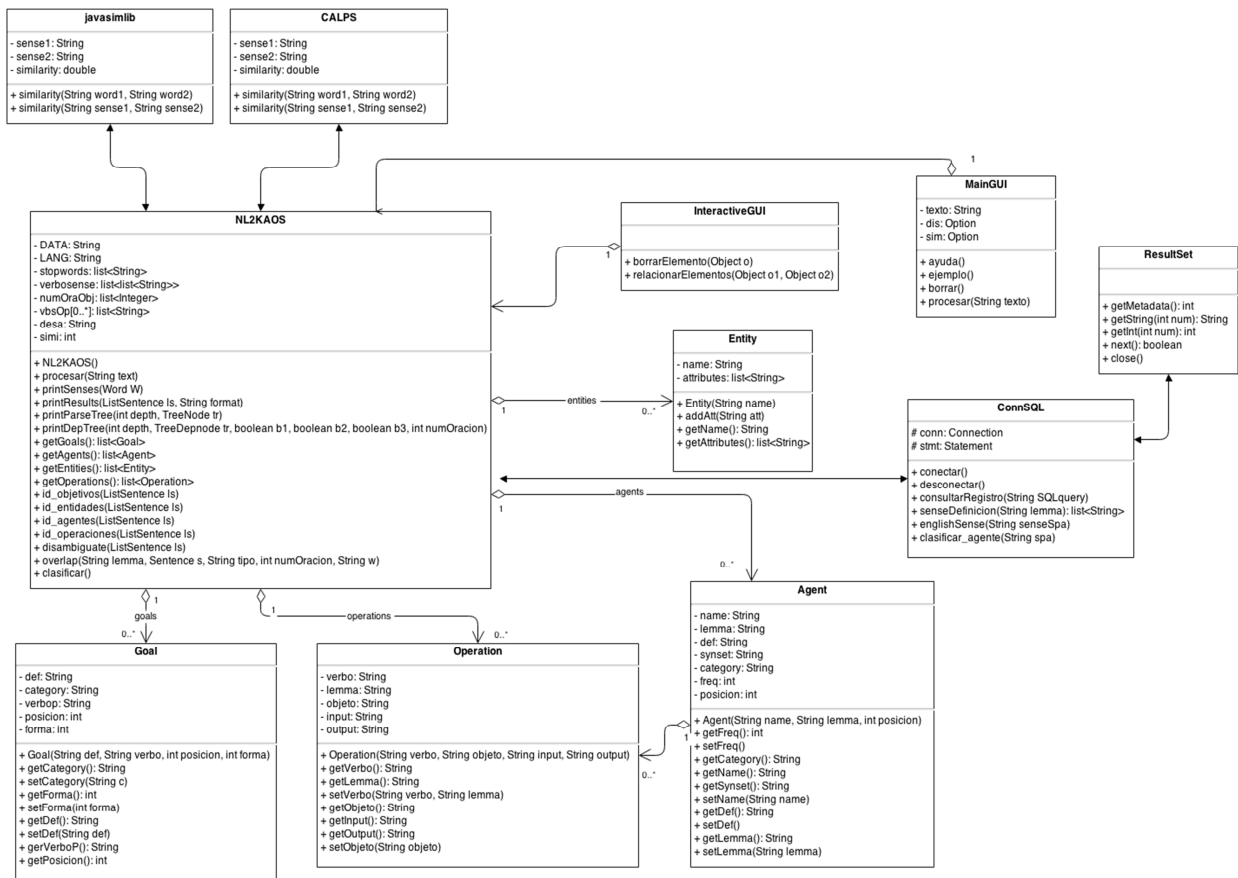


Figura 5.2. Diagrama de clases del modelo propuesto. Fuente: elaboración propia

A continuación se describen los módulos que componen la arquitectura del modelo propuesto:

4.1.1. Especificación en Lenguaje Natural Textual

El diagrama de clases perteneciente a este módulo se puede observar en la figura 5.2., el proceso desarrollado en este módulo, utiliza específicamente la clase MainGUI, la cual representa la página web de inicio de la aplicación (ver anexo “D” casos de uso), en ésta, se capturan los siguientes datos que necesita la aplicación: (i) especificación textual de los requisitos, (ii) forma de desambiguación (dis), y (iii) forma de medir la similitud semántica para la caracterización semántica

(sim). Estos datos se pasan a la clase controladora principal NL2KAOS. Ésta, a su vez, hace uso de paquetes para el etiquetado morfosintáctico del texto, como se explica en el siguiente módulo.

5.1.2 Etiquetado morfosintáctico

La comunidad científica de PLN, utiliza diferentes técnicas y paquetes que son utilizados para la desambiguación semántica. En esta tesis, se seleccionó el paquete Freeling 3.1, dado que, es el más completo para este tipo de procesos asociados al idioma español (Padró y Stanilovsky, 2012). En la figura 5.3, se puede observar el diagrama de clases de este paquete. Las clases básicas en esta librería son usadas para representar datos lingüísticos (palabras, etiquetas, oraciones, árboles sintácticos, documentos, etc.) que se obtienen del análisis realizado.

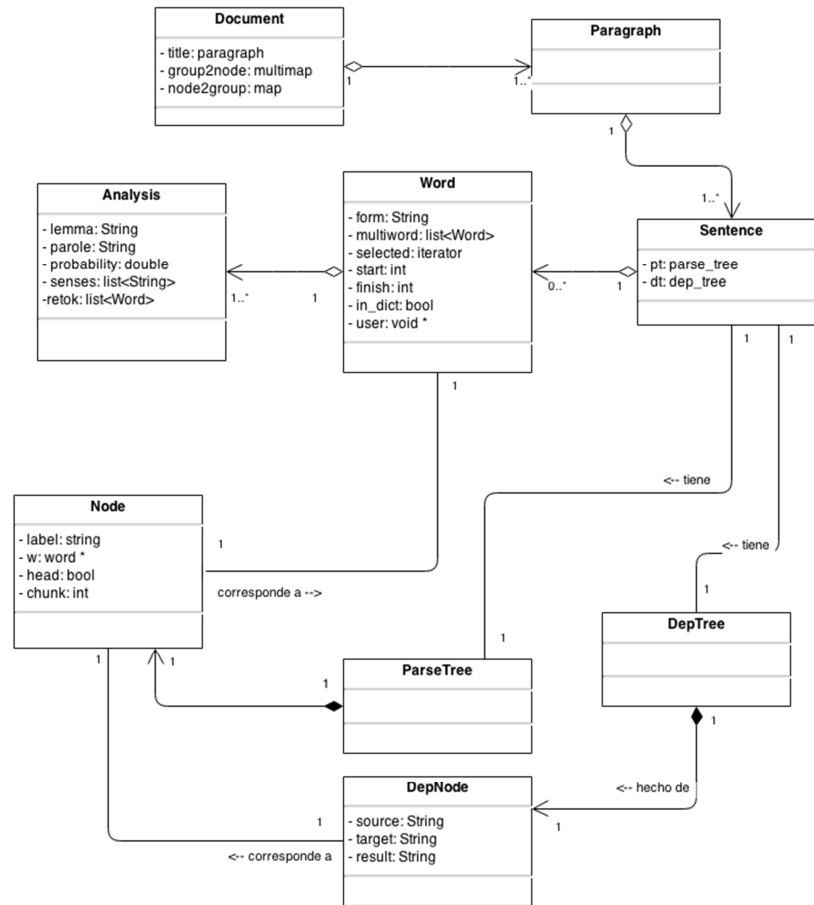


Figura 5.3. Diagrama de clases del Freeling. Fuente: adaptada de (Padró y Stanilovsky, 2012).

Las clases básicas de freeling utilizadas en este modelo se detallan a continuación:

Analysis: utiliza una tupla <lemma, PoS tag, probability, sense list> que hacen referencia al lexema, etiquetas morfosintácticas (PoS), probabilidad de ocurrencia y lista de sentidos, respectivamente.

Word: es una forma de expresión con una lista de posibles objetos de la clase *Analysis*. Es decir, cada expresión es susceptible de tener un lema, etiqueta PoS, probabilidad y lista de sentidos, como se indica en la figura 5.3.

Sentence: lista de objetos de la clase *Word* que expresan pertenencia a una oración completa, la cual puede incluir un árbol de sintáctico y de dependencias.

Paragraph: utiliza una lista de objetos de la clase *Sentence*, interpretados como un párrafo independiente.

Document: utiliza una lista de objetos de la clase *Paragraph*, que forman un documento completo.

Adicional a las clases básicas que contienen datos lingüísticos, Freeling posee otras clases para transformar los textos de entrada, en textos más completos en información lingüística. En las figuras 5.4 y 5.5 se presenta el diagrama clases de procesamiento y la estructura de la clase abstracta *processor* utilizadas en este proceso, en éstas, se puede observar que la mayoría de las clases heredan de la clase abstracta *processor*.

```
class processor {
public:
    /// constructor
    processor();
    /// destructor
    virtual ~processor() {};

    /// analyze and enrich sentence
    virtual void analyze(sentence &)=0;
    /// analyze and enrich sentences in list
    void analyze(std::list<sentence> &);
    /// analyze sentence , return enriched copy
    sentence analyze(const sentence &);
    /// analyze sentences in list , return enriched copy
    std::list<sentence> analyze(const std::list<sentence> &);
};
```

Figura 5.5. Estructura de la clase abstracta *processor*. Fuente: (Padró y Stanilovsky, 2012).

Las demás clases utilizan los métodos heredados de la clase *processor* para facilitar el proceso de análisis en una oración (*sentence*) o lista de oraciones (list *sentence*), además, para obtener los resultados del procesamiento de los datos originales. Las características de estas clases se detallan a continuación:

Morfo: recibe una lista de *sentence* y etiqueta morfológicamente cada palabra *Word* en cada oración de la lista. Esta clase es un meta-módulo que aplica un conjunto de procesadores especializados: (i) detección de entidades de nombre (*ner*), (ii) números (*numbers*), (iii) fecha (*dates*), (iv) puntuación (*punts*), (v) multi-palabras (*locutions*), (vi) búsqueda en el diccionario (*dictionary*), (vii) unidades de medida físicas, monetarias, entre otras (*quantities*), (viii) etiquetador de probabilidades y palabras desconocidas (*probabilities*), y (ix) reconocedor de nombres propios (*ner*).

Tagger: recibe una lista de *sentence* y etiqueta morfosintácticamente cada palabra. Esto lo logra con el etiquetador de Part of Speech (PoS) presentado en (Brants, 2000), el cual es un etiquetador basado en Modelos Ocultos de Markov (HMM), que aprende reglas estadísticas de corpus previamente etiquetados.

NE classifier: recibe una lista *sentence* y clasifica cada palabra etiquetada que sea un nombre propio, con base en el sistema CoNLL-2002 (Carreras, Marqués y Pradó, 2002).

Sense annotator: recibe una lista de *sentence* y agrega a cada palabra la información semántica, a través de la búsqueda en el diccionario de Freeling de los synsets respectivos, es decir, de los sentidos asociados a esa palabra.

Word sense disambiguator: recibe una lista de *sentence* y ordena los sentidos de acuerdo al análisis seleccionado en el contexto dado. El Freeling incluye desambiguación semántica con MFS y UKB.

Chunk parser: recibe una lista de *sentence* y crea el árbol sintáctico asociado a la misma. Ésta es una implementación presentada en (Atserias y Rodríguez, 1998).

Dependency parser: recibe una lista de *sentence* analizadas previamente, y la perfecciona utilizando árbol de dependencias. Para ello, se aplican en primera instancia las reglas de completado para transformar el árbol sintáctico en uno de dependencias, luego, se etiquetan las funciones y el árbol es convertido en dependencias. Este módulo es una extensión del trabajo descrito en (Atserias, Comelles y Mayor, 2005).

5.1.3 Identificador de palabras vacías

Las palabras vacías son un conjunto de palabras sin significado en sí mismo, como lo son los artículos, pronombres, preposiciones, entre otros. Para identificarlas en el documento, se toman los lexemas de todas las palabras, puesto que, la lista considera todos los casos posibles de palabras vacías asociadas a los lexemas. Luego se compara cada una con la lista de palabras vacías del español que aparecen en (Tartarus, 2013). Las palabras que estén en ambos conjuntos se almacenan en una lista denominada {stopwords_encontradas}. La lista de palabras vacías se define al comienzo de la ejecución del programa, asignándole los respectivos valores al atributo *stopwords*, dentro de la clase NL2KAOS.

5.1.4 Caracterización morfosintáctica

Todas las reglas que se definieron en el capítulo 4 para la identificación morfosintáctica de los elementos KAOS, son implementadas en este módulo. Después de tener todo el texto etiquetado morfosintácticamente, se invocan los métodos *id_objetivos()*, *id_entidades()*, *id_agentes()*, *id_operaciones()*, los cuales, con base en la estructura morfosintáctica y las reglas definidas, pueden hacer

instancias de los elementos identificados. Dado lo anterior, se puede obtener una lista de objetos de las clases Goal, Entity, Agent y Operation.

5.1.5 Proceso de desambiguación

Este proceso se presenta en la arquitectura general en módulos diferentes, sin embargo, los procedimientos utilizados pertenecen al paquete NL2KAOS (para el caso del algoritmo Lesk), y al paquete de Freeling (para el caso de las técnicas MFS y UKB). Las técnicas utilizadas en esta tesis para el proceso de desambiguación, fueron definidas en el capítulo 2.

Para el caso de esta tesis, el Lesk simplificado fue implementado en el modelo propuesto siguiendo el algoritmo propuesto en (Lesk, 1986).

Algoritmo de Lesk

Dadas dos palabras, W_1 y W_2 , cada una con sus respectivos sentidos N_{W_1} y N_{W_2} definidos en un diccionario, para cada par de posibles sentidos W_1^i y W_2^j , $i = 1..N_{W_1}$, $j = 1..N_{W_2}$, primero se determina el solapamiento con las correspondientes definiciones contando el número de palabras que tienen en común. A continuación, el par de sentidos con mayor solapamiento es seleccionado y entonces se le asigna un sentido a cada palabra del par inicial.

1. Para cada sentido i de W_1
2. Para cada sentido j de W_2
3. Calcular el solapamiento(i, j), el número de palabras en común entre las definiciones del sentido i y el sentido j
4. Encontrar i y j tales que el solapamiento(i, j) sea el máximo
5. Asignar el sentido i a W_1 y el sentido j a W_2

Asimismo, se consideraron en este proceso las técnicas presentadas en (Ng, Lee, 1996), que utiliza el mejor sentido de acuerdo con el puntaje más alto que haya obtenido una expresión en los diferentes contextos y (Agirre y Soroa, 2009), que utiliza la técnica de puntos de grafo y el mejor sentido de una expresión. Estas técnicas de desambiguación semántica, vienen programadas en el Freeling a través del MFS y UKB. Para obtener mayor información del funcionamiento de esta técnicas, se recomienda consultar (Agirre y Soroa, 2009). Finalmente, para lograr el proceso de desambiguación, se utilizan en este módulo el: (i) MCR como base de datos léxica en idioma español de relaciones semánticas entre términos (Atserias et al. 2004), y (ii) WORDNET, definido por (Fellbaum, 1998), como una Ontología que establece jerarquías y relaciones semánticas entre términos. Ambas herramientas computacionales fueron definidas en el capítulo 2 de esta tesis. A continuación, se describen los algoritmos utilizados en este proceso.

Algoritmo MFS

Este algoritmo funciona utilizando como insumo de entrada un corpus y, para cada expresión de éste, le encuentra todos los sentidos con que se etiquetó, luego, cuenta la frecuencia de cada sentido, para finalmente, entregar el más frecuente.

MFS (palabra): retorna sentidos

Leer corpus; //etiquetado con el sentido para cada palabra

Para cada lema del corpus, Contar los sentidos existentes para ese lema

Ordenar por sentido más frecuente

Retornar los sentidos de la palabra dada

Fin MFS

Algoritmo UKB

Sea W_i , $i = 1 \dots m$, las palabras de contenido (nombres, verbos, adjetivos, etc.), que tienen una entrada en el diccionario y estén relacionados con los conceptos LKB (*Lexical Knowledge Base*). Sea $\text{Conceptos}_i = \{v_1, v_2, \dots, v_{im}\}$ los i_m -concepto asociados a la palabra W_i en el grafo LKB.

La idea principal del UKB es extraer el subgrafo de G_{KB} cuyos vértices y relaciones sean relevantes para un contexto dado. Tal subgrafo es llamado G_D (*Desambiguation Subgraph*) y se construye de la siguiente manera: Para cada palabra W_i del contexto de entrada y cada concepto $v_i \in \text{Conceptos}_i$, un algoritmo BFS (*Breath-First Search*) sobre G_{KB} es realizado, iniciando en el nodo v_i . Cada ejecución del BFS calcula la ruta más corta entre v_i y el resto de conceptos de G_{KB} (rmc_{vi}). La ejecución del BFS se repite para cada concepto de cada palabra en el contexto de entrada. El grafo de desambiguación (G_D) es entonces sólo la unión de los vértices y aristas de las rutas más cortas.

Una vez que el grafo G_D está construido, se ejecuta el algoritmo PageRank sobre él. Como es habitual, el paso de desambiguación se realiza asignando a cada palabra W_i el concepto asociado en Conceptos_i que tiene el máximo puntaje.

UKB (palabra): retorna sentidos

Leer corpus; //etiquetado con el sentido para cada palabra

Para cada lema del corpus

$\text{Conceptos}_i = \text{Synsets}(\text{lema})$ //Obtener los Conceptos_i para ese lema

$G_{KB} = \text{ConceptosRelevantes}(\text{corpus})$

Para cada v_i en Conceptos_i

$rmc = \text{BFS}(v_i, G_{KB})$

$G_D(\text{lema}) = rmc$

$\text{PageRank}(G_D)$

Retornar los sentidos de la palabra dada

Fin UKB

5.1.6 Caracterización semántica

En este módulo se aplican las reglas semánticas definidas en el capítulo 4. Al igual que en el paso anterior, el proceso utilizado no se configura como un módulo aparte, dado que, dentro del paquete NL2KAOS, se realiza el procesamiento de las reglas respectivas. Sin embargo, para aplicarlas, es indispensable el cálculo de una medida de similitud semántica. Para calcular esta medida, se utilizaron en esta tesis las siguientes técnicas de similitud semántica:

- (i) *Adapted Lesk*: incorporada en el paquete CALPS (Hope, 2008). Aunque esta técnica es usada para realizar desambiguación semántica, también es utilizada para evaluar la similitud semántica entre dos conceptos. El propósito general de esta técnica, consiste en medir el overlap, es decir, la mayor cadena de palabras iguales que hay en las definiciones de cada uno de los conceptos. Además de medir la similitud entre conceptos y sus synsets, también considera los hiperónimos, hipónimos, sinónimos y las relaciones de atributo asociados en ambos conceptos. El mayor overlap será la medida de similitud semántica entre ambos conceptos.
- (ii) *Jiang y Conrath con IC (information content)*: esta técnica está basada en el contenido de la información (IC) (Jiang y Conrath, 1998), además, está incorporada en el paquete Javasilib (Seco et al., 2004). El punto de partida de esta técnica, se inicia tras reconocer que WordNet puede ser usada como una fuente estadística que no necesita fuentes externas. De hecho, los autores señalan que la estructura taxonómica de WordNet está organizada en una forma significativa, donde los conceptos con muchos hipónimos poseen menos información que los conceptos hoja. Por su parte, los conceptos hoja son los que tienen mayor especificidad en la taxonomía, de manera que la información que expresan es mayor. Dado lo anterior, se plantea la siguiente ecuación

para obtener la cantidad de información de un concepto, como una función que depende de la cantidad de hipónimos que posee.

$$ic_{wn}(c) = \frac{\log(\frac{hypo(c)+1}{max_{wn}})}{\log(\frac{1}{max_{wn}})} = 1 - \frac{\log(hypo(c) + 1)}{\log(max_{wn})}$$

Donde la función *hypo* devuelve el número de hipónimos de un concepto, y *maxwn* es una constante que se establece como el máximo número de conceptos que existen en la taxonomía u ontología. El denominador, que es equivalente al valor del concepto con más información, sirve como un factor normalizador, en la medida en que asegura que la cantidad de información está en $[0, \dots, 1]$.

Con el propósito de utilizar este índice dentro de las medidas de similitud semántica, se normaliza aplicando una transformación lineal a la fórmula de Jiang y Conrath, obteniendo una función de similitud. La fórmula resultante es:

$$sim_{jcn}(c1, c2) = 1 - \left(\frac{ic_{wn}(c1) + ic_{wn}(c2) - 2 \times sim_{res'}(c1, c2)}{2} \right)$$

Para este caso, *simres'* corresponde a la función de similitud de Resnik definida en (Resnik y Yarowsky, 1999) y adapta a los valores de IC obtenidos.

5.1.7 Validación elementos KAOS

En este módulo, se realiza la interacción con el usuario, mediante la disposición de un formulario interactivo, como se presenta en los casos de uso (ver anexo D). Para ello, se crea una clase *InteractiveGUI*, que tiene relación directa con la instancia de la clase principal *NL2KAOS*, permitiéndole al usuario remover los

elementos que considera no pertinentes para el modelo. Asimismo, le permite al usuario definir la jerarquización del diagrama de objetivos.

5.1.8 Salida

Finalmente, este módulo permite obtener el diagrama KAOS construido, en formatos XML (válido para Objectiver), la ontología KAOS en formato OWL, o en un archivo con extensión TXT que contiene las palabras desambiguadas.

5.2 Conclusiones del capítulo

El alcance de la arquitectura propuesta, tiene como propósito presentar los componentes que conforman sistema (NL2KAOS), desarrollado en el marco de esta tesis. En este capítulo se presentaron los diagramas de clases y los componentes que conforman la arquitectura para satisfacer el modelo propuesto en el capítulo 4. Asimismo, se detallaron los módulos programados y la forma en que se interactúa con paquetes y repositorios externos a la aplicación. Además, se presentaron los casos de uso y un fragmento del diagrama de secuencias (ver anexo D) correspondientes a la arquitectura propuesta.

El aporte de este capítulo, está orientado a representar el modelo conceptual a través del nivel tecnológico para posibilitar la validación del modelo propuesto en esta tesis doctoral. Además, ofrece un proceso interactivo que permite mejorar la especificación del software futuro con el apoyo de la metodología KAOS.

Capítulo 6

Validación del modelo

En este capítulo se presenta la evaluación del modelo propuesto en el capítulo 4, para ello, se hace uso del prototipo desarrollado en esta tesis. Asimismo, se identifican las fortalezas y debilidades de dicho modelo. Particularmente, se registran los resultados de experimentación realizados que se enfocan en valorar los aspectos de exhaustividad, precisión, completitud, correctitud y consistencia del modelo frente a la identificación de requisitos extraídos de su especificación en lenguaje natural en español. Con este capítulo se pretende satisfacer el cuarto objetivo que indica: validar el modelo de procesamiento terminológico propuesto mediante la utilización de casos de prueba con el fin de valorar sus fortalezas y debilidades.

Los resultados que se exponen en este capítulo, se obtuvieron utilizando un computador con las siguientes especificaciones técnicas: Hardware (Procesador Intel CORE i3 - 2.4 GHz con 3 GB de RAM); y (ii) Software (Sistema operativo Ubuntu Desktop 12.1).

6.1 Variables a medir

Existen tres factores que son fundamentales, especialmente en las primeras etapas de la educación de requisitos de software, como es el caso del presente estudio. Estos factores son la Completitud, la Correctitud y la Consistencia (Zowghi y Gervasi, 2004). La validación de estos tres factores en el modelo propuesto, se detallan en la sección 6.2.3.

Por otra parte, (Baeza y Berthier, 1999), establecen varias métricas para medir la exhaustividad y la precisión de modelos de recuperación de información. A continuación se detalla la forma en que se van a medir estos dos factores (exhaustividad y precisión) con relación al modelo propuesto en esta tesis.

6.1.1 Exhaustividad

Permite determinar la fracción de elementos que han sido hallados por el modelo propuesto frente a los realmente presentes (identificados y validados previamente). La exhaustividad se halla para cada elemento (objetivo, entidad, operación y agentes) de acuerdo con la siguiente expresión adaptada de (Baeza y Berthier, 1999).

$$Exhaustividad_{elemento} = \frac{|Elementos\ hallados \cap Elementos\ presentes|}{|Elementos\ presentes|} \times 100$$

Este índice se refiere a la exhaustividad, dado que evalúa la proporción entre aquellos elementos que el sistema halla y los que realmente están presentes (identificados por el analista). Una exhaustividad alta indicaría que casi todos los elementos indicados por el analista, son identificados por el sistema (NL2KAOS). Sin embargo, es posible que el sistema encuentre más del 100% de los elementos hallados por el analista, en ese sentido, este índice no logra reflejar esta última situación, la cual se resuelve con la precisión que se explica en el siguiente numeral.

6.1.2 Precisión

Otro aspecto a tener en cuenta es la fracción de elementos que el sistema halló frente al conjunto de elementos realmente presentes. Este índice se conoce como

precisión, y también debe realizarse ponderando los resultados de cada uno de los elementos básicos del diagrama de objetivos de KAOS.

$$Precisión_{elemento} = \frac{|Elementos\ hallados \cap Elementos\ presentes|}{|Elementos\ hallados|} \times 100$$

La precisión indica qué tanto de los elementos hallados y presentes según el analista, están en la cantidad de elementos hallados por el sistema (NL2KAOS). Si el sistema halla la misma cantidad de elementos frente a los realmente presentes se puede indicar que la precisión es del 100%, sin embargo, si el sistema halla más elementos con relación a los realmente presentes, la precisión sería menor al 100%.

6.2 Experimentación

6.2.1 Introducción

El modelo propuesto se validó con los siguientes tres casos de estudio, los cuales han sido descritos previamente en la literatura: (i) el sistema de servicio de ambulancias de Londres (Heaven y Finkelstein, 2004) y (Zapata, Lezcano y Tamayo, 2007), (ii) sistema asociado a un elevador (Respect IT, 2007), y (iii) el sistema de una pizzería (Zapata, Villegas y Arango, 2006) y (Zapata, Lezcano y Tamayo, 2007). Asimismo, se realizaron siete nuevos casos de estudios, los cuales fueron analizados por un grupo de analistas expertos identificando los valores reales de las variables asociadas a los requisitos de dichos casos con el fin de evaluar los valores obtenidos por el modelo. La descripción de los casos de estudio del estado del arte y los siete nuevos casos se detallan en el anexo B.

6.2.2 Resultados y Análisis

La validación se realiza evaluando la exhaustividad y la precisión comparando para los casos de estudio los resultados arrojados por el sistema (NL2KAOS) versus los resultados reales obtenidos por un analista experto de requisitos.

Además, se evalúan las técnicas utilizadas en este modelo para realizar la desambiguación y medir la similitud semántica. Para la desambiguación se consideraron las siguientes técnicas: (i) Mejor sentido y puntaje más alto, y (ii) Puntos de grafo y mejor sentido. Para medir la similitud semántica se utilizaron las técnicas: (i) AdaptedLesk, y (ii) Jiang y Conrath con IC. Estas técnicas fueron presentadas en el capítulo 5 de esta tesis.

En las tablas 6.1 – 6.4 se presentan los resultados de exhaustividad y precisión para el caso de estudio de la ambulancia utilizando los métodos de desambiguación propuestos en esta tesis.

Tabla 6.1. Resultados de los objetivos identificados por el modelo para el caso de estudio de la ambulancia. Fuente: elaboración propia.

Objetivos reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Garantizar que la ambulancia llegue al lugar del incidente en el menor tiempo posible.	1. [Mantener] Garantizar que los auxiliares de la ambulancia lleguen al lugar del incidente en el menor tiempo posible.	1. [Mantener] Garantizar que los auxiliares de la ambulancia lleguen al lugar del incidente en el menor tiempo posible.	1. [Mantener] Garantizar que los auxiliares de la ambulancia lleguen al lugar del incidente en el menor tiempo posible.	1. [Mantener] Garantizar que los auxiliares de la ambulancia lleguen al lugar del incidente en el menor tiempo posible.

	2. [Mantener] Llamar telefónicas.	2. [Mantener] Llamar telefónicas.	2. [Mantener] Llamar telefónicas.	2. [Mantener] Llamar telefónicas.
Atender llamadas de emergencia.	3. [Mantener] Atender llamadas de emergencia.	3. [Lograr] Atender llamadas de emergencia.	3. [Mantener] Atender llamadas de emergencia.	3. [Lograr] Atender llamadas de emergencia.
[Lograr] Rellenar el formulario del incidente.	4. [Mantener] Rellenar el formulario del incidente.	4. Rellenar el formulario del incidente.	4. [Mantener] Rellenar el formulario del incidente.	4. Rellenar el formulario del incidente.
Indicar en el formulario la localización del incidente y la hora en la que la llamada fue hecha.	5. [Mantener] Indicar en el formulario la localización del incidente y la hora en la que la llamada fue hecha.	5. [Mantener] Indicar en el formulario la localización del incidente y la hora en la que la llamada fue hecha.	5. [Mantener] Indicar en el formulario la localización del incidente y la hora en la que la llamada fue hecha.	5. [Mantener] Indicar en el formulario la localización del incidente y la hora en la que la llamada fue hecha.
Examinar el formulario del incidente.	6. [Mantener] Examinar el formulario del incidente.	6. Examinar el formulario del incidente.	6. [Mantener] examinar el formulario del incidente.	6. Examinar el formulario del incidente.
Identificar a la ambulancia más cercana.	7. Identificar a la ambulancia más cercana.	7. Identificar a la ambulancia más cercana.	7. Identificar a la ambulancia más cercana.	7. Identificar a la ambulancia más cercana.
Seleccionar la ambulancia más cercana al lugar del incidente.	8. [Mantener] Seleccionar la ambulancia más cercana al incidente.	8. Seleccionar la ambulancia más cercana al incidente.	8. [Mantener] Seleccionar la ambulancia más cercana al incidente.	8. Seleccionar la ambulancia más cercana al incidente.
Proponer al operario de localización de recursos la ambulancia más apropiada.	9. Proponer al operario de localización de recursos la ambulancia más apropiada.	9. Proponer al operario de localización de recursos la ambulancia más apropiada.	9. Proponer al operario de localización de recursos la ambulancia más apropiada.	9. Proponer al operario de localización de recursos la ambulancia más apropiada.
Aceptar o rechazar propuesta del sistema CAD.	10. [Mantener] Aceptar o rechazar propuesta del sistema CAD.	10. Aceptar o rechazar propuesta del sistema CAD.	10. [Mantener] Aceptar o rechazar propuesta del sistema CAD.	10. Aceptar o rechazar propuesta del sistema CAD.

Proponer otra ambulancia	11. [Mantener] Proponer otra ambulancia, en caso de que el operario de localización de recursos rechace la propuesta inicial.	11. Proponer otra ambulancia, en caso de que el operario de localización de recursos rechace la propuesta inicial.	11. [Mantener] Proponer otra ambulancia, en caso de que el operario de localización de recursos rechace la propuesta inicial.	11. Proponer otra ambulancia, en caso de que el operario de localización de recursos rechace la propuesta inicial.
Llamar a los auxiliares de la ambulancia.	12. Llamar a los auxiliares de la ambulancia.	12 Llamar a los auxiliares de la ambulancia.	12 Llamar a los auxiliares de la ambulancia.	12. Llamar a los auxiliares de la ambulancia.
	13. Llamar telefónicas o por radio.	13. Llamar telefónicas o por radio	13. Llamar telefónicas o por radio.	13. Llamar telefónicas o por radio.
Dar datos al auxiliar de la ambulancia para llegar al lugar del incidente.	14. Darle los datos al auxiliar de la ambulancia para llegar al lugar del incidente.	14. Darle los datos al auxiliar de la ambulancia para llegar al lugar del incidente.	14. Darle los datos al auxiliar de la ambulancia para llegar al lugar del incidente.	14. Darle los datos al auxiliar de la ambulancia para llegar al lugar del incidente.
Guiar al personal de ambulancia hacia el lugar del incidente	15. [Mantener] Guiar al personal de ambulancia hacia el lugar del incidente.	15. Guiar al personal de ambulancia hacia el lugar del incidente.	15. [Lograr] Guiar al personal de ambulancia hacia el lugar del incidente.	15. Guiar al personal de ambulancia hacia el lugar del incidente.
Recibir información de la ambulancia seleccionada	16. [Mantener] Garantizar que el operador de radio reciba la información de la ambulancia seleccionada.	16. [Mantener] Garantizar que el operador de radio reciba la información de la ambulancia seleccionada.	16. [Mantener] Garantizar que el operador de radio reciba la información de la ambulancia seleccionada.	16. [Mantener] Garantizar que el operador de radio reciba la información de la ambulancia seleccionada.

Dar instrucciones de localización al auxiliar de ambulancia.	17. [Lograr] Dar instrucciones de localización al auxiliar de la ambulancia constantemente.	17. Dar instrucciones de localización al auxiliar de la ambulancia constantemente.	17. [Mantener] Dar instrucciones de localización al auxiliar de la ambulancia constantemente.	17. Dar instrucciones de localización al auxiliar de la ambulancia constantemente.
Reportar estado de operación al operador de radio	18. [Mantener] Reportar el estado de la operación al operador de radio.	18. [Mantener] Reportar el estado de la operación al operador de radio.	18. [Mantener] Reportar el estado de la operación al operador de radio.	18. [Mantener] Reportar el estado de la operación al operador de radio.
Hallados AND Presentes	16	16	16	16
#Elementos presentes	16	16	16	16
#Elementos hallados	18	18	18	18
exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	88,89%	88,89%	88,89%	88,89%

Tabla 6.2. Resultados de los agentes identificados por el modelo para el caso de estudio de la ambulancia. Fuente: elaboración propia.

Agentes reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Auxiliares de ambulancia.	1. [Agente de ambiente] Auxiliares de ambulancia.	1. [Agente de ambiente] Auxiliares de ambulancia.	1. [Agente de ambiente] Auxiliares de ambulancia.	1. [Agente de ambiente] Auxiliares de ambulancia.
Asistente de control.	2. [Agente de ambiente] Asistente de control.	2. [Agente de ambiente] Asistente de control.	2. [Agente de ambiente] Asistente de control.	2. [Agente de ambiente] Asistente de control.
Operario de localización de	3. [Agente de ambiente]	3. [Agente de ambiente]	3. [Agente de ambiente]	3. [Agente de ambiente] Operario

recursos.	Operario de localización de recursos.	Operario de localización de recursos.	Operario de localización de recursos.	de localización de recursos.
Sistema CAD.	4. [Agente de software] Sistema CAD.	4. [Agente de software] Sistema CAD.	4. Sistema CAD.	4. Sistema CAD.
Operador logístico.	5. [Agente de ambiente] Operador logístico.	5. [Agente de ambiente] Operador logístico.	5. [Agente de ambiente] Operador logístico.	5. [Agente de ambiente] Operador logístico.
Operador de radio.	6. [Agente de ambiente] Operador de radio.	6. [Agente de ambiente] Operador de radio.	6. [Agente de ambiente] Operador de radio.	6. [Agente de ambiente] Operador de radio.
	7. [Agente de ambiente] Auxiliar de ambulancia.	7. [Agente de ambiente] Auxiliar de ambulancia.	7. [Agente de ambiente] Auxiliar de ambulancia.	7. [Agente de ambiente] Auxiliar de ambulancia.
Hallados AND Presentes	6	6	6	6
#Elementos presentes	6	6	6	6
#Elementos hallados	7	7	7	7
Exhaustividad	100%	100%	100%	100%
Precisión	85,71%	85,71%	85,71%	85,71%

Tabla 6.3. Resultados de las operaciones identificadas por el modelo para el caso de estudio de la ambulancia. Fuente: elaboración propia.

Operaciones reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Recibir llamada.				
Ingresar datos.	1. Ingresar los datos.		1. Ingresar los datos.	
Leer datos.	2. Leer los datos.		2. Leer los datos.	1. Leer los datos.
Medir distancia.	3. Medir la distancia.	1. Medir la distancia.	3. Medir la distancia.	2. Medir la distancia.
Seleccionar ambulancia.	4. Seleccionar una ambulancia.		4. Seleccionar una ambulancia.	

Realizar llamada.				
Comunicar datos.	5. Comunicar los datos.		5. Comunicar los datos.	
Hallados AND Presentes	5	1	12	4
#Elementos presentes	7	15	15	15
#Elementos hallados	5	1	14	4
Exhaustividad	71,43%	6,67%	80,00%	26,67%
Precisión	100%	100%	85,71%	100%

Tabla 6.4. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio de la ambulancia. Fuente: elaboración propia.

Entidades con atributos reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
1. Ambulancia	1. Ambulancia	1. Ambulancia	1. Ambulancia	1. Ambulancia
-auxiliar	-auxiliar	-auxiliar	-auxiliar	-auxiliar
-personal	-personal	-personal	-personal	-personal
-información	-información	-información	-información	-información
-localización	2. Localización	2. Localización	2. Localización	2. Localización
	-instrucción	-instrucción	-instrucción	-instrucción
	-operario			
2. Instrucción				
-localización		-operario	-operario	-operario
	3. Operador	3. Operador	3. Operador	3. Operador
	-localización	-localización	-localización	-localización
3. Radio	4. Radio	4. Radio	4. Radio	4. Radio
-operador	-operador	-operador	-operador	-operador
	5. Control	5. Control	5. Control	5. Control
4. Emergencia	-asistente	-asistente	-asistente	-asistente
-localización	6. Emergencia	6. Emergencia	6. Emergencia	6. Emergencia
-llamada	-llamada	-llamada	-llamada	-llamada
5. Incidente	7. Incidente	7. Incidente	7. Incidente	7. Incidente

-formulario	-formulario	-formulario	-formulario	-formulario
-localización	-localización	-localización	-localización	-localización
	-lugar	-lugar	-lugar	-lugar
	8. Recurso	8. Recurso	8. Recurso	8. Recurso
	-localización	-localización	-localización	-localización
6. Sistema	9. Sistema	9. Sistema	9. Sistema	9. Sistema
-propuesta	-propuesta	-propuesta	-propuesta	-propuesta
7. Personal	10. Personal	10. Personal	10. Personal	10. Personal
-información	-información	-información	-información	-información
Hallados AND Presentes	15	15	15	15
#Elementos presentes	19	19	19	19
#Elementos hallados	25	25	25	25
Exhaustividad	78,95%	78,95%	78,95%	78,95%
Precisión	60%	60%	60%	60%

De acuerdo con lo anterior, las técnicas planteadas y utilizadas en este proyecto para realizar la desambiguación y medir la similitud semántica, no afectan la identificación de los elementos básicos (objetivo, entidad y agente) del diagrama de objetivos de KAOS, dado que el modelo semántico se definió para cumplir con la clasificación de los elementos. Para el caso de las operaciones, se debe corroborar si éstas tienen relación semántica con la lista de los verbos de operación presentados por (Jaramillo, Zapata y Arango, 2005). Dado lo anterior, las operaciones pueden cambiar de acuerdo con la técnica utiliza para medir la similitud semántica y la desambiguación. En las tablas 6.5 y 6.6 se presenta un compendio de los resultados obtenidos para los 10 casos de estudio. El detalle de los resultados obtenidos se puede observar en el anexo B.

Tabla 6.5. Resumen de resultados de objetivos, agentes y entidades, para los 10 casos de estudio. Fuente: elaboración propia.

CASO DE ESTUDIO	Objetivos		Agentes		Entidades	
	Exhaustividad	Precisión	Exhaustividad	Precisión	Exhaustividad	Precisión
AMBULANCIA	100,00%	88,89%	100,00%	85,71%	78,95%	60,00%
ASCENSOR	92,86%	86,67%	100,00%	57,14%	56,00%	48,28%
PIZZERÍA	93,75%	93,75%	100,00%	62,50%	64,29%	69,23%
CARREFOUR	100,00%	100,00%	100,00%	66,67%	86,67%	76,47%
COLÁCTEOS	100,00%	87,50%	100,00%	75,00%	100,00%	71,43%
COLCERÁMICAS	100,00%	100,00%	100,00%	87,50%	100,00%	68,00%
ENCA	100,00%	80,00%	100,00%	85,71%	100,00%	80,95%
GUAYACÁN	81,82%	69,23%	88,89%	80,00%	100,00%	58,82%
MARINOZZI	100,00%	85,71%	100,00%	83,33%	85,71%	75,00%
YAMAHA	100,00%	100,00%	100,00%	85,71%	100,00%	82,35%
Promedio	96,84%	89,18%	98,89%	76,93%	87,16%	69,05%
Desv. Est.	0,057	0,094	0,033	0,105	0,155	0,102

Con base en los resultados obtenidos a través de las pruebas realizadas, se puede indicar que: (i) los agentes y los objetivos se logran identificar con un alto grado de exhaustividad y precisión, (ii) en términos generales, las entidades tienen resultados inferiores con relación a los agentes y a los objetivos siendo mayor la exhaustividad que la precisión, evidenciando una posible falencia del sistema en este aspecto.

Con el propósito de comparar los resultados obtenidos para los 10 casos de estudio, frente a la exhaustividad y la precisión, se define el índice de palabras por elemento (IDPPE), el cual indica en promedio la cantidad de palabras utilizadas por cada elemento identificado.

$$IDPPE = \frac{|Palabras\ del\ texto|}{|Elementos\ identificados|}$$

En las figura 6.1 y 6.2 se compara este índice con relación a la exhaustividad y la precisión para los 10 casos de estudio utilizados en el experimento. Estos resultados muestran una tendencia a obtener una mayor exhaustividad y una mayor precisión al aumentar las palabras del texto con relación a los elementos identificados.

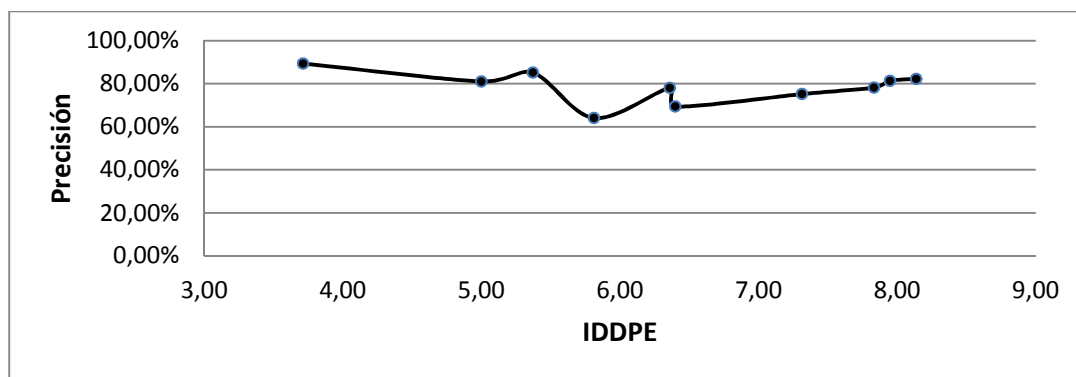


Figura 6.1 Precisión vs. IDPPE diez casos de estudio. Fuente: elaboración propia

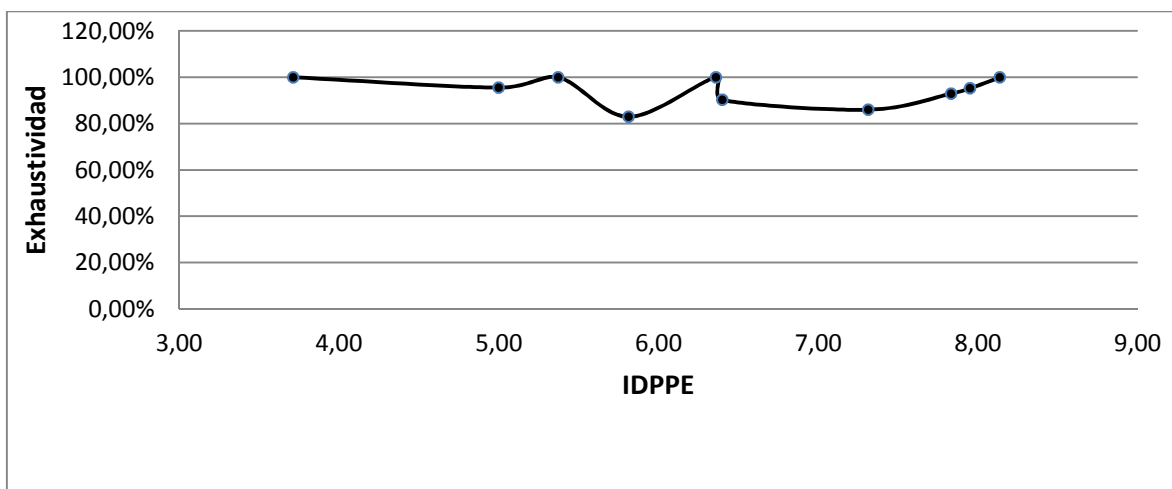


Figura 6.2. Exhaustividad vs. IDPPE diez casos de estudio. Fuente: elaboración propia.

Tabla 6.6. Resumen de resultados de operaciones para los diez casos de estudio.

Fuente: elaboración propia.

Las operaciones arrojan una exhaustividad promedio de 71,86%, y una precisión promedio de 75,85%.					
CASO DE ESTUDIO	PARÁMETRO	Técnica de desambiguación utilizada			
		Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
		Técnica de similitud semántica utilizada			
		AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
AMBULANCIA	Exhaustividad	71,43%	6,67%	80,00%	26,67%
	Precisión	100,00%	100,00%	85,71%	100,00%
ASCENSOR	Exhaustividad	66,67%	11,11%	66,67%	33,33%
	Precisión	85,71%	100,00%	85,71%	100,00%
PIZZERÍA	Exhaustividad	50,00%	12,50%	50,00%	12,50%
	Precisión	100,00%	100,00%	100,00%	100,00%
CARREFOUR	Exhaustividad	100,00%	50,00%	100,00%	100,00%
	Precisión	50,00%	50,00%	50,00%	66,67%
COLÁCTEOS	Exhaustividad	100,00%	50,00%	100,00%	50,00%
	Precisión	50,00%	100,00%	50,00%	50,00%
COLCERÁMICAS	Exhaustividad	100,00%	100,00%	100,00%	100,00%
	Precisión	33,33%	50,00%	33,33%	66,67%
ENCA	Exhaustividad	100,00%	50,00%	100,00%	100,00%
	Precisión	40,00%	100,00%	40,00%	66,67%
GUAYACÁN	Exhaustividad	100,00%	40,00%	100,00%	80,00%
	Precisión	83,33%	100,00%	83,33%	100,00%
MARINOZZI	Exhaustividad	66,67%	66,67%	66,67%	66,67%
	Precisión	40,00%	66,67%	40,00%	66,67%
YAMAHA	Exhaustividad	100,00%	100,00%	100,00%	100,00%
	Precisión	100,00%	100,00%	100,00%	100,00%

En la figura 6.3 se presentan de forma gráfica los resultados registrados en la tabla 6.6

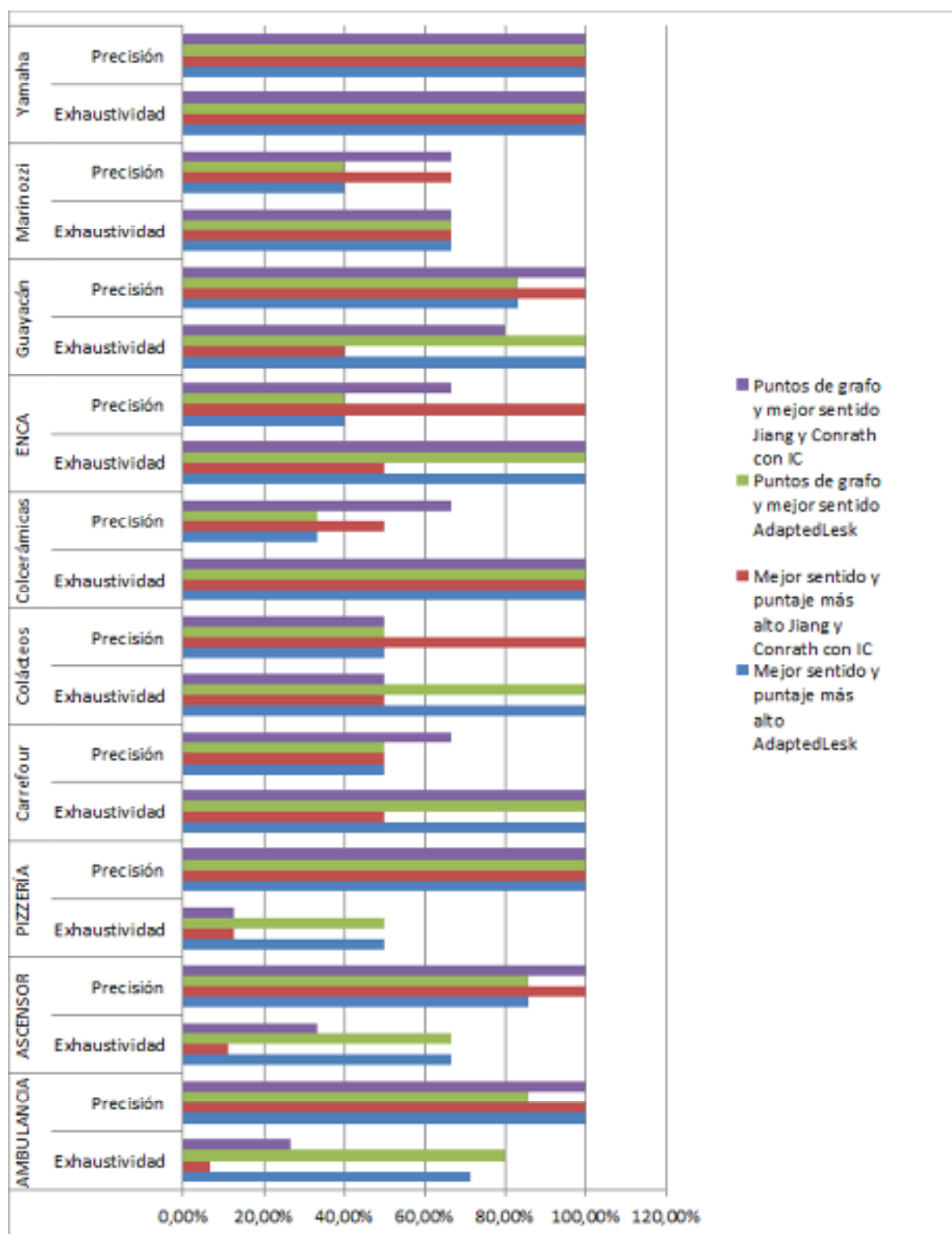


Figura 6.3. Resultados identificación de operaciones del experimento realizado a los 10 casos de estudio. Fuente: elaboración propia

En este análisis, se puede establecer que la técnica AdaptedLesk es superior en la exhaustividad frente a *Jiang y Conrath con IC*, por ejemplo, para el caso de la ambulancia, la primera obtiene un exhaustividad del 71,43%, mientras que en la segunda sólo es de 6,67%.

Como resultado importante del experimento se puede indicar que, al obtener un promedio general de los resultados arrojados por el sistema (software) NL2KAOS (desarrollado en el marco de esta tesis), para los 10 casos de estudio, se logra establecer una exhaustividad promedio del 88,69% y una precisión promedio del 77,75%. En consecuencia, se puede indicar que los resultados obtenidos son buenos, sin embargo, quedan aspectos por mejorar con relación a la exhaustividad y la precisión. Los resultados conseguidos, demuestran que es posible realizar un buen trabajo de identificación y clasificación de los elementos básicos del diagrama de objetivos de KAOS, utilizando técnicas de Procesamiento de Lenguaje Natural. En términos generales, se puede establecer que, el modelo propuesto en esta tesis, logra identificar satisfactoriamente los agentes y los objetivos. Sin embargo, las entidades y las operaciones constituyen la mayor falencia del modelo.

6.2.2.1 Modelo propuesto Vs. Ontología de dominio

Se realizó en este experimento una comparación (para el caso de estudio del servicio de ambulancia de Londres) de los conceptos identificados en el modelo propuesto frente a una Ontología de dominio, en dicha comparación se logró concluir que los conceptos identificados en el modelo mantienen una relación semántica del 95,65% frente a los conceptos definidos en la Ontología de dominio. Ver tablas 6.7 - 6.10.

Tabla 6.7. Comparativo de relación semántica asociada a los objetivos. Fuente: elaboración propia.

Objetivos	SPA	SENSE	SENSE cercanos a la Ontología de dominio
Garantizar que la ambulancia llegue al lugar del incidente en el menor tiempo posible.	spa-30-00889555-v	guarantee.3	guarantee.3
Llamar telefónicas	spa-30-01629958-v	arouse.3	arouse.3
Atender llamadas de emergencia.	spa-30-00080304-v	doctor.2	doctor.2
Rellenar el formulario del incidente.	spa-30-00172381-v	fill_out.2	
Indicar en el formulario la localización del incidente y la hora en la que la llamada fue hecha.	spa-30-01058574-v	comment.1	comment.1
Examinar el formulario del incidente.	spa-30-00627520-v	read.4	read.4
Identificar a la ambulancia más cercana.	spa-30-00652346-v	describe.4	describe.4
Seleccionar la ambulancia más cercana al lugar del incidente.	spa-30-00674607-v	choose.1	choose.1
Proponer al operario de localización de recursos la ambulancia más apropiada.	spa-30-00879540-v	nominate.1	nominate.1
Aceptar o rechazar propuesta del sistema CAD.	spa-30-00668805-v	accept.7	accept.7
Proponer otra ambulancia.	spa-30-00879540-v	nominate.1	nominate.1
Llamar a los auxiliares de la ambulancia.	spa-30-01629958-v	arouse.3	arouse.3
Llamar telefónicas o por radio	spa-30-01629958-v	arouse.3	arouse.3
Dar datos al auxiliar de la ambulancia para llegar al lugar del incidente.	spa-30-02200686-v	gift.2	gift.2
Guiar al personal de ambulancia hacia el lugar del incidente.	spa-30-01212230-v	draw.29	
Dar instrucciones de localización al auxiliar de ambulancia.	spa-30-00108604-v	give.44	give.44
Reportar estado de operación al operador de radio.	spa-30-00965035-v	account.3	account.3

Tabla 6.8. Comparativo de relación semántica asociada a los agentes. Fuente: elaboración propia.

Agentes	SPA	SENSE	SENSE cercanos a la Ontología
Auxiliar de ambulancia	spa-30-09780249-n	aide.2	aide.2
Asistente de control	spa-30-09608002-n	attendant.2	attendant.2
Operario de localización de recursos	spa-30-10318293-n	factory_worker.1	factory_worker.1
Sistema CAD	spa-30-04377057-n	system.1	system.1
Operador logístico	spa-30-10223294-n	jockey.2	jockey.2
Operador de radio	spa-30-10223294-n	jockey.2	jockey.2

Tabla 6.9. Comparativo de relación semántica asociada a las operaciones.

Fuente: elaboración propia.

OPERACIONES	SPA	SENSE	SENSE cercanos a la Ontología
Leer datos.	spa-30-00620239-v	read.10	read.10
Medir distancia.	spa-30-00647094-v	measure.1	measure.1
Seleccionar ambulancia.	spa-30-00674607-v	choose.1	choose.1
Comunicar datos.	spa-30-00720961-v	tell.5	tell.5

Tabla 6.10. Comparativo de relación semántica asociada a las entidades. Fuente: elaboración propia.

ENTIDADES	SPA	SENSE	SENSE cercanos a la Ontología
1. Ambulancia			
-auxiliar	spa-30-09780249-n	aide.2	aide.2
-personal			
-información	spa-30-05816287-n	information.2	information.2
2. incidente	spa-30-00959645-n	contretemps.1	contretemps.1
-lugar	spa-30-05611822-n	place.3	place.3
-formulario	spa-30-06473563-n	form.8	form.8
-localización	spa-30-00151087-n	catching.2	catching.2
3. control	spa-30-00803617-n	control.5	control.5
-asistente	spa-30-09608002-n	attendant.2	attendant.2
4. localización	spa-30-00151087-n	catching.2	catching.2

-operario	spa-30-10318293-n	factory_worker.1	factory_worker.1
-instrucción	spa-30-00887081-n	instruction.3	instruction.3
5. recurso	spa-30-05154908-n	aid.1	aid.1
-localización	spa-30-00151087-n	catching.2	catching.2
6. sistema	spa-30-04377057-n	system.1	system.1
-propuesta	spa-30-07161741-n	marriage_offer.1	marriage_offer.1
7. radio	spa-30-04041544-n	radio.3	radio.3
-operador	spa-30-10223294-n	jockey.2	jockey.2

6.2.2.2 Técnica utilizada Vs. Tiempo

Para realizar esta prueba se tomaron los 10 casos de estudio propuestos para la experimentación, en ésta, se analizaron los tiempos utilizados por cada una de las técnicas de similitud semántica y desambiguación planteadas en el modelo. En la tabla 6.11 se pueden observar los resultados obtenidos.

Tabla 6.11. Tiempo de procesamiento de acuerdo con la técnica de desambiguación utilizada.

CASO DE ESTUDIO	Tiempo (segundos)			
	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
AMBULANCIA	581,23	11,65	721,62	15,58
ASCENSOR	791,78	12,85	1035,99	29,97
PIZZERÍA	240,28	21,93	437,47	16,89
CARREFOUR	525,29	6,70	1162,52	3,76
COLÁCTEOS	283,03	3,69	337,86	3,47
COLCERÁMICAS	831,32	4,32	1324,81	5,32
ENCA	228,57	4,13	161,97	4,26
GUAYACÁN	907,49	8,33	1009,73	8,99
MARINOZZI	471,95	4,10	452,00	3,65
YAMAHA	114,35	1,96	213,93	3,19
PROMEDIO	497,53	7,97	685,79	9,51

De acuerdo con los resultados presentados en la tabla 6.7, se puede inferir que cuando se utiliza la técnica de similitud semántica *AdaptedLesk* en el modelo, el tiempo utilizado en el proceso es mayor con relación a la técnica *Jiang y Conrath con IC*. Sin embargo, es importante resaltar que la mayor cantidad de tiempo se toma durante la clasificación de los objetivos y sólo una pequeña fracción se utiliza para identificar las operaciones.

6.2.3 Validación factores de calidad

A continuación se presentan los tres factores de calidad: (i) Consistencia, (ii) Completitud, y (iii) Corrección (Zowghi y Gervasi, 2004), definidos de la siguiente manera:

Completitud (cantidad de elementos obtenidos): la validación de este factor se realizará a partir de la especificación textual. Se identifican los elementos básicos (objetivo, entidad, operación y agente) que sirven como base para la evaluación del modelo propuesto.

Correctitud (uso adecuado de la sintaxis): la Correctitud se logra con el uso adecuado de la sintaxis, es decir, que los elementos básicos obtenidos sean consistentes con los conceptos utilizados en la metodología de KAOS para lograr su jerarquización.

Consistencia (adecuación con la especificación): la validación de este factor se realizará a partir la especificación textual, los elementos básicos (objetivo, entidad, operación y agente) del diagrama de objetivos de KAOS identificados deberán corresponder con la especificación textual de los requisitos.

La validación de los tres factores en el modelo propuesto, se confrontaron frente a los resultados obtenidos por otros autores. Puntualmente, se analizaron los

siguientes casos de estudio: (i) el sistema de servicio de ambulancia de Londres, y (ii) Sistema de la pizzería.

6.2.3.1 Experimento

El experimento se realizó con base con en los casos de estudio del sistema de servicio de ambulancia de Londres presentado por (Heaven y Finkelstein, 2004) y (Zapata et al., 2007) y para el sistema de la pizzería (Zapata et al., 2006) y (Zapata et al., 2007).

6.2.3.1.1 Resultados y Análisis

Sistema de servicio de ambulancia de Londres.

Compleitud: en las tablas 6.12 - 6.14, se puede observar la evaluación comparativa entre los autores sobre completitud.

Tabla 6.12. Evaluación comparativa completitud. Fuente: elaboración propia

Autor	elemento	Cantidad	observación
(Heaven y Finkelstein, 2004)	objetivo	4	No agrupa los objetivos de acuerdo con la metodología KAOS.
	Entidad	0	No logra identificar entidad.
	Agente	0	No logra identificar agentes.
	Operación	0	No logra identificar operaciones

Tabla 6.13. Evaluación comparativa completitud. Fuente: elaboración propia

Autor	elemento	Cantidad	observación
(Zapata et al., 2007)	objetivo	10	No agrupa los objetivos de acuerdo con la metodología KAOS.
	Entidad	0	No logra identificar entidad.
	Agente	3	
	Operación	0	No logra identificar operaciones

Tabla 6.14. Evaluación comparativa completitud. Fuente: elaboración propia

Autor	elemento	Cantidad	observación
(Lezcano, 2014). Resultados de esta tesis.	objetivo	16	Logra agrupar algunos objetivos de acuerdo con la clasificación (Lograr, Mantener, Evitar, Parar) que se presenta en la metodología KAOS.
	Entidad	10	Logra identificar entidad y atributos.
	Agente	6	Los clasifica en agente de ambiente o de software de acuerdo con (Letier, 2001).
	Operación	5	

En resumen, con relación a la completitud a través del modelo propuesto en esta tesis doctoral, se puede indicar que se logró identificar un mayor número de actores, entidades, operaciones, y objetivos en relación con los presentados en (Heaven y Finkelstein, 2004) y (Zapata et al., 2007).

Correctitud.

La sintaxis utilizada en (Heaven y Finkelstein, 2004), no es consistente con la metodología de KAOS dado que sólo identifican objetivos y (Zapata et al., 2007) sólo identifican objetivos y agentes. Los elementos de la metodología KAOS obtenidos en esta tesis, logran minimizar las inconsistencias de correctitud con relación al uso adecuado de la sintaxis, puesto que, logra identificar objetivos,

entidades, agentes y operaciones, elementos básicos que permiten la elaboración y jerarquización del diagrama de objetivos de KAOS.

Consistencia.

En (Heaven y Finkelstein, 2004), se presentan inconsistencia, puesto que, no se logran identificar los elementos (actores, entidades y operaciones) expresados en el universo del discurso y (Zapata et al., 2007) no identifican entidades y operaciones. En el modelo propuesto en esta tesis, los elementos básicos (objetivo, entidad, actor, y operación) identificados corresponden al universo del discurso, es decir, a la especificación textual de los requisitos.

Sistema de la pizzería

Compleitud: En las tablas 6.15 y 6.17, se puede observar la evaluación comparativa entre los autores sobre completitud.

Tabla 6.15. Evaluación comparativa completitud. Fuente: elaboración propia

Elementos caracterizados			
Autor	elemento	Cantidad	observación
(Zapata et al., 2006)	Objetivo	16	No agrupa los objetivos de acuerdo con la metodología KAOS.
	Entidad	0	No logra identificar entidad.
	Agente	0	No logra identificar agentes.
	Operación	0	No logra identificar Operaciones.

Tabla 6.16. Evaluación comparativa completitud. Fuente: elaboración propia

Elementos caracterizados			
Autor	elemento	Cantidad	observación
(Zapata et al., 2007)	Objetivo	8	No agrupa los objetivos de acuerdo con la metodología KAOS.
	Entidad	0	No logra identificar entidad.
	Agente	3	No logra identificar agentes.
	Operación	0	No logra identificar Operaciones.

Tabla 6.17. Evaluación comparativa completitud. Fuente: elaboración propia

Autor	elemento	Cantidad	observación
(Lezcano, 2014). Resultados de esta tesis	objetivo	16	Logra agrupar algunos objetivos de acuerdo con la clasificación (Lograr, Mantener, Evitar, Parar) que se presenta en la metodología KAOS.
	Entidad	6	Logra identificar entidad y atributos.
	Agente	5	Los clasifica en agente de ambiente o de software de acuerdo con (Letier, 2001).
	Operación	4	

En resumen, con relación a la completitud a través del modelo propuesto en esta tesis doctoral, se puede indicar que se logró identificar un mayor número de actores, entidades, operaciones y la misma cantidad de objetivos en relación con los presentados en (Zapata et al., 2006). Asimismo, en (Zapata et al., 2007) se logró identificar un menor número de objetivos y agentes, además, no se identifican entidades y operaciones.

Correctitud

La sintaxis utilizada en (Zapata et al., 2006), no es consistente con la metodología de KAOS dado que sólo identifica objetivos y (Zapata et al., 2007) no identifican entidades y operaciones. Los elementos de la metodología KAOS obtenidos en esta tesis, logran minimizar las inconsistencias de correctitud con relación al uso adecuado de la sintaxis, puesto que, logra identificar objetivos, entidades, agentes y operaciones, elementos básicos que permiten la elaboración y jerarquización del diagrama de objetivos de KAOS.

Consistencia.

En (Zapata et al., 2006), se presentan inconsistencia, puesto que, no se logran identificar los elementos (actores, entidades y operaciones) expresados en el universo del discurso y (Zapata et al., 2007) no identifican entidades y operaciones. En el modelo propuesto en esta tesis, los elementos básicos (objetivo, entidad, actor, y operación) identificados corresponden al universo del discurso, es decir, a la especificación textual de los requisitos.

6.3 Conclusiones del capítulo

En este capítulo se presentaron las pruebas de validación mediante la utilización de casos de estudio registrados en la literatura científica, éstos permitieron medir las características de exhaustividad, precisión, correctitud, completitud y consistencia.

Asimismo, se presentaron los textos de las especificaciones de los casos de estudio que sirvieron como insumo de entrada para la evaluación del modelo. Además, se mostraron los resultados y sus respectivos análisis, llevando a concluir que:

- (i) El sistema NL2KAOS, posee en promedio, una exhaustividad del 88,69% y una precisión del 77,75%.
- (ii) Los elementos de la metodología KAOS obtenidos a través del modelo propuesto en esta tesis, presentan mejores resultados que los registrados en la literatura y utilizados en el análisis realizado en este capítulo.
- (iii) Mejora la consistencia que debe existir entre el lenguaje natural y los elementos básicos que conforman el diagrama de objetivos de KAOS.
- (iv) El modelo desarrollado mejora las características de correctitud, completitud y consistencia durante el proceso de identificación y validación de requisitos mediante el uso de la metodología KAOS.

Capítulo 7

CONCLUSIONES

En este capítulo se registran las principales contribuciones obtenidas en esta tesis doctoral, asimismo, se enuncian los trabajos futuros que se derivan de ésta.

7.1 Contribuciones principales

Los aportes metodológicos desarrollados en esta tesis son: (i) caracterización de los elementos que conforman el diagrama de objetivos de KAOS, (ii) construcción de una Ontología para representar los elementos que conforman el diagrama de objetivos de KAOS, (iii) caracterización morfosintáctica y semántica de los elementos básicos del diagrama de objetivos de KAOS, (iv) arquitectura y diseño del modelo propuesto, y (v) validación del modelo.

La identificación y validación de requisitos de software a través de la elaboración del diagrama de objetivos de KAOS presenta los siguientes problemas: (i) la alta participación del analista durante el proceso, lo cual implica que no exista consistencia entre el lenguaje natural, los requisitos obtenidos y los elementos básicos que conforman el diagrama de objetivos de KAOS, (ii) no se obtienen de manera semiautomática los elementos básicos que conforman el diagrama de objetivos de KAOS, (iii) fallas en el manejo del procesamiento terminológico que evitan minimizar la ambigüedad semántica presente en el universo del discurso durante la educación de requisitos de software, y (iv) falta de validación semiautomática de requisitos de software por parte del interesado en la etapa inicial.

Con el propósito de presentar algunas soluciones a esta problemática, se propuso en esta tesis doctoral, crear un modelo de procesamiento terminológico para la educación y validación interactiva de requisitos de software a partir del diagrama de objetivos de KAOS, este modelo incluyó:

- (i) Realizar la caracterización de los principales elementos utilizados en el diagrama de objetivos de KAOS. Esta caracterización permitió identificar y seleccionar los siguientes elementos: Objetivo, Objetivo de evitar, Objetivo de mantenimiento, Objetivo de logro, Objetivo de parar, Entidad, Agente, Agente de Software, Agente de ambiente, y Operación, utilizados en esta tesis.
- (ii) Desarrollar una Ontología con base en el metamodelo del diagrama de objetivos de KAOS propuesto por (Matulevičius et al., 2007). La creación de esta Ontología, permitió en esta tesis una mejor definición y caracterización de los elementos que conforman el diagrama de objetivos de KAOS. La Ontología se desarrolló en OWL.
- (iii) Definir cuatro formas morfosintácticas para caracterizar objetivos, una para caracterizar agente, tres para caracterizar entidades y dos para caracterizar operaciones. La definición de las formas morfosintácticas, tienen como propósito principal en esta tesis, brindar una solución a la necesidad de identificar la consistencia que debe existir entre el lenguaje natural y los elementos básicos (objetivo, entidad, agente y operación) del diagrama de objetivos de KAOS.
- (iv) Analizar la ambigüedad semántica de tipo polisémica que influye de forma negativa durante el proceso de identificación y validación de requisitos de software a través de la elaboración del diagrama de objetivos de KAOS. Asimismo, se analizaron los algoritmos y las

técnicas que permiten realizar diferentes formas de desambiguación. Este análisis permitió elegir las técnicas MFS, Lesk y UKB para el modelo propuesto en esta tesis. La justificación de dicha elección obedece a que estas técnicas son utilizadas con mayor frecuencia a nivel mundial en el tópico de la desambiguación semántica, además, porque pueden ser personalizadas y aplicadas al campo de la Ingeniería de Software.

- (v) Definir cuatro reglas semánticas. Una vez definidas las formas morfosintácticas, se procedió con la definición de las reglas semánticas, las cuales permiten mejorar procesamiento terminológico propuesto en esta tesis, puesto que permiten minimizar la ambigüedad semántica.
- (vi) Diseñar y construir el sistema (software) NL2KAOS. Con el propósito de implementar y validar el modelo propuesto, se diseñó la arquitectura correspondiente al modelo de procesamiento terminológico para la educación y validación interactiva de requisitos de software a partir del diagrama de objetivos de KAOS. La arquitectura incluyó el uso de: (1) Freeling 3.1: herramienta de procesamiento de lenguaje natural que permite la identificación de roles temáticos dentro de una oración, permitiendo identificar sujeto, verbo y objeto, además permite el etiquetado morfosintáctico del árbol de dependencias parcial, (2) Multilingual central repository: base de datos léxica en idioma español, de relaciones semánticas entre términos, tales como hiperónimos, hipónimos y sinónimos, (3) las técnicas CALPS y javasimlib utilizadas para evaluar la similitud semántica, (4) tres alternativas de desambiguación (Lesk, MFS y UKB), (5) formas morfosintácticas utilizadas para brindar una solución a la necesidad de identificar la consistencia que debe existir entre el lenguaje natural y los elementos básicos del diagrama de objetivos de KAOS, (6) reglas semánticas para

minimizar la ambigüedad semántica. Asimismo, se diseñaron los casos de uso, el diagrama de clases y el diagrama de secuencias.

- (vii) Las reglas que permiten identificar los elementos básicos del diagrama de objetivos de KAOS y la obtención semiautomática de éste con la participación interactiva del interesado (usuario).
- (viii) La validación del modelo se realizó mediante casos de prueba registrados en la literatura científica, además, se tuvieron en cuenta las características exhaustividad, precisión (Baeza y Berthier, 1999), completitud, correctitud y consistencia (Zowghi y Gervasi, 2002). Esta validación permitió concluir que: (1) el modelo propuesto y validado a través del sistema (software) NL2KAOS desarrollado en el marco de esta tesis, posee en promedio, una exhaustividad del 88,69% y una precisión del 77,75%, (2) los elementos de la metodología KAOS obtenidos a través del modelo propuesto, presenta mejores resultados que los registrados en la literatura, (3) el modelo mejora la consistencia que debe existir entre el lenguaje natural y los elementos básicos que conforman el diagrama de objetivos de KAOS, (4) el modelo propuesto minimiza la ambigüedad semántica de tipo polisémica presente en el universo del discurso, (5) el modelo incluye la validación semiautomática de requisitos de software por parte del interesado en la etapa inicial del ciclo de vida del software, y (7) el modelo propuesto contempla las características de exhaustividad, precisión, correctitud, completitud y consistencia durante el proceso de identificación y validación de requisitos mediante el uso de la metodología KAOS.

7.2 Trabajos futuros

La Ingeniería de Software constituye un campo amplio de investigación. En consecuencia, se pueden formular nuevos proyectos de investigación que puedan dar continuidad a los trabajos realizados a la fecha. Algunos de los tópicos que pueden proporcionar continuidad al trabajo realizado en esta tesis doctoral se describen a continuación:

- (i) Identificar los demás elementos que conforman el diagrama de objetivos de KAOS (conflicto, relación AND, propiedad del dominio, entre otros).
- (ii) Definir otras formas de representación morfosintácticas para los elementos básicos del diagrama de objetivos de KAOS.
- (iii) Definir nuevas reglas que logren generar variaciones en los elementos del diagrama de objetivos de KAOS identificados en el marco de esta tesis.
- (iv) Caracterizar e incluir nuevos verbos que puedan presentar sinonimia con alguno de los cuatro tipos de verbos de objetivos (Lograr, Mantener, Para, Evitar) definidos en la metodología KAOS y presentados en esta tesis.
- (v) Resolver el tema de la anáfora (no contemplado en el alcance de esta tesis), puesto que, normalmente, en el lenguaje natural no siempre se repite el sujeto de la acción. En los casos de estudio presentados en esta tesis, se repite el agente, ejemplo: “El personal de ambulancia recibe instrucciones. El personal de ambulancia envía información”. Una forma más natural de expresarlo sería “El personal de ambulancia recibe instrucciones, además, éste envía información”.

- (vi) Realizar de manera automática la conversión de algunos diagramas UML (p.e, Casos de uso) desde el diagrama de objetivos de KAOS. Esta conversión podría garantizar la consistencia entre los requisitos expresados por los interesados y el diseño de la pieza de software representado a través de artefactos UML.
- (vii) Realizar un análisis exhaustivo para determinar una posible articulación del modelo obtenido en esta tesis con herramientas CASE para lograr la obtención de código que permita generar los prototipos correspondientes a la pieza de software a desarrollar.
- (viii) Realizar estudios detallados que permitan generar un análisis a las herramientas utilizadas en el Procesamiento de Lenguaje Natural, para mejorar el tema articulado con la teoría lingüística (morfología, sintaxis, semántica) asociada al idioma español, la cual podría mejorar el procesamiento automático del texto (universo del discurso) durante la educación y validación de requisitos de software.
- (ix) Efectuar nuevos experimentos que logren demostrar las ventajas del modelo desarrollado en esta tesis.

Anexo A. Manual de usuario

1. Introducción

NL2KAOS (Procesamiento del Lenguaje Natural a Modelo de KAOS), es un software que permite la clasificación de diversos componentes del modelo de objetivos de KAOS propuesto por (Lamsweerde, 2000). Puntualmente NL2KAOS permite el procesamiento de un discurso escrito en español y a partir de este obtener los componentes del diagrama de objetivos de KAOS, que permite exportarlos a formatos XMI (válido para Objectiver), la ontología KAOS en formato OWL, o en un archivo con extensión TXT.

NL2KAOS es un software construido bajo el lenguaje de programación de JAVA, JSP y PHP que utiliza componentes de dos herramientas para el Procesamiento del Lenguaje Natural, por ejemplo, el Freeling 3.1 que permite el etiquetado morfosintáctico de documentos y textos en distintos idiomas.

A continuación se enuncian los pasos que debe seguir el usuario para la utilización de NL2KAOS:

Paso 1: Ingresar información en lenguaje natural escrito.

Ingresar el Universo del Discurso (U del D), el cual debe contemplar las reglas morfosintácticas definidas en esta tesis. Una vez ingresado el U del D, el usuario debe seleccionar el tipo de algoritmo que desea utilizar para realizar la desambiguación semántica y el paquete que utilizará para medir la similitud semántica. En la figura A1 se puede observar con mayor detalle este paso.



Figura A.1. Ingresar U del D y seleccionar algoritmo. Fuente: Elaboración propia.

Paso 2: Depurar elementos identificados:

Verificar los elementos identificados, dado el caso que requiere adicionar un nuevo elemento deberá regresar al paso anterior. Ver figuras de la A2-A4.

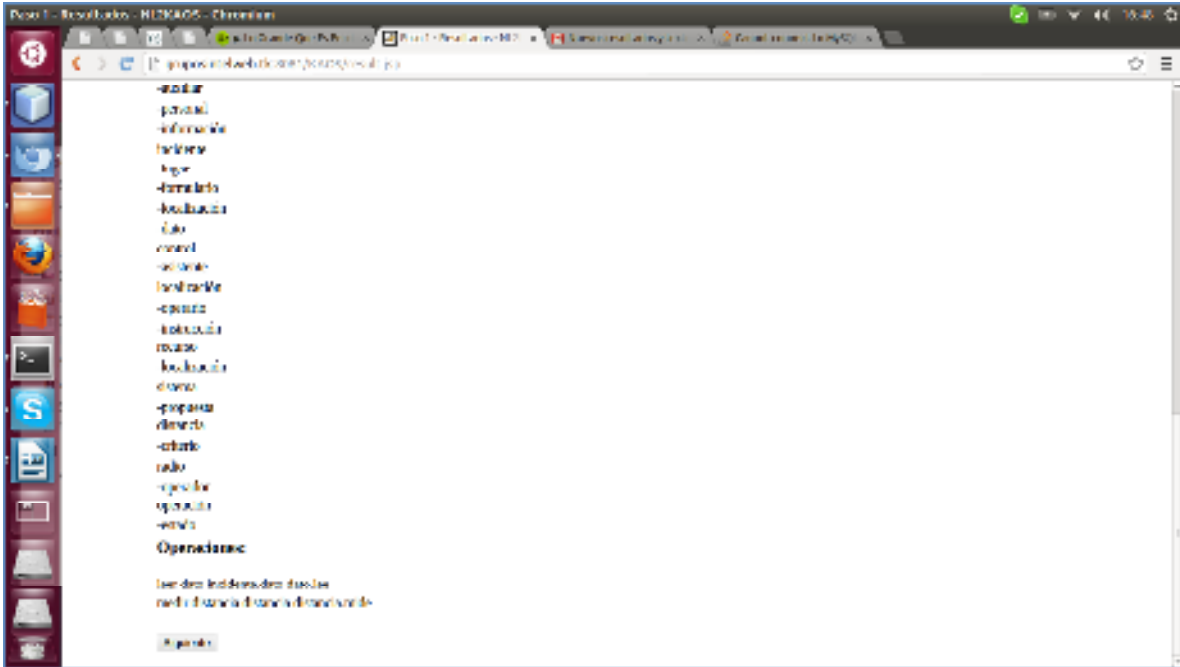


Figura A.4. Depurar elementos identificados. Fuente elaboración propia

Paso 3: Remover elementos.

El usuario selecciona y remueve los elementos que considera no pertinentes. Luego presiona el botón “Enviar”. El sistema almacena esta información en la base de datos. En las figuras de la A5-A7 se puede observar con mayor detalle este paso.

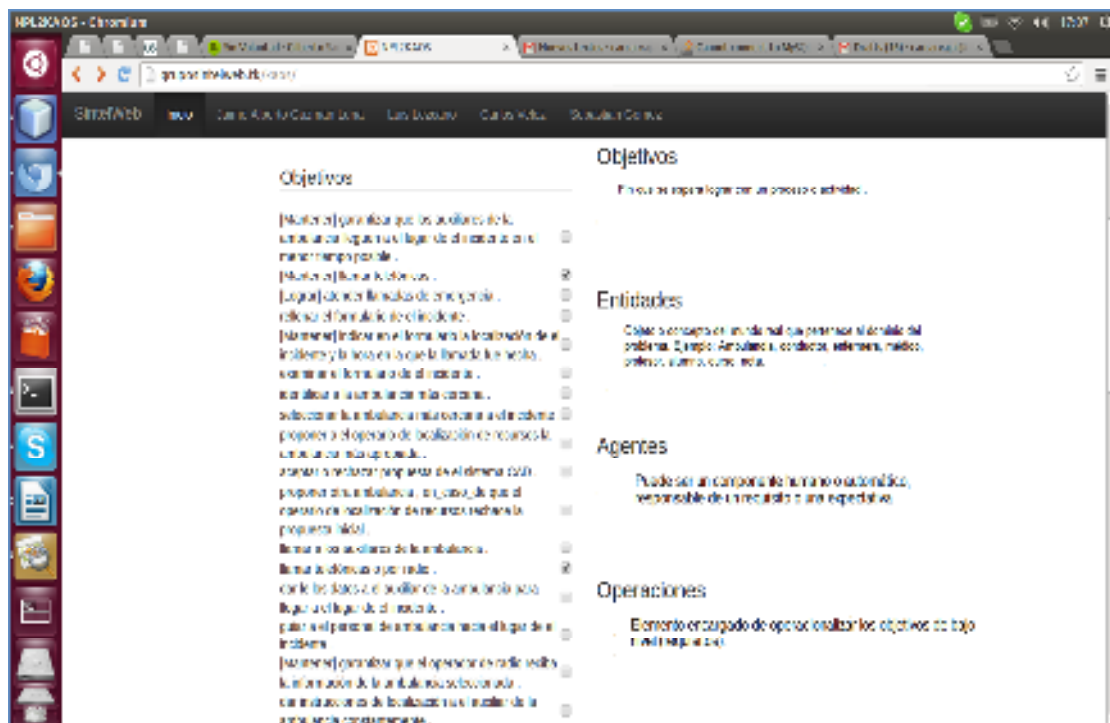


Figura A.5. Seleccionar los elementos básicos. Fuente elaboración propia

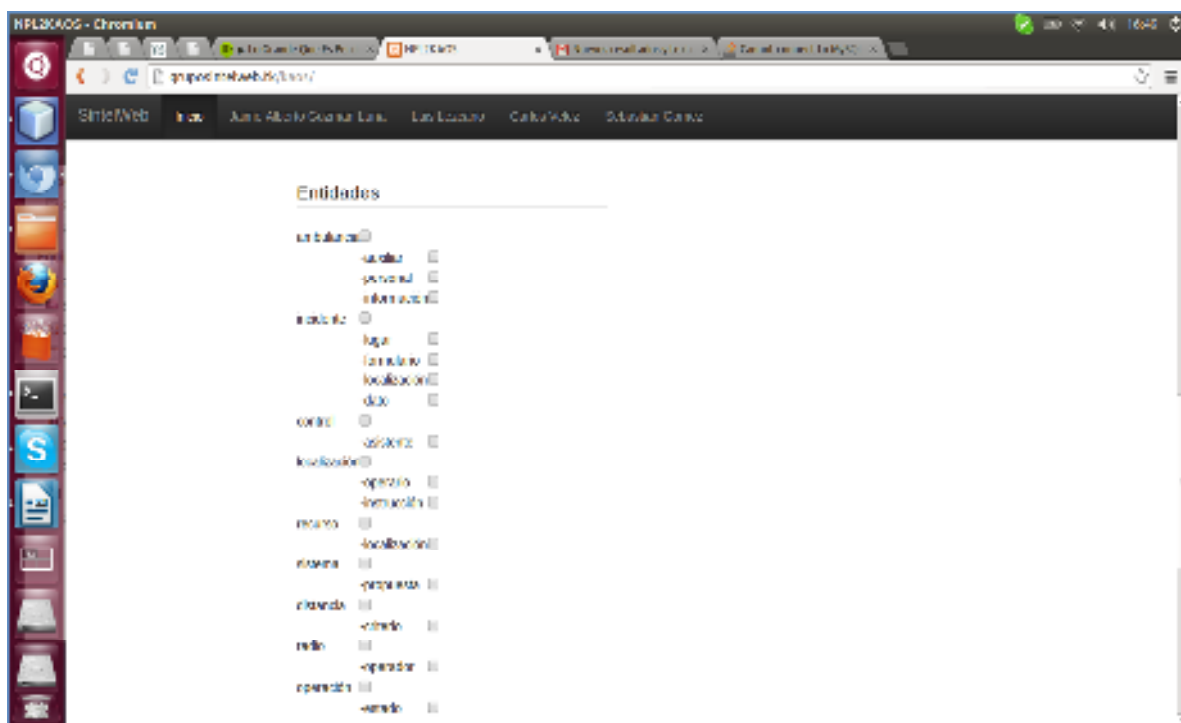


Figura A.6. Seleccionar los elementos básicos. Fuente elaboración propia

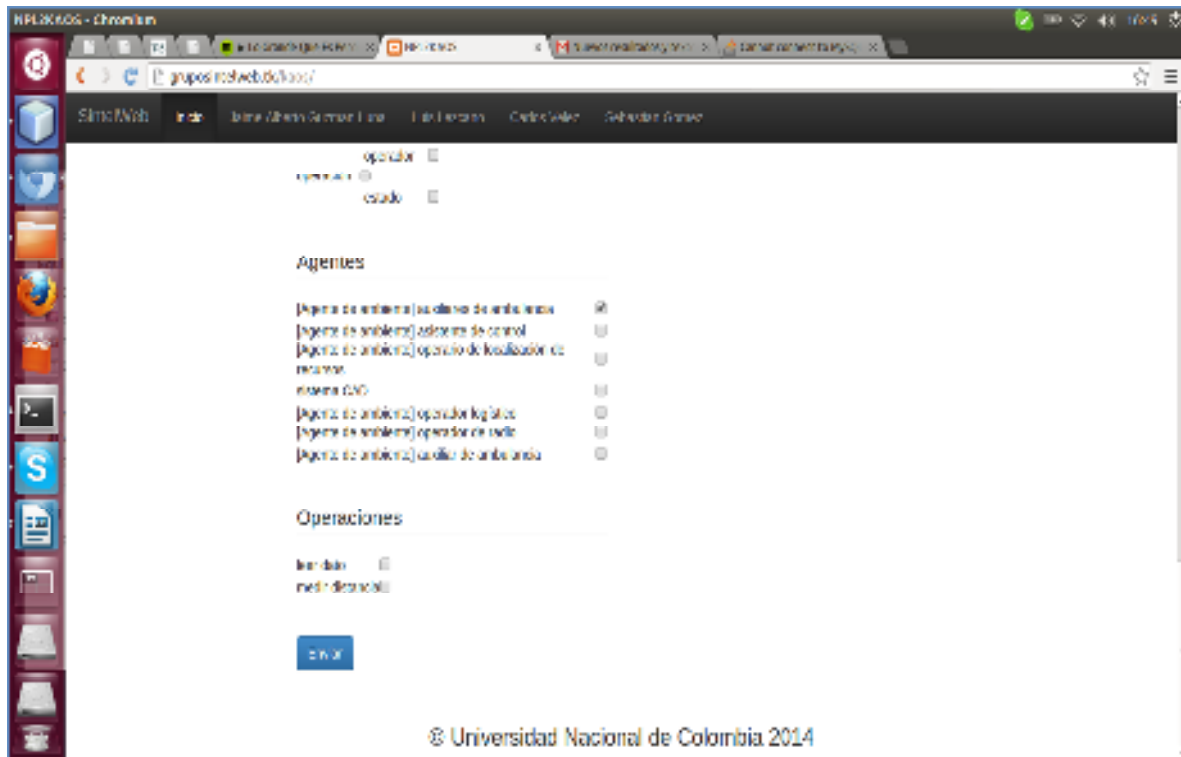


Figura A.7. Seleccionar los elementos básicos. Fuente elaboración propia

Paso 4: Seleccionar objetivo principal.

El usuario debe seleccionar el objetivo general y luego dar clic en el botón “Enviar”. El sistema (software) almacena esta información en la base de datos. Ver figura A8.





Figura A.9. Seleccionar el padre para cada objetivo. Fuente elaboración propia

Paso 6: Asignar agentes.

El usuario debe asignar los agentes responsables de cumplir los objetivos (operacionalizables). El usuario presiona el botón “Enviar”. El sistema almacena las responsabilidades en la base de datos. Ver figura A10.

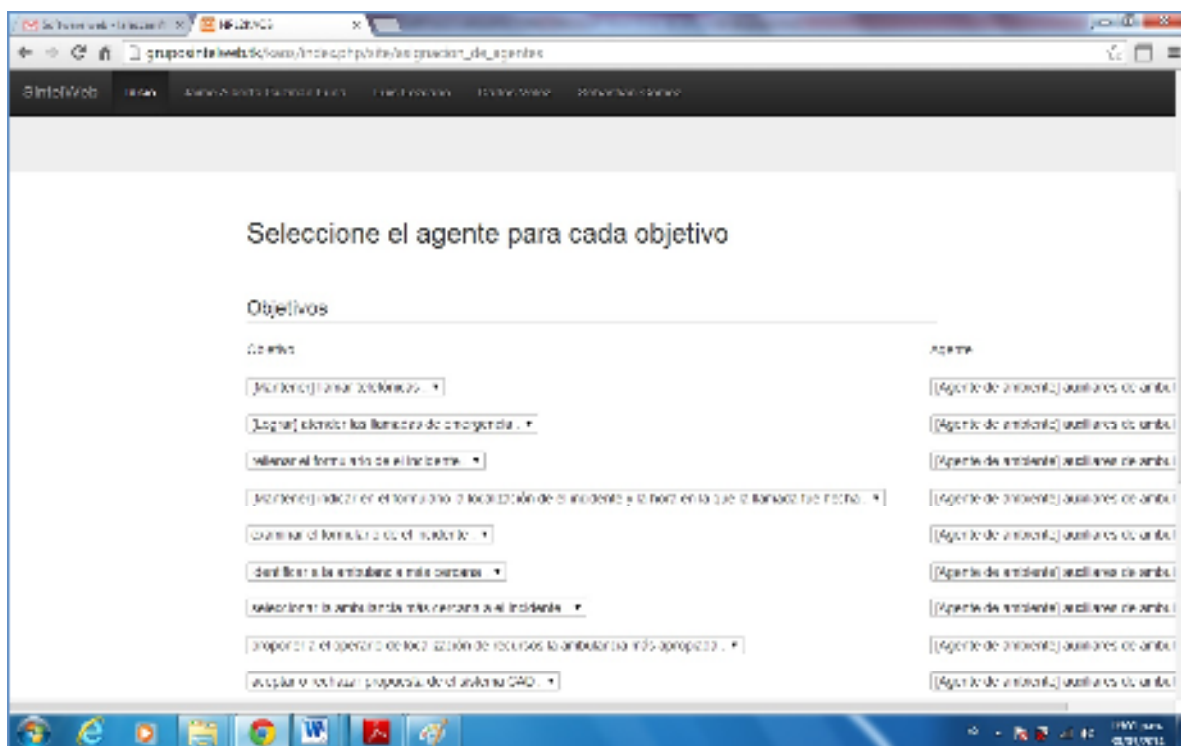


Figura A.10. Asignar agentes. Fuente elaboración propia

Paso 7: Establecer operaciones.

El usuario indica la operación que corresponde a cada objetivo operacionalizable. Dado el caso en que un objetivo posea más de una operación, el usuario da clic en el botón “+ Agregar Campo”, seleccionando la operación y el objetivo respectivo. Cuando finalice, el usuario da clic en el botón “Enviar” y el sistema almacena la información procesada en la base de datos. Ver figura A11.

Seleccione la operación para cada objetivo

Objetivos	Operación
[Seleccionar] [Insertar] [Eliminar]...	[modo de acceso]...
[Logueo] [Identificación] [Búsqueda de información]...	[modo de acceso]...
[Seleccionar] [Insertar] [Eliminar]...	[modo de acceso]...
[Verificación] [Identificación] [Información] [Información de incidentes] [Información de la causa o impacto]...	[modo de acceso]...
[Consultar] [Eliminar] [Insertar]...	[modo de acceso]...
[Añadir] [Eliminar] [Insertar]...	[modo de acceso]...
[Seleccionar] [Insertar] [Eliminar]...	[modo de acceso]...
[Proporcionar] [Actualizar] [Eliminar]...	[modo de acceso]...

Figura A.11. Asignar operación. Fuente elaboración propia

Paso 8: Exportar diagrama.

El usuario debe indicar si desea: (i) exportar el diagrama de objetivos de KAOS al Objectiver, (ii) Realizar instancias de la ontología KAOS, en OWL, o (iii) exportar un archivo de texto con los resultados, incluyendo el sentido de cada palabra que se obtuvo en la desambiguación. Ver figura A12.

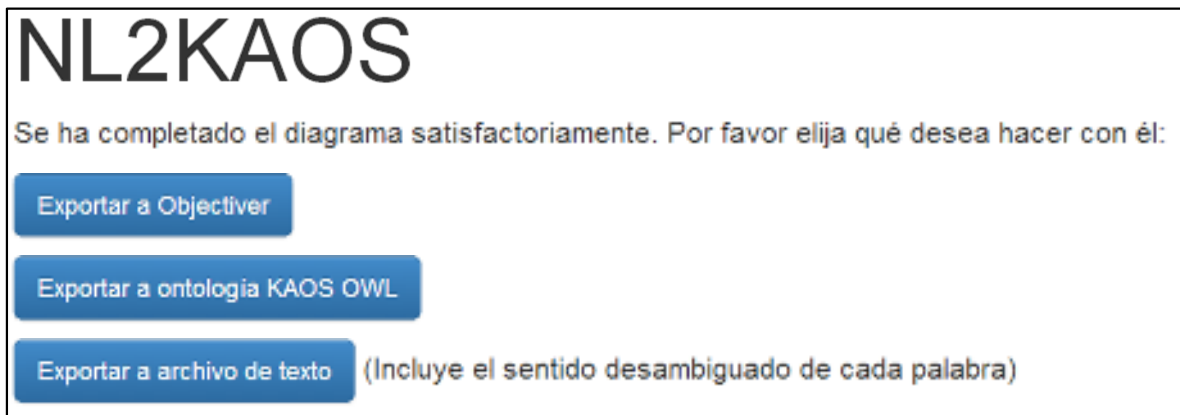


Figura A.12. Exportar diagrama. Fuente elaboración propia

Anexo B. Casos de estudio adaptados y utilizados en la experimentación

Caso de estudio 1: servicio de ambulancia de Londres

Es imprescindible garantizar que los auxiliares de la ambulancia lleguen al lugar del incidente en el menor tiempo posible.

El asistente de control recibe llamadas telefónicas. El asistente de control debe atender llamadas de emergencia. El asistente de control tiene que rellenar el formulario del incidente. Es necesario que el asistente de control indique en el formulario la localización del incidente y la hora en la que la llamada fue hecha. El asistente de control ingresa los datos en el sistema.

Por su parte, el operario de localización de recursos debe examinar el formulario del incidente. El operario de localización de recursos lee los datos del incidente. Seguidamente, el operario de localización de recursos tiene que identificar a la ambulancia más cercana.

El operario de localización de recursos debe seleccionar la ambulancia más cercana al incidente. Para ello, el sistema CAD debe proponer al operario de localización de recursos la ambulancia más apropiada. El operario de localización de recursos puede aceptar o rechazar propuesta del sistema CAD. El sistema CAD mide la distancia entre la ambulancia y el lugar del incidente. El sistema CAD selecciona una ambulancia de acuerdo con algún criterio de distancia.

El sistema CAD puede proponer otra ambulancia, en caso de que el operario de localización de recursos rechace la propuesta inicial. Es necesario que el operador

logístico llame a los auxiliares de la ambulancia. Por lo tanto, el operador logístico realiza llamadas telefónicas o por radio. El operador logístico debe darle los datos al auxiliar de la ambulancia para llegar al lugar del incidente. Así, el operador logístico comunica los datos.

Es menester que el operador de radio guíe al personal de ambulancia hacia el lugar del incidente. Se debe garantizar que el operador de radio reciba la información de la ambulancia seleccionada. El operador de radio debe dar instrucciones de localización al auxiliar de la ambulancia constantemente. El auxiliar de la ambulancia tiene que reportar el estado de la operación al operador de radio.

Caso de estudio 2: sistema de un ascensor

La compañía encargada del ascensor debe garantizar el funcionamiento correcto del ascensor. Además, la compañía tiene que asegurar controles de emergencia. La compañía tiene que realizar el montaje del ascensor. Es necesario que el ingeniero civil diseñe la estructura del ascensor.

El ingeniero civil debe garantizar una forma de escape del ascensor. El sistema de control debe evitar que las puertas se abran mientras esté en movimiento. Es importante que el sistema de control detenga el elevador siempre que haya una falla de energía. El sistema de control debe encender la luz de emergencia cuando sea necesario. El sistema de control puede abrir las puertas, cuando esté en el nivel indicado. El sistema de control cierra las puertas cuando esté en movimiento. Se desea que el sistema de control informe a los pasajeros del estado de su petición. Es menester que el sistema de control reporte a los pasajeros las condiciones de sobrepeso. El sistema de control puede dar alertas.

El ingeniero civil construye el ascensor. El ingeniero civil construye el escape.

El ingeniero de control tiene que verificar el funcionamiento del botón de emergencia. La compañía monta una planta para la energía de emergencia. El ingeniero de control realiza pruebas. La compañía tiene que realizar una interfaz usable por el usuario. Para ello, el ingeniero de control debe crear una interfaz basada en botones.

El técnico de sistemas tiene que desarrollar el software para el funcionamiento de los ascensores. El técnico de sistemas programa el software. El técnico de sistemas verifica la calidad del programa.

Caso de estudio 3: sistema pizzería

El administrador de la pizzería desea aumentar la rentabilidad del negocio. La pizzería tiene un administrador, un operador logístico, un cocinero y dos repartidores. El administrador de la pizzería debe ofrecer a los clientes diversos tipos de pizza. El administrador de la pizzería debe ofrecer a los clientes diversos tamaños de pizza. También, el administrador de la pizzería quiere ofrecer a los clientes la posibilidad de ordenar aditivos para la pizza.

Los clientes pueden ordenar pizza en el local o a domicilio. Es necesario que el administrador de la pizzería defina una zona de cobertura determinada.

El administrador de la pizzería realiza promociones constantemente. Una promoción es: la pizzería debe entregar el pedido al cliente en menos de 30 minutos; de lo contrario, será gratis. Así, los repartidores deben evitar entregar pedidos retardados.

Es necesario que los empleados preparen la pizza ordenada por el cliente. Para ello, el operador logístico recibe las llamadas. El operador logístico debe ingresar el pedido en el sistema. El operador logístico debe enviar la orden al cocinero. El cocinero tiene que preparar la masa, los ingredientes y el horno. El cocinero debe

meter la pizza al horno. Una vez lista, el cocinero debe despachar la orden. El operador logístico envía a los repartidores. Los repartidores entregan la pizza.

Los repartidores deben entregar la pizza en menos de 30 minutos, de lo contrario, el administrador de la pizzería cobra a los repartidores el pedido. Así, el administrador de la pizzería busca evitar la pérdida de dinero.

Caso de estudio 4: Guayacán

Agencia de viajes Guayacán es una reconocida empresa guatemalteca cuya principal función es ofrecer a las personas distintas guías turísticas dentro del país. Desde su fundación en 1992 el equipo de Guayacán ha recorrido los caminos de Guatemala, mostrando la intensidad de sus paisajes y riqueza de su cultura. Ahora Guayacán busca incrementar su economía.

Para llevar su objetivo a pleno, la empresa debe ofrecer un servicio de buena calidad a los turistas y a sus principales clientes. Para esto, debe tener empleados especializados en los temas viajeros. El contratista de la empresa debe contratar asesores especializados constantemente. Adicionalmente, para esta capacitación, el director de servicio desea realizar un examen de valoración para los empleados y así poder saber quiénes están cumpliendo con lo requerido y quiénes no lo están haciendo. El director de servicio reporta los resultados en el sistema. El presidente lee los resultados. El presidente estudia los resultados. El sistema almacena los datos ingresados por el director de servicio. El presidente de la empresa pretende adquirir tecnología de primera calidad.

Otro aspecto importante para permitir el desarrollo económico de la empresa es organizar viajes a otros lugares turísticos nunca antes visitados por Guayacán. En esto es fundamental el coordinador de la empresa. El coordinador de la empresa debe gestionar los proyectos adecuadamente. Así, el coordinador de la empresa

tiene que realizar contratos con diferentes empresas. El coordinador envía resultados al presidente. El presidente lee los resultados. Además, el coordinador tiene que llevar un manejo de clientes de todo el país para hacer de estos nuevos viajes algo seguro.

Caso de estudio 5: Marinozzi

Alimentos Marinozzi es una reconocida empresa Rosarina (Argentina), fundada en el año de 1950 por el señor Don Vicente Marinozzi y su esposa Doña Alba Pellizari. La principal fabricación de Alimentos Marinozzi son los productos del trigo. Además, siempre ha tenido una visión futurista con un buen concepto respecto a su actualidad. Por eso, su principal objetivo es aumentar sus utilidades.

Para poder cumplir este objetivo hay algunos aspectos primordiales. El gerente desea crecer en el mercado de la producción. Para esto, el director debe ofrecer en el mercado alimentos variados en el sector. También es fundamental que la empresa busque la manera de ampliar sus instalaciones. El presidente gestiona este proceso. El gerente reporta los resultados del proceso. Por otra parte, don Vicente debe cumplir con las demandas hechas por los clientes. Don Vicente mide las ventas realizadas. Para lograr este objetivo el contratista debe mirar la forma de buscar personal para laborar las 24 horas del día.

Caso de estudio 6: Carrefour

Carrefour es una distinguida cadena francesa internacional de hipermercados. Su actividad se centra en tres mercados: Europa, Asia y América Latina. El principal objetivo de esta gran empresa es ser líder de los supermercados de cadena.

Debido a la gran competencia internacional debe buscar alternativas para mejorar. Esto conlleva a que Carrefour venda productos de excelente calidad. Así, el jefe de producción debe buscar excelentes proveedores. Además, para cumplir con la calidad, debe existir en la empresa un control de calidad. El jefe de producción tiene que vigilar el control de calidad. El jefe de producción reporta los controles de calidad. Otro aspecto importante para buscar el liderazgo es brindar una buena atención al cliente. Para ello, Carrefour necesita contar con personal de excelentes competencias. Adicionalmente, el jefe de personal debe buscar las mejores alternativas de tecnología. Por su parte, al presidente le interesa mejorar la atención al usuario. El presidente construye estrategias de marketing.

Caso de estudio 7: Colácteos

Colácteos es una empresa no muy reconocida a nivel nacional. Su principal producción son los derivados de la leche. Por lo que su primordial objetivo es expandir su mercado a nivel nacional. Para ello, Colácteos considera ideal buscar más inversionistas a nivel local y nacional. Estos inversionistas podrían aparecer a medida que se logre un reconocimiento de la empresa por los productos y su buena calidad. Para esto, el coordinador debe certificar que los productos sean de buena calidad. El presidente realiza operaciones financieras que aumenten los ingresos. Los inversionistas reportan sus intereses para con la empresa.

Es indispensable que el presidente busque la manera de construir sucursales en las principales ciudades del país. Otro objetivo muy importante y que ayudaría a la expansión sería el de buscar innovar en el mercado con productos nuevos. Es entonces necesario que el área de marketing proponga nuevas estrategias. El jefe de producción debe innovar productos dependiendo de las necesidades del consumidor.

Caso de estudio 8: Colcerámicas

Colcerámicas es la empresa dominante en el mercado colombiano en todo lo relacionado con la cerámica. Sin embargo, hoy en día está al acecho de pequeñas empresas de cerámicas en crecimiento. Por este motivo, tiene como objetivo principal mantener el liderazgo en la producción de cerámicas del país.

Colcerámicas debe conseguir materiales de primera calidad. Por lo tanto, el presidente debe mantener una excelente comunicación con los proveedores. Otro aspecto importante es emplear maquinaria de primera calidad y de muy buen desempeño. Sin embargo, el mercado está en constante cambio, por lo que el presidente debe adquirir maquinaria eficiente constantemente también. El ingeniero de procesos mide el desempeño de las máquinas. El ingeniero de control reporta los resultados de la evaluación del ingeniero de procesos.

El gerente desea mantener el mercado a su favor. El gerente decide contratar personal de muy buen desempeño. Este grupo de trabajadores lo conforman las personas comprometidas con el progreso de la empresa, tales como técnicos muy bien capacitados. El jefe de personal debe contratar a los nuevos trabajadores. El jefe de personal tiene que promover pequeños incentivos para mejorar el desempeño empresarial.

Caso de estudio 9: ENCA

La Escuela Nacional Central de Agricultura ENCA se ubica en la finca Bárcena, del municipio de Villa Nueva, departamento de Guatemala. Su clima es templado; el área de la finca está comprendida en la zona denominada bosque húmedo subtropical templado, con una altitud de 1,400 metros sobre el nivel del mar. Se encuentra situada a 3 kilómetros de la cabecera municipal de Villa Nueva y a 17.5

kilómetros de la ciudad capital. Su acceso es a través de una carretera asfaltada que se comunica con la autopista CA-4, carretera al pacífico.

Fue fundada en 1921 con el nombre de Escuela Nacional de Agricultura ENA, en la finca nacional La aurora, de la ciudad capital. En 1944 se trasladó a la finca Bárcena que es donde se encuentra actualmente. El rector pretende integrar el sistema de educación y capacitación agrícola y ambiental. El director general debe integrar a la universidad desde escuelas de formación agrícola, escuelas de educación agrícola media, nivel universitario y postgrado orientado hacia el desarrollo sostenible, agronegocios, manejo de recursos naturales, y desarrollo rural integral. Para ello, el director general construye proyectos productivos estudiantiles, talleres, giras de estudio, prácticas supervisadas, investigaciones de campo y bibliográficas. Además, los docentes tienen que garantizar la realización de investigaciones de campo. Por su parte, el asistente de dirección debe garantizar que los estudiantes tengan giras de estudio y prácticas supervisadas. Los docentes deben medir el desempeño de los estudiantes.

Caso de estudio 10: YAMAHA

YAMAHA quiere tener una excelente gestión de sus recursos. YAMAHA desea brindar una administración de calidad.

El gerente regional de YAMAHA debe garantizar que la empresa cumpla con sus responsabilidades empresariales, ambientales y sociales. Para ello, el jefe de recursos humanos tiene que garantizar el bienestar de los empleados. Para tal fin, el jefe mide la satisfacción de los empleados. Además, el director de ventas busca garantizar la satisfacción de las necesidades de los clientes. Para esto, el director de ventas mide la satisfacción de los clientes. Por otra parte, el jefe de personal pretende garantizar el bienestar de los empleados y sus familias.

Detalle de los resultados obtenidos para cada caso de estudio

Caso de estudio Ascensor

Tabla B.1. Resultados de los objetivos identificados por el modelo para el caso de estudio del ascensor. Fuente: elaboración propia.

Objetivos reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Garantizar el funcionamiento correcto del ascensor.	1. [Mantener] Garantizar el funcionamiento correcto del ascensor.	1. [Mantener] Garantizar el funcionamiento correcto del ascensor.	1. [Mantener] Garantizar el funcionamiento correcto del ascensor.	1. [Mantener] Garantizar el funcionamiento correcto del ascensor.
Asegurar controles de emergencia.	2. [Mantener] Asegurar controles de emergencia.	[Mantener] Asegurar controles de emergencia.	[Mantener] Asegurar controles de emergencia.	[Mantener] Asegurar controles de emergencia.
Realizar el montaje del ascensor.	3. [Mantener] Realizar el montaje del ascensor.	3. [Mantener] Realizar el montaje del ascensor.	3. [Mantener] Realizar el montaje del ascensor.	3. [Mantener] Realizar el montaje del ascensor.
Diseñar la estructura del ascensor.	4. [Mantener] Diseñar la estructura del ascensor.	4. [Mantener] Diseñar la estructura del ascensor.	4. [Mantener] Diseñar la estructura del ascensor.	4. [Mantener] Diseñar la estructura del ascensor.
Garantizar una forma de escape del ascensor.	5. [Mantener] Garantizar una forma de escape del ascensor.	[Mantener] Garantizar una forma de escape del ascensor.	[Mantener] Garantizar una forma de escape del ascensor.	[Mantener] Garantizar una forma de escape del ascensor.
Evitar que las puertas se abran mientras esté en movimiento.	6. [Mantener] Evitar que las puertas se abran mientras esté en movimiento.	[Mantener] Evitar que las puertas se abran mientras esté en movimiento.	[Mantener] Evitar que las puertas se abran mientras esté en movimiento.	[Mantener] Evitar que las puertas se abran mientras esté en movimiento.
Detener el elevador siempre que haya una falla	7. [Parar] Detener el elevador siempre que	[Parar] Detener el elevador siempre que	[Parar] Detener el elevador siempre que	[Parar] Detener el elevador siempre que haya una falla de

de energía.	haya una falla de energía.	haya una falla de energía.	haya una falla de energía.	energía.
Encender la luz de emergencia cuando sea necesario.	8. [Parar] Encender la luz de emergencia cuando sea necesario.	[Parar] Encender la luz de emergencia cuando sea necesario.	[Parar] Encender la luz de emergencia cuando sea necesario.	[Parar] Encender la luz de emergencia cuando sea necesario.
	9. [Mantener] Abrir las puertas, cuando esté en el nivel indicado.	9. [Mantener] Abrir las puertas, cuando esté en el nivel indicado.	9. [Mantener] Abrir las puertas, cuando esté en el nivel indicado.	9. [Mantener] Abrir las puertas, cuando esté en el nivel indicado.
Informar a los pasajeros del estado de su petición.	10. Desear que el sistema de control informe a los pasajeros del estado de su petición.	10. Desear que el sistema de control informe a los pasajeros del estado de su petición.	10. Desear que el sistema de control informe a los pasajeros del estado de su petición.	10. Desear que el sistema de control informe a los pasajeros del estado de su petición.
	11. [Mantener] Dar alertas.	[Mantener] Dar alertas.	[Mantener] Dar alertas.	[Mantener] Dar alertas.
Verificar el funcionamiento del botón de emergencia.	12. [Mantener] Verificar el funcionamiento del botón de emergencia.	[Mantener] Verificar el funcionamiento del botón de emergencia.	[Mantener] Verificar el funcionamiento del botón de emergencia.	[Mantener] Verificar el funcionamiento del botón de emergencia.
Realizar una interfaz usable por el usuario.	13. [Mantener] Realizar una interfaz usable por el usuario.	13. [Mantener] Realizar una interfaz usable por el usuario.	13. [Mantener] Realizar una interfaz usable por el usuario.	13. [Mantener] Realizar una interfaz usable por el usuario.
Crear una interfaz basada en botones.	14. [Lograr] Crear una interfaz basada en botones.	14. [Lograr] Crear una interfaz basada en botones.	14. [Lograr] Crear una interfaz basada en botones.	14. [Lograr] Crear una interfaz basada en botones.
Desarrollar el software para el funcionamiento de los ascensores.	15. [Lograr] Desarrollar el software para el funcionamiento de los ascensores.	15. [Lograr] Desarrollar el software para el funcionamiento de los ascensores	15. [Lograr] Desarrollar el software para el funcionamiento de los ascensores	15. [Lograr] Desarrollar el software para el funcionamiento de los ascensores
Reportar a los pasajeros las condiciones de				

sobrepeso.				
Hallados AND Presentes	13	13	13	13
#Elementos presentes	14	14	14	14
#Elementos hallados	15	15	15	15
Exhaustividad	92,86%	92,86%	92,86%	92,86%
Precisión	86,67%	86,67%	86,67%	86,67%

Tabla B.2. Resultados de los agentes identificados por el modelo para el caso de estudio del ascensor. Fuente: elaboración propia.

	Técnica de desambiguación utilizada			
Agentes reales	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. Compañía encargada de ascensor.	1. Compañía encargada de ascensor.	1. Compañía encargada de ascensor.	1. Compañía encargada de ascensor.
	2. Compañía.	2. Compañía.	2. Compañía.	2. Compañía.
Ingeniero civil.	3. [Agente de ambiente] Ingeniero civil.	3. [Agente de ambiente] Ingeniero civil.	3. [Agente de ambiente] Ingeniero civil.	3. [Agente de ambiente] Ingeniero civil.
Sistema de control.	4. [Agente de software] Sistema de control	4. [Agente de software] Sistema de control	4. [Agente de software] Sistema de control	4. [Agente de software] Sistema de control
	5. [Agente de software] Puertas.	5. [Agente de software] Puertas.	5. [Agente de software] Puertas.	5. [Agente de software] Puertas.
Ingeniero de control.	6. [Agente de ambiente] Ingeniero de control.	6. [Agente de ambiente] Ingeniero de control.	6. [Agente de ambiente] Ingeniero de control.	6. [Agente de ambiente] Ingeniero de control.
Técnico de sistemas.	7. [Agente de ambiente] Técnico de	7. [Agente de ambiente] Técnico de sistemas.	7. [Agente de ambiente] Técnico de	7. [Agente de ambiente] Técnico de sistemas.

	sistemas.		sistemas.	
Hallados AND Presentes	4	4	4	4
#Elementos presentes	4	4	4	4
#Elementos hallados	7	7	7	7
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	57,14%	57,14%	57,14%	57,14%

Tabla B.3. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio del ascensor. Fuente: elaboración propia.

Entidades reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. Emergencia -control -luz -botón -energía	1. Emergencia -control -luz -botón -energía	1. Emergencia -control -luz -botón -energía	1. Emergencia -control -luz -botón -energía
1. Planta -energía -falla	2. Ascensor -montaje -estructura -escape - funcionamiento	2. Ascensor -montaje -estructura -escape -funcionamiento	2. Ascensor -montaje -estructura -escape - funcionamiento	2. Ascensor -montaje -estructura -escape -funcionamiento
2. Ascensor -montaje -estructura -escape -controlador -estado -escape -forma	3. escape -forma	3. escape -forma	3. escape -forma	3. escape -forma
3. Control -sistema -ingeniero	4. Control -sistema -ingeniero	4. Control -sistema -ingeniero	4. Control -sistema -ingeniero	4. Control -sistema -ingeniero
4. Botón - funcionamiento	5. Energía -falla	5. Energía -falla	5. Energía -falla	5. Energía -falla

5. Luz	6. Estado -pasajero	6. Estado -pasajero	6. Estado -pasajero	6. Estado -pasajero
6. Técnico	7. Petición -estado	7. Petición -estado	7. Petición -estado	7. Petición -estado
7. Compañía	8. Sobrepeso -condición	8. Sobrepeso -condición	8. Sobrepeso -condición	8. Sobrepeso -condición
8. Pasajero	9. Botón - funcionamiento	9. Botón -funcionamiento	9. Botón - funcionamiento	9. Botón -funcionamiento
9. Botón - funcionamiento	10. Sistema -técnico	10. Sistema -técnico	10. Sistema -técnico	10. Sistema -técnico
10. Sistema	11. Programa -calidad	11. Programa -calidad	11. Programa -calidad	11. Programa -calidad
11. Programa -calidad				
Hallados AND Presentes	14	14	14	14
#Elementos presentes	25	25	25	25
#Elementos hallados	29	29	29	29
Exhaustividad	56,00%	56,00%	56,00%	56,00%
Precisión	48,28%	48,28%	48,28%	48,28%

Tabla B.4. Resultados de las operaciones identificadas por el modelo para el caso de estudio del ascensor. Fuente: elaboración propia.

Operaciones reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Construir escape.	1. Construir el escape.		1. Construir el escape.	1. Construir el escape.
Realizar pruebas.	2. Realizar pruebas		2. Realizar pruebas	
	3. Estar en movimiento.		3. Estar en movimiento.	

Montar planta.	4. Montar una planta.	1. Montar una planta	4. Montar una planta.	
Abrir puertas.				
Cerrar puertas.	5. Cerrar las puertas.		5. Cerrar las puertas.	2. Cerrar las puertas.
Construir ascensor.	6. Construir el ascensor.		6. construir el ascensor.	3. Construir el ascensor.
Programar software.				
Verificar calidad.	7. Verificar la calidad.		7. verificar la calidad	
Dar alerta.				
Hallados AND Presentes	6	1	6	3
#Elementos presentes	9	9	9	9
#Elementos hallados	7	1	7	3
Exhaustividad	66,67%	11,11%	66,67%	33,33%
Precisión	85,71%	100,00%	85,71%	100,00%

Caso de estudio pizzería

Tabla B.5. Resultados de los objetivos identificados por el modelo para el caso de estudio de la pizzería. Fuente: elaboración propia.

Objetivos reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Aumentar rentabilidad del negocio.	1. [Mantener] Aumentar la rentabilidad del negocio.	1. [Mantener] Aumentar la rentabilidad del negocio.	1. [Mantener] Aumentar la rentabilidad del negocio.	1. [Mantener] Aumentar la rentabilidad del negocio.
Ofrecer a los clientes diversos tipos de pizza.	2. [Mantener] Ofrecer a los clientes diversos tipos	2. [Mantener] Ofrecer a los clientes diversos tipos de pizza.	2. [Mantener] Ofrecer a los clientes	2. [Mantener] Ofrecer a los clientes diversos tipos de pizza.

	de pizza.		diversos tipos de pizza.	
Ofrecer a los clientes diversos tamaños de pizza.	3. [Mantener] Ofrecer a los clientes diversos tamaños de pizza.	3. [Mantener] Ofrecer a los clientes diversos tamaños de pizza.	3. [Mantener] Ofrecer a los clientes diversos tamaños de pizza.	3. [Mantener] Ofrecer a los clientes diversos tamaños de pizza.
Ofrecer a los clientes la posibilidad de ordenar aditivos para la pizza.	4. [Mantener] Ofrecer a los clientes la posibilidad de ordenar aditivos para la pizza.	4. [Mantener] Ofrecer a los clientes la posibilidad de ordenar aditivos para la pizza.	4. [Mantener] Ofrecer a los clientes la posibilidad de ordenar aditivos para la pizza.	4. [Mantener] Ofrecer a los clientes la posibilidad de ordenar aditivos para la pizza.
Ordenar pizza en el local o a domicilio.	5. [Mantener] Ordenar pizza en el local o a domicilio.	5. [Mantener] Ordenar pizza en el local o a domicilio.	5. [Mantener] Ordenar pizza en el local o a domicilio.	5. [Mantener] Ordenar pizza en el local o a domicilio.
Definir zona de cobertura.	6. [Lograr] Definir una zona de cobertura determinada.	6. [Lograr] Definir una zona de cobertura determinada.	6. [Lograr] Definir una zona de cobertura determinada.	6. [Lograr] Definir una zona de cobertura determinada.
Entregar el pedido al cliente en menos de 30 minutos.	7. Entregar el pedido al cliente en menos de 30 minutos.	7. Entregar el pedido al cliente en menos de 30 minutos.	7. Entregar el pedido al cliente en menos de 30 minutos.	7. Entregar el pedido al cliente en menos de 30 minutos.
Evitar entregar pedidos retardados.	8. [Evitar] Evitar entregar pedidos retardados.	8. [Evitar] Evitar entregar pedidos retardados.	8. [Evitar] Evitar entregar pedidos retardados.	8. [Evitar] Evitar entregar pedidos retardados.
Preparar la pizza ordenada por el cliente.	9. Preparar la pizza ordenada por el cliente.	9. Preparar la pizza ordenada por el cliente.	9. Preparar la pizza ordenada por el cliente.	9. Preparar la pizza ordenada por el cliente.
Ingresar el pedido en el sistema.	10. [Mantener] Ingresar el pedido en el sistema.	10. [Mantener] Ingresar el pedido en el sistema.	10. [Mantener] Ingresar el pedido en el sistema.	10. [Mantener] Ingresar el pedido en el sistema.

			sistema.	
Enviar la orden al cocinero.	11. Enviar la orden al cocinero.	11. Enviar la orden al cocinero.	11. Enviar la orden al cocinero.	11. Enviar la orden al cocinero.
Preparar la masa, los ingredientes y el horno.	12. Preparar la masa, los ingredientes y el horno.	12. Preparar la masa, los ingredientes y el horno.	12. Preparar la masa, los ingredientes y el horno.	12. Preparar la masa, los ingredientes y el horno.
Meter pizza al horno.	13. [Mantener] Meter la pizza al horno.	13. [Mantener] Meter la pizza al horno.	13. [Mantener] Meter la pizza al horno.	13. [Mantener] Meter la pizza al horno.
Despachar la orden.	14. [Parar] Despachar la orden.	14. [Parar] Despachar la orden.	14. [Parar] Despachar la orden.	14. [Parar] Despachar la orden.
	15. [Mantener] Entregar la pizza en menos de 30 minutos.	15. [Mantener] Entregar la pizza en menos de 30 minutos.	15. [Mantener] Entregar la pizza en menos de 30 minutos.	15. [Mantener] Entregar la pizza en menos de 30 minutos.
Pérdida de dinero.	16. [Evitar] Evitar la pérdida de dinero.	16. [Evitar] Evitar la pérdida de dinero.	16. [Evitar] Evitar la pérdida de dinero.	16. [Evitar] Evitar la pérdida de dinero.
Proponer promoción.				
Hallados AND Presentes	15	15	15	15
#Elementos presentes	16	16	16	16
#Elementos hallados	16	16	16	16
Exhaustividad	93,75%	93,75%	93,75%	93,75%
Precisión	93,75%	93,75%	93,75%	93,75%

Tabla B.6. Resultados de los agentes identificados por el modelo para el caso de estudio de la pizzería. Fuente: elaboración propia.

Agentes reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Administrador de la pizzería.	1. [Agente de ambiente] Administrador de pizzería.	1. [Agente de ambiente] Administrador de pizzería.	1. [Agente de ambiente] Administrador de pizzería.	1. [Agente de ambiente] Administrador de pizzería.
	2. Pizzería.	2. Pizzería.	2. Pizzería.	2. Pizzería.
Clientes.	3. [Agente de ambiente] Clientes.	3. [Agente de ambiente] Clientes.	3. [Agente de ambiente] Clientes.	3. [Agente de ambiente] Clientes.
	4. Promoción.	4. Promoción.	4. Promoción.	4. Promoción.
Repartidores.	5. [Agente de ambiente] Repartidores.	5. [Agente de ambiente] Repartidores.	5. [Agente de ambiente] Repartidores.	5. [Agente de ambiente] Repartidores.
	6. Empleados	6. Empleados	6. Empleados	6. Empleados
Operador logístico.	7. [Agente de ambiente] Operador logístico.	7. [Agente de ambiente] Operador logístico.	7. [Agente de ambiente] Operador logístico.	7. [Agente de ambiente] Operador logístico.
Cocinero.	8. [Agente de ambiente] Cocinero.	8. [Agente de ambiente] Cocinero.	8. [Agente de ambiente] Cocinero.	8. [Agente de ambiente] Cocinero.
Hallados AND Presentes	5	5	5	5
#Elementos presentes	5	5	5	5
#Elementos hallados	8	8	8	8
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	62,50%	62,50%	62,50%	62,50%

Tabla B.7. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio de la pizzería. Fuente: elaboración propia.

Entidades reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
1. Pizzería -administrador	1. Pizzería -administrador	1. Pizzería -administrador	1. Pizzería -administrador	1. Pizzería -administrador
2. Negocio -rentabilidad	2. Negocio -rentabilidad	2. Negocio -rentabilidad	2. Negocio -rentabilidad	2. Negocio -rentabilidad
3. Pizza -tamaño -aditivo -tipo	3. Pizza -tipo -tamaño	3. Pizza -tipo -tamaño	3. Pizza -tipo -tamaño	3. Pizza -tipo -tamaño
5. Cobertura -zona	4. Aditivo -posibilidad	4. Aditivo -posibilidad	4. Aditivo -posibilidad	4. Aditivo -posibilidad
7. Cliente	5. Cobertura -zona	5. Cobertura -zona	5. Cobertura -zona	5. Cobertura -zona
8. Repartidor	6. Dinero -pérdida	6. Dinero -pérdida	6. Dinero -pérdida	6. Dinero -pérdida
9. Entrega -tiempo				
Hallados AND Presentes	9	9	9	9
#Elementos presentes	14	14	14	14
#Elementos hallados	13	13	13	13
Exhaustividad	64,29%	64,29%	64,29%	64,29%
Precisión	69,23%	69,23%	69,23%	69,23%

Tabla B.8. Resultados de las operaciones identificadas por el modelo para el caso de estudio de la pizzería. Fuente: elaboración propia.

Operaciones reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Ofrecer posibilidad.				
Ofrecer tipos.				
Ofrecer tamaños.				
Recibir llamadas.	1. Recibir las llamadas.		1. Recibir las llamadas.	
Ingresar pedido.				
Realizar promoción.	2. Realizar promociones.		2. Realizar promociones.	
Enviar repartidores.	3. Enviar a los repartidores.	1. Enviar a los repartidores.	3. Enviar a los repartidores.	1. Enviar a los repartidores.
Entregar pizza.	4. Entregar la pizza.		4. Entregar la pizza.	
Hallados AND Presentes	4	1	4	1
#Elementos presentes	8	8	8	8
#Elementos hallados	4	1	4	1
Exhaustividad	50,00%	12,50%	50,00%	12,50%
Precisión	100,00%	100,00%	100,00%	100,00%

Caso de estudio Guayacán

Tabla B.9. Resultados de los objetivos identificados por el modelo para el caso de estudio Guayacán. Fuente: elaboración propia.

Objetivos reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. [Mantener] Ofrecer a las personas distintas guías turísticas del país.	1. Ofrecer a las personas distintas guías turísticas del país.	1. [Mantener] Ofrecer a las personas distintas guías turísticas del país.	1. [Mantener] Ofrecer a las personas distintas guías turísticas del país.
	2. [Mantener] Recorrer los caminos de Guatemala, mostrando la intensidad de sus paisajes y riqueza de su cultura.	2. Recorrer los caminos de Guatemala, mostrando la intensidad de sus paisajes y riqueza de su cultura.	2. [Mantener] recorrer los caminos de Guatemala, mostrando la intensidad de sus paisajes y riqueza de su cultura.	2. Recorrer los caminos de Guatemala, mostrando la intensidad de sus paisajes y riqueza de su cultura.
Incrementar su economía.	3. [Lograr] Incrementar su economía.	3. [Lograr] Incrementar su economía.	3. [Lograr] Incrementar su economía.	3. [Lograr] Incrementar su economía.
Ofrecer un servicio de buena calidad.	4. [Mantener] Ofrecer un servicio de buena calidad a los turistas y a sus principales clientes.	4. [Mantener] Ofrecer un servicio de buena calidad a los turistas y a sus principales clientes.	4. [Mantener] Ofrecer un servicio de buena calidad a los turistas y a sus principales clientes.	4. [Mantener] Ofrecer un servicio de buena calidad a los turistas y a sus principales clientes.
Tener empleados especializados	5. [Mantener] Tener empleados especializados en los temas viajeros.	5. [Mantener] Tener empleados especializados en los temas viajeros.	5. [Mantener] Tener empleados especializados en los temas viajeros.	5. [Mantener] Tener empleados especializados en los temas viajeros.

Contratar asesores especializados.	6. [Mantener] Contratar asesores especializados constantemente.	6. Contratar asesores especializados constantemente.	6. [Mantener] Contratar asesores especializados constantemente.	6. contratar asesores especializados constantemente.
Realizar un examen de valoración.	7. [Mantener] Realizar un examen de valoración para los empleados.	7. Realizar un examen de valoración para los empleados.	7. [Mantener] Realizar un examen de valoración.	7. Realizar un examen de valoración.
saber quiénes están cumpliendo con lo requerido				
Adquirir tecnología de primera calidad.	8. [Mantener] Adquirir tecnología de primera calidad.	8. Adquirir tecnología de primera calidad.	8. [Mantener] Adquirir tecnología de primera calidad.	8. [Mantener] Adquirir tecnología de primera calidad.
Organizar viajes a otros lugares turísticos.	9. [Mantener] Organizar viajes a otros lugares turísticos.	9. [Lograr] Organizar viajes a otros lugares turísticos.	9. [Mantener] Organizar viajes a otros lugares turísticos.	9. Organizar viajes a otros lugares turísticos.
	10. Ser fundamental el coordinador de la empresa.	10. Ser fundamental el coordinador de la empresa.	10. Ser fundamental el coordinador de la empresa.	10. Ser fundamental el coordinador de la empresa.
Gestionar los proyectos adecuadamente .	11. Gestionar los proyectos adecuadamente.	11. Gestionar los proyectos adecuadamente .	11. Gestionar los proyectos adecuadamente .	11. Gestionar los proyectos adecuadamente .
Realizar contratos con diferentes empresas.	12. [Mantener] Tener que realizar contratos con diferentes empresas.	12. [Lograr] Tener que realizar contratos con diferentes empresas.	12. [Mantener] Tener que realizar contratos con diferentes empresas.	12. [Lograr] Tener que realizar contratos con diferentes empresas .
Llevar un manejo de cliente.	13. [Mantener] llevar un manejo de clientes.	13. llevar un manejo de clientes.	13. [Mantener] llevar un manejo de clientes.	13. llevar un manejo de clientes.
Hallados AND Presentes	9	9	9	9
#Elementos presentes	11	11	11	11

#Elementos hallados	13	13	13	13
Exhaustividad	81,82%	81,82%	81,82%	81,82%
Precisión	69,23%	69,23%	69,23%	69,23%

Tabla B.10. Resultados de los agentes identificados por el modelo para el caso de estudio Guayacán. Fuente: elaboración propia.

Agentes reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Equipo de Guayacán	1. Equipo de Guayacán.	1. Equipo de Guayacán.	1. Equipo de Guayacán.	1. Equipo de Guayacán.
Guayacán	2. Guayacán	2. Guayacán	2. Guayacán	2. Guayacán
Empresa	3. Empresa	3. Empresa	3. Empresa	3. Empresa
Contratista de la empresa.	4. [Agente de ambiente] Contratista de empresa.	4. [Agente de ambiente] Contratista de empresa.	4. [Agente de ambiente] Contratista de empresa.	4. [Agente de ambiente] Contratista de empresa.
Director de servicio	5. [Agente de ambiente] Director de servicio.	5. [Agente de ambiente] Director de servicio.	5. [Agente de ambiente] Director de servicio.	5. [Agente de ambiente] Director de servicio.
Empleados				
Presidente	6. [Agente de ambiente] Presidente	6. [Agente de ambiente] Presidente	6. [Agente de ambiente] Presidente	6. [Agente de ambiente] Presidente
Sistema	7. [Agente de software] Sistema	7. [Agente de software] Sistema	7. sistema	7. sistema
	8. [Agente de ambiente] Presidente de empresa	8. [Agente de ambiente] Presidente de empresa	8. [Agente de ambiente] Presidente de empresa	8. [Agente de ambiente] Presidente de empresa
	9. Coordinador de empresa.	9. Coordinador de empresa.	9. Coordinador de empresa	9. Coordinador de empresa.

Coordinador	10. Coordinador	10. Coordinador	10. Coordinador	10. Coordinador
Hallados AND Presentes	8	8	8	8
#Elementos presentes	9	9	9	9
#Elementos hallados	10	10	10	10
Exhaustividad	88,89%	88,89%	88,89%	88,89%
Precisión	80,00%	80,00%	80,00%	80,00%

Tabla B.11. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio Guayacán. Fuente: elaboración propia.

Entidades reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
1. Paisaje -intensidad	1. Paisaje -intensidad	1. Paisaje -intensidad	1. Paisaje -intensidad	1. Paisaje -intensidad
	2. Cultura -riqueza	2. Cultura -riqueza	2. Cultura -riqueza	2. Cultura -riqueza
2. Calidad -servicio -tecnología	3. Calidad -servicio -tecnología	3. Calidad -servicio -tecnología	3. Calidad -servicio -tecnología	3. Calidad -servicio -tecnología
3. Empresa -contratista -presidente -coordinador	4. Empresa contratista -presidente -coordinador	4. Empresa contratista -presidente -coordinador	4. Empresa contratista -presidente -coordinador	4. Empresa contratista -presidente -coordinador
4. Servicio -director	5. Servicio -director	5. Servicio -director	5. Servicio -director	5. Servicio -director
5. Valoración -examen	6. Valoración -examen	6. Valoración -examen	6. Valoración -examen	6. Valoración -examen
6. Cliente -manejo	7. Cliente -manejo	7. Cliente -manejo	7. Cliente -manejo	7. Cliente -manejo
Hallados AND Presentes	10	10	10	10
#Elementos presentes	10	10	10	10
#Elementos hallados	17	17	17	17

Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	58,82%	58,82%	58,82%	58,82%

Tabla B.12. Resultados de las operaciones identificadas por el modelo para el caso de estudio Guayacán. Fuente: elaboración propia.

Operaciones reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Reportar resultados.	1. Reportar resultados.	1. Reportar resultados.	1. Reportar resultados.	1. Reportar resultados.
Leer resultados.	2. Leer resultados.		2. Leer resultados.	2. Leer resultados.
Almacenar datos.	3. Almacenar datos.		3. Almacenar datos.	3. Almacenar datos.
	4. Ingresar director.		4. Ingresar director.	
Enviar resultados.	5. Enviar resultados.	2. Enviar resultados.	5. Enviar resultados.	
Estudiar resultados.	6. Estudiar resultados.		6. Estudiar resultados	4. Estudiar resultados
Hallados AND Presentes	5	2	5	4
#Elementos presentes	5	5	5	5
#Elementos hallados	6	2	6	4
Exhaustividad	100,00%	40,00%	100,00%	80,00%
Precisión	83,33%	100,00%	83,33%	100,00%

Caso de estudio Marinozzi

Tabla B.13. Resultados de los objetivos identificados por el modelo para el caso de estudio Marinozzi. Fuente: elaboración propia.

Objetivos reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. [Mantener] Tener una visión futurista con un buen concepto respecto a su actualidad.	1. [Mantener] Tener una visión futurista con un buen concepto respecto a su actualidad.	1. [Mantener] Tener una visión futurista con un buen concepto respecto a su actualidad.	1. Tener una visión futurista con un buen concepto respecto a su actualidad.
Aumentar sus utilidades.	2. [Lograr] Aumentar sus utilidades.	2. Aumentar sus utilidades.	2. [Lograr] Aumentar sus utilidades.	2. [Lograr] Aumentar sus utilidades.
Crecer en el mercado de producción.	3. [Lograr] Crecer en el mercado de la producción.	3. Crecer en el mercado de la producción.	3. [Lograr] Crecer en el mercado de la producción.	3. Crecer en el mercado de la producción.
Ofrecer en el mercado alimentos variados en el sector.	4. [Mantener] Ofrecer en el mercado alimentos variados en el sector.	4. [Mantener] Ofrecer en el mercado alimentos variados en el sector.	4. [Mantener] Ofrecer en el mercado alimentos variados en el sector.	4. [Mantener] Ofrecer en el mercado alimentos variados en el sector.
Buscar la manera de ampliar sus instalaciones.	5. [Mantener] Buscar la manera de ampliar sus instalaciones.	5. Buscar la manera de ampliar sus instalaciones.	5. [Mantener] Buscar la manera de ampliar sus instalaciones.	5. [Mantener] Buscar la manera de ampliar sus instalaciones.
Cumplir con las demandas hechas por lo clientes.	6. [Mantener] Cumplir con las demandas hechas por los clientes.	6. Cumplir con las demandas hechas por los clientes.	6. [Mantener] Cumplir con las demandas hechas por los clientes.	6. [Mantener] Cumplir con las demandas hechas por los clientes.

Mirar la forma de buscar personal para laborar las 24 horas del día.	7. [Mantener] Mirar la forma de buscar personal para laborar las 24 horas del día.	7. Mirar la forma de buscar personal para laborar las 24_horas del día.	7. [Mantener] Mirar la forma de buscar personal para laborar las 24 horas del día.	7. [Mantener] Mirar la forma de buscar personal para laborar las 24 horas del día.
Hallados AND Presentes	6	6	6	6
#Elementos presentes	6	6	6	6
#Elementos hallados	7	7	7	7
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	85,71%	85,71%	85,71%	85,71%

Tabla B.14. Resultados de los agentes identificados por el modelo para el caso de estudio Marinozzi. Fuente: elaboración propia.

Agentes reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Gerente	1. [Agente de ambiente] Gerente	1. [Agente de ambiente] Gerente	1. [Agente de ambiente] Gerente	1. [Agente de ambiente] Gerente
Director	2. [Agente de ambiente] Director	2. [Agente de ambiente] Director	2. [Agente de ambiente] Director	2. [Agente de ambiente] Director
Presidente	3. [Agente de ambiente] Presidente	3. [Agente de ambiente] Presidente	3. [Agente de ambiente] Presidente	3. [Agente de ambiente] Presidente
Don Vicente	4. [Agente de ambiente] Don Vicente	4. [Agente de ambiente] Don Vicente	4. [Agente de ambiente] Don Vicente	4. [Agente de ambiente] Don Vicente
Empresa	5.Emprea	5. Empresa	5. Empresa	5. Empresa
	6. [Agente de ambiente] Contratista	6. [Agente de ambiente] Contratista	6. [Agente de ambiente] Contratista	6. [Agente de ambiente] Contratista
Hallados AND Presentes	5	5	5	5

#Elementos presentes	5	5	5	5
#Elementos hallados	6	6	6	6
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	83,33%	83,33%	83,33%	83,33%

Tabla B.15. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio Marinozzi. Fuente: elaboración propia.

Entidades reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
1. Trigo -producto	1. trigo -producto	1. trigo -producto	1. trigo -producto	1. trigo -producto
2. Producción -mercado	2. producción -mercado	2. producción -mercado	2. producción	2. producción -mercado
3. Proceso -resultado	3. proceso -resultado	3. proceso -resultado	3. proceso -resultado	3. proceso -resultado
	4. personal -forma	4. personal -forma	4. personal -forma	4. personal -forma
4. Cliente				
Hallados AND Presentes	6	6	6	6
#Elementos presentes	7	7	7	7
#Elementos hallados	8	8	8	8
Exhaustividad	85,71%	85,71%	85,71%	85,71%
Precisión	75,00%	75,00%	75,00%	75,00%

Tabla B.16. Resultados de las operaciones identificadas por el modelo para el caso de estudio Marinozzi. Fuente: elaboración propia.

Operaciones reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. Reconocer empresa.		1. Reconocer empresa.	
	2. Fundar año.	1. Fundar año.	2. Fundar año.	
	3. Cumplir objetivo.		3. Cumplir objetivo.	1. Cumplir objetivo.
Reportar resultados.	4. Reportar resultados.	2. Reportar resultados.	4. Reportar resultados.	2. Reportar resultados.
Medir ventas	5. Medir ventas	3. Medir ventas.	5. Medir ventas venta.	3. Medir ventas.
gestionar proceso				
Hallados AND Presentes	2	2	2	2
#Elementos presentes	3	3	3	3
#Elementos hallados	5	3	5	3
Exhaustividad	66,67%	66,67%	66,67%	66,67%
Precisión	40,00%	66,67%	40,00%	66,67%

Caso de estudio Carrefour

Tabla B.17. Resultados de los objetivos identificados por el modelo para el caso de estudio Carrefour. Fuente: elaboración propia.

Objetivos reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Ser líder de los supermercados de cadena.	1. Ser líder de los supermercados de cadena.	1. [Parar] Ser líder de los supermercados de cadena.	1. Ser líder de los supermercados de cadena.	1. [Parar] Ser líder de los supermercados de cadena.
Buscar alternativas para mejorar.	2. [Mantener] Buscar alternativas para mejorar.	2. [Mantener] Buscar alternativas para mejorar.	2. [Mantener] Buscar alternativas para mejorar.	2. [Mantener] Buscar alternativas para mejorar.
Vender productos de excelente calidad.	3. [Mantener] Vender productos de excelente calidad.	3. Vender productos de excelente calidad.	3. [Mantener] Vender productos de excelente calidad.	3. [Mantener] Vender productos de excelente calidad.
Buscar excelentes proveedores.	4. [Mantener] Buscar excelentes proveedores.	4. [Mantener] Buscar excelentes proveedores.	4. [Mantener] Buscar excelentes proveedores.	4. [Mantener] Buscar excelentes proveedores.
Existir en la empresa un control de calidad.	5. [Mantener] Existir en la empresa un control de calidad.	5. Existir en la empresa un control de calidad.	5. [Mantener] Existir en la empresa un control de calidad.	5. Existir en la empresa un control de calidad.
Vigilar el control de calidad.	6. Vigilar el control de calidad.	6. Vigilar el control de calidad.	6. Vigilar el control de calidad.	6. Vigilar el control de calidad.
Brindar una buena atención al cliente.	7. [Mantener] Brindar una buena atención al cliente.	7. [Mantener] Brindar una buena atención al cliente.	7. [Mantener] Brindar una buena atención al cliente.	7. [Mantener] Brindar una buena atención al cliente.

Contar con personal de excelentes competencias	8. [Lograr] Contar con personal de excelentes competencias.	8. Contar con personal de excelentes competencias.	8. [Lograr] Contar con personal de excelentes competencias.	8. [Lograr] Contar con personal de excelentes competencias.
Buscar mejores alternativas tecnológicas.	9. [Mantener] Buscar las mejores alternativas de tecnología.	9. Buscar las mejores alternativas de tecnología.	9. [Mantener] Buscar las mejores alternativas de tecnología.	9. [Mantener] Buscar las mejores alternativas de tecnología.
Mejorar la atención al usuario.	10. [Mantener] Mejorar la atención al usuario.	10. Mejorar la atención al usuario.	10. [Mantener] mejorar la atención al usuario.	10. mejorar la atención al usuario.
Hallados AND Presentes	10	10	10	10
#Elementos presentes	10	10	10	10
#Elementos hallados	10	10	10	10
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	100,00%	100,00%	100,00%	100,00%

Tabla B.18. Resultados de los agentes identificados por el modelo para el caso de estudio Carrefour. Fuente: elaboración propia.

Agentes reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. Actividad.	1. Actividad.	1. Actividad.	1. Actividad.
	2. Esto.	2. Esto.	2. Esto.	2. Esto.
Carrefour.	3. Carrefour.	3. Carrefour.	3. Carrefour.	3. Carrefour.
Jefe de producción.	4. [Agente de ambiente] Jefe de producción.	4. [Agente de ambiente] Jefe de producción.	4. [Agente de ambiente] Jefe de producción.	4. [Agente de ambiente] Jefe de producción.

Jefe de personal.	5. [Agente de ambiente] jefe de personal.	5. [Agente de ambiente] jefe de personal.	5. [Agente de ambiente] jefe de personal.	5. [Agente de ambiente] jefe de personal.
Presidente.	6. [Agente de ambiente] Presidente.	6. [Agente de ambiente] Presidente.	6. [Agente de ambiente] Presidente.	6. [Agente de ambiente] Presidente.
Hallados AND Presentes	4	4	4	4
#Elementos presentes	4	4	4	4
#Elementos hallados	6	6	6	6
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	66,67%	66,67%	66,67%	66,67%

Tabla B.19. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio Carrefour. Fuente: elaboración propia.

Entidades reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
1. Supermercado -líder	1. supermercado -líder	1. supermercado -líder	1. supermercado -líder	1. supermercado -líder
2. Cadena -supermercado	2. cadena -supermercado	2. cadena -supermercado	2. cadena -supermercado	2. cadena -supermercado
3. Calidad -producto -control	3. calidad -producto -control	3. calidad -producto -control	3. calidad -producto -control	3. calidad -producto -control
4. Producción -jefe	4. producción -jefe	4. producción -jefe	4. producción -jefe	4. producción -jefe
5. Competencia -personal	5. competencia -personal	5. competencia -personal	5. competencia -personal	5. competencia -personal
	6. personal	6. personal	6. personal	6. personal
	-jefe	-jefe	-jefe	-jefe
	7. tecnología	7. tecnología	7. tecnología	7. tecnología

	-alternativa	-alternativa	-alternativa	-alternativa
7. Marketing	8. marketing	8. marketing	8. marketing	8. marketing
-estrategia	-estrategia	-estrategia	-estrategia	-estrategia
6. Usuario -Id				
Hallados AND Presentes	13	13	13	13
#Elementos presentes	15	15	15	15
#Elementos hallados	17	17	17	17
Exhaustividad	86,67%	86,67%	86,67%	86,67%
Precisión	76,47%	76,47%	76,47%	76,47%

Tabla B.20. Resultados de las operaciones identificadas por el modelo para el caso de estudio Carrefour. Fuente: elaboración propia.

Operaciones reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. Distinguir cadena.		1. Distinguir cadena.	
	2. Centrar mercados.	1. Centrar mercados.	2. Centrar mercados.	1. Centrar mercados.
Reportar controles	3. Reportar controles.	2. Reportar controles.	3. Reportar controles.	2. Reportar controles.
Construir estrategias de marketing	4. Construir estrategias.		4. Construir estrategias.	3. Construir estrategias.
Hallados AND Presentes	2	1	2	2
#Elementos presentes	2	2	2	2
#Elementos hallados	4	2	4	3
Exhaustividad	100,00%	50,00%	100,00%	100,00%
Precisión	50,00%	50,00%	50,00%	66,67%

Caso de estudio Colácteos

Tabla B.21. Resultados de los objetivos identificados por el modelo para el caso de estudio Colácteos. Fuente: elaboración propia.

Objetivos reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Expandir su mercado a nivel nacional.	1. [Mantener] Expandir su mercado a nivel nacional.	1. Expandir su mercado a nivel nacional.	1. [Mantener] Expandir su mercado a nivel nacional.	1. Expandir su mercado a nivel nacional.
Buscar más inversionistas a nivel local y nacional.	2. Buscar más inversionistas a nivel local y nacional.	2. Buscar más inversionistas a nivel local y nacional.	2. Buscar más inversionistas a nivel local y nacional.	2. Buscar más inversionistas a nivel local y nacional.
	3. Aparecer a medida que se logre un reconocimiento de la empresa por los productos y su buena calidad.	3. Aparecer a medida que se logre un reconocimiento de la empresa por los productos y su buena calidad.	3. Aparecer a medida que se logre un reconocimiento de la empresa por los productos y su buena calidad.	3. Aparecer a medida que se logre un reconocimiento de la empresa por los productos y su buena calidad.
Certificar que los productos sean de buena calidad.	4. [Mantener] Certificar que los productos sean de buena calidad.	4. Certificar que los productos sean de buena calidad.	4. [Mantener] Certificar que los productos sean de buena calidad.	4. Certificar que los productos sean de buena calidad.
Construir sucursales.	5. [Mantener] Buscar la manera de construir sucursales en las principales ciudades del país.	5. Buscar la manera de construir sucursales en las principales ciudades del país.	5. [Mantener] Buscar la manera de construir sucursales en las principales ciudades del país.	5. [Mantener] Buscar la manera de construir sucursales en las principales ciudades del país.

Innovar en el mercado.	6. Innovar en el mercado con productos nuevos.	6. Innovar en el mercado con productos nuevos.	6. Innovar en el mercado con productos nuevos.	6. Innovar en el mercado con productos nuevos.
Proponer nuevas estrategias.	7. [Mantener] Proponer nuevas estrategias.	7. Proponer nuevas estrategias.	7. [Mantener] Proponer nuevas estrategias.	7. Proponer nuevas estrategias.
Innovar productos.	8. Innovar productos dependiendo de las necesidades del consumidor.	8. Innovar productos dependiendo de las necesidades del consumidor.	8. Innovar productos dependiendo de las necesidades del consumidor.	8. Innovar productos dependiendo de las necesidades del consumidor.
Hallados AND Presentes	7	7	7	7
#Elementos presentes	7	7	7	7
#Elementos hallados	8	8	8	8
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	87,50%	87,50%	87,50%	87,50%

Tabla B.22. Resultados de los agentes identificados por el modelo para el caso de estudio Colácteos. Fuente: elaboración propia.

Agentes reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. [Agente de software] Objetivo primordial.	1. [Agente de software] Objetivo primordial.	1. Objetivo primordial.	1. Objetivo primordial.
Colácteos.	2. Colácteos.	2. Colácteos.	2. Colácteos.	2. Colácteos.
Inversionistas.	3. Inversionistas.	3. Inversionistas.	3. Inversionista s.	3. Inversionistas.
Coordinador.	4. Coordinador.	4. Coordinador.	4. Coordinador.	4. Coordinador.

	5. [Agente de software] Productos.	5. [Agente de software] Productos.	5. [Agente de software] Productos.	5. [Agente de software] Productos.
Presidente.	6. [Agente de ambiente] Presidente.	6. [Agente de ambiente] Presidente.	6. [Agente de ambiente] Presidente.	6. [Agente de ambiente] Presidente.
Área de marketing.	7. [Agente de software] Área de marketing.	7. [Agente de software] Área de marketing.	7. Área de marketing.	7. Área de marketing.
Jefe de producción.	8. [Agente de ambiente] Jefe de producción.	8. [Agente de ambiente] Jefe de producción.	8. [Agente de ambiente] Jefe de producción.	8. [Agente de ambiente] Jefe de producción.
Hallados AND Presentes	6	6	6	6
#Elementos presentes	6	6	6	6
#Elementos hallados	8	8	8	8
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	75,00%	75,00%	75,00%	75,00%

Tabla B.23. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio Colácteos. Fuente: elaboración propia.

Entidades reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
1. Leche	1. Leche	1. Leche	1. Leche	1. Leche
-derivado	-derivado	-derivado	-derivado	-derivado
2. Empresa	2. Empresa	2. Empresa	2. Empresa	2. Empresa
-reconocimiento	-reconocimiento	-reconocimiento	-reconocimiento	-reconocimiento
	3. Sucursal	3. Sucursal	3. Sucursal	3. Sucursal
	-manera	-manera	-manera	-manera
4. País	4. País	4. País	4. País	4. País
-ciudad	-ciudad	-ciudad	-ciudad	-ciudad
5. Marketing	5. Marketing	5. Marketing	5. Marketing	5. Marketing
-área	-área	-área	-área	-área
6. Producción	6. Producción	6. Producción	6. Producción	6. Producción
-jefe	-jefe	-jefe	-jefe	-jefe

	7. Consumidor	7. Consumidor	7. Consumidor	7. Consumidor
	-necesidad	-necesidad	-necesidad	-necesidad
Hallados AND Presentes	10	10	10	10
#Elementos presentes	10	10	10	10
#Elementos hallados	14	14	14	14
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	71,43%	71,43%	71,43%	71,43%

Tabla B.24. Resultados de las operaciones identificadas por el modelo para el caso de estudio Colácteos. Fuente: elaboración propia.

Operaciones reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. Reconocer nivel.		1. Reconocer nivel.	
Realizar operaciones financieras.	2. Realizar operaciones.		2. Realizar operaciones.	
	3. Aumentar ingresos.		3. Aumentar ingresos.	1. Aumentar ingresos.
Reportar intereses.	4. Reportar intereses.	1. Reportar intereses.	4. Reportar intereses.	2. Reportar intereses.
Hallados AND Presentes	2	1	2	1
#Elementos presentes	2	2	2	2
#Elementos hallados	4	1	4	2
Exhaustividad	100,00%	50,00%	100,00%	50,00%
Precisión	50,00%	100,00%	50,00%	50,00%

Caso de estudio Colcerámicas

Tabla B.25. Resultados de los objetivos identificados por el modelo para el caso de estudio Colcerámicas. Fuente: elaboración propia.

Objetivos reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Mantener el liderazgo en las cerámicas del país.	1. [Mantener] Mantener el liderazgo en la producción de cerámicas del país.	1. [Mantener] Mantener el liderazgo en la producción de cerámicas del país.	1. [Mantener] Mantener el liderazgo en la producción de cerámicas del país.	1. [Mantener] Mantener el liderazgo en la producción de cerámicas del país.
Conseguir materiales de primera calidad.	2. [Mantener] Conseguir materiales de primera calidad.	2. Conseguir materiales de primera calidad.	2. [Mantener] Conseguir materiales de primera calidad.	2. Conseguir materiales de primera calidad.
Mantener una excelente comunicación con los proveedores.	3. [Mantener] Mantener una excelente comunicación con los proveedores.	3. Mantener una excelente comunicación con los proveedores.	3. [Mantener] Mantener una excelente comunicación con los proveedores.	3. [Mantener] Mantener una excelente comunicación con los proveedores.
Emplear maquinaria de primera calidad.	4. Emplear maquinaria de primera calidad y de muy buen desempeño.	4. Emplear maquinaria de primera calidad y de muy buen desempeño.	4. Emplear maquinaria de primera calidad y de muy buen desempeño.	4. Emplear maquinaria de primera calidad y de muy buen desempeño.
Adquirir maquinaria eficiente constantemente.	5. [Mantener] Adquirir maquinaria eficiente constantemente también.	5. Adquirir maquinaria eficiente constantemente también.	5. [Mantener] Adquirir maquinaria eficiente constantemente también.	5. Adquirir maquinaria eficiente constantemente también.
Mantener el mercado a favor.	6. [Mantener] Mantener el mercado a su favor.	6. [Mantener] Mantener el mercado a su favor.	6. [Mantener] Mantener el mercado a su favor.	6. [Lograr] Mantener el mercado a su favor.

Contratar personal de muy buen desempeño.	7. [Mantener] Contratar personal de muy buen desempeño.	7. Contratar personal de muy buen desempeño.	7. [Mantener] Contratar personal de muy buen desempeño.	7. Contratar personal de muy buen desempeño.
Contratar a los nuevos trabajadores.	8. [Mantener] Contratar a los nuevos trabajadores.	8. Contratar a los nuevos trabajadores.	8. [Mantener] Contratar a los nuevos trabajadores.	8. Contratar a los nuevos trabajadores.
Promover pequeños incentivos para mejorar.	9. [Lograr] Promover pequeños incentivos para mejorar el desempeño empresarial.	9. [Lograr] Promover pequeños incentivos para mejorar el desempeño empresarial.	9. [Lograr] Promover pequeños incentivos para mejorar el desempeño empresarial.	9. [Lograr] Promover pequeños incentivos para mejorar el desempeño empresarial.
Hallados AND Presentes	9	9	9	9
#Elementos presentes	9	9	9	9
#Elementos hallados	9	9	9	9
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	100,00%	100,00%	100,00%	100,00%

Tabla B.26. Resultados de los agentes identificados por el modelo para el caso de estudio Colcerámicas. Fuente: elaboración propia.

Agentes reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Colcerámicas.	1. Colcerámicas.	1. Colcerámicas.	1. Colcerámicas.	1. Colcerámicas.
Presidente.	2. Presidente.	2. Presidente.	2. Presidente.	2. Presidente.
	3. Mercado.	3. Mercado.	3. Mercado.	3. Mercado.
Ingeniero de procesos.	4. [Agente de ambiente] Ingeniero de procesos.	4. [Agente de ambiente] Ingeniero de procesos.	4. [Agente de ambiente] Ingeniero de procesos.	4. [Agente de ambiente] Ingeniero de procesos.

Ingeniero de control.	5. [Agente de ambiente] Ingeniero de control.	5. [Agente de ambiente] Ingeniero de control.	5. [Agente de ambiente] Ingeniero de control.	5. [Agente de ambiente] Ingeniero de control.
Gerente.	6. Gerente.	6. Gerente.	6. Gerente.	6. Gerente.
Grupo de trabajadores.	7. Grupo de trabajadores.	7. Grupo de trabajadores.	7. Grupo de trabajadores.	7. Grupo de trabajadores.
Jefe de personal.	8. [Agente de ambiente] Jefe de personal.	8. [Agente de ambiente] Jefe de personal.	8. [Agente de ambiente] Jefe de personal.	8. [Agente de ambiente] Jefe de personal.
Hallados AND Presentes	7	7	7	7
#Elementos presentes	7	7	7	7
#Elementos hallados	8	8	8	8
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	87,50%	87,50%	87,50%	87,50%

Tabla B.27. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio Colcerámicas. Fuente: elaboración propia.

Entidades reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. Empresa	1. Empresa	1. Empresa	1. Empresa
	-acecho	-acecho	-acecho	-acecho
	-progreso	-progreso	-progreso	-progreso
	2. Cerámica	2. Cerámica	2. Cerámica	2. Cerámica
	-empresa	-empresa	-empresa	-empresa
	-producción	-producción	-producción	-producción
	3. País	3. País	3. País	3. País
	-cerámica	-cerámica	-cerámica	-cerámica
1. Calidad	4. Calidad	4. Calidad	4. Calidad	4. Calidad
-material	-material	-material	-material	-material
-maquinaria	-maquinaria	-maquinaria	-maquinaria	-maquinaria
2. Proceso -ingeniero	5. Proceso -ingeniero	5. Proceso -ingeniero	5. Proceso -ingeniero	5. Proceso -ingeniero
3. Máquina -desempeño	6. Máquina -desempeño	6. Máquina -desempeño	6. Máquina -desempeño	6. Máquina -desempeño

4. Control -ingeniero	7. Control -ingeniero	7. Control -ingeniero	7. Control -ingeniero	7. Control -ingeniero
5. Evaluación -resultado	8. Evaluación -resultado	8. Evaluación -resultado	8. Evaluación -resultado	8. Evaluación -resultado
6. Ingeniero -evaluación	9. Ingeniero -evaluación	9. Ingeniero -evaluación	9. Ingeniero -evaluación	9. Ingeniero -evaluación
7. Trabajador -grupo	10. Trabajador -grupo	10. Trabajador -grupo	10. Trabajador -grupo	10. Trabajador -grupo
8. Personal -jefe	11. Personal -jefe	11. Personal -jefe	11. Personal -jefe	11. Personal -jefe
Hallados AND Presentes	17	17	17	17
#Elementos presentes	17	17	17	17
#Elementos hallados	25	25	25	25
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	68,00%	68,00%	68,00%	68,00%

Tabla B.28. Resultados de las operaciones identificadas por el modelo para el caso de estudio Colcerámicas. Fuente: elaboración propia.

Operaciones reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. Relacionar cerámica.	1. Relacionar cerámica.	1. Relacionar cerámica.	1. Relacionar cerámica.
	2. Estar acecho.		2. Estar acecho.	
Medir desempeño.	3. Medir desempeño máquinas.	2. Medir desempeño máquinas.	3. Medir desempeño máquinas.	2. Medir desempeño máquinas.
Reportar resultados.	4. Reportar resultados evaluación.	3. Reportar resultados evaluación.	4. Reportar resultados evaluación.	3. Reportar resultados evaluación.
	5. Conformar personas.		5. Conformar personas.	
	6. Comprometer progreso.	4. Comprometer progreso.	6. Comprometer progreso.	
Hallados AND	2	2	2	2

Presentes				
#Elementos presentes	2	2	2	2
#Elementos hallados	6	4	6	3
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	33,33%	50,00%	33,33%	66,67%

Caso de estudio ENCA

Tabla B.29. Resultados de los objetivos identificados por el modelo para el caso de estudio ENCA. Fuente: elaboración propia.

Objetivos reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. Medir el desempeño de los estudiantes.	1. Medir el desempeño de los estudiantes.	1. Medir el desempeño de los estudiantes.	1. Medir el desempeño de los estudiantes.
Integrar el sistema de educación y capacitación.	2. [Mantener] Integrar el sistema de educación y capacitación agrícola y ambiental.	2. integrar el sistema de educación y capacitación agrícola y ambiental.	2. [Mantener] integrar el sistema de educación y capacitación agrícola y ambiental.	2. integrar el sistema de educación y capacitación agrícola y ambiental.
Integrar a la universidad desde escuelas.	3. [Mantener] Integrar a la Universidad desde escuelas de formación agrícola.	3. Integrar a la Universidad desde escuelas de formación agrícola.	3. [Mantener] Integrar a la Universidad desde escuelas de formación agrícola.	3. Integrar a la Universidad desde escuelas de formación agrícola.

Garantizar realización de investigación.	4. [Mantener] Garantizar la realización de investigaciones de campo.	4. [Mantener] Garantizar la realización de investigaciones de campo.	4. [Mantener] Garantizar la realización de investigaciones de campo.	4. [Mantener] Garantizar la realización de investigaciones de campo.
Garantizar que los estudiantes tengan giras de estudio y prácticas.	5. [Mantener] Garantizar que los estudiantes tengan giras de estudio y prácticas supervisadas.	5. [Mantener] Garantizar que los estudiantes tengan giras de estudio y prácticas supervisadas.	5. [Mantener] Garantizar que los estudiantes tengan giras de estudio y prácticas supervisadas.	5. [Mantener] Garantizar que los estudiantes tengan giras de estudio y prácticas supervisadas.
Hallados AND Presentes	4	4	4	4
#Elementos presentes	4	4	4	4
#Elementos hallados	5	5	5	5
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	80,00%	80,00%	80,00%	80,00%

Tabla B.30. Resultados de los agentes identificados por el modelo para el caso de estudio ENCA. Fuente: elaboración propia.

Agentes reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Escuela Nacional Central de Agricultura.	1. Escuela Nacional Central de Agricultura.	1. Escuela Nacional Central de Agricultura.	1. Escuela Nacional Central de Agricultura.	1. Escuela Nacional Central de Agricultura.
	2. Área de finca	2.Área de finca	2. Área de finca	2. Área de finca
Rector.	3. [Agente de ambiente] Rector	3. [Agente de ambiente] Rector	3. [Agente de ambiente] Rector	3. [Agente de ambiente] Rector

Director general.	4. [Agente de ambiente] Director general.	4. [Agente de ambiente] Director general.	4. [Agente de ambiente] Director general.	4. [Agente de ambiente] Director general.
Docentes.	5. [Agente de ambiente] Docentes.	5. [Agente de ambiente] Docentes.	5. [Agente de ambiente] Docentes.	5. [Agente de ambiente] Docentes.
Asistente de dirección.	6. [Agente de ambiente] Asistente de dirección.	6. [Agente de ambiente] Asistente de dirección.	6. [Agente de ambiente] Asistente de dirección.	6. [Agente de ambiente] Asistente de dirección.
Estudiantes.	7. [Agente de ambiente] Estudiantes.	7. [Agente de ambiente] Estudiantes.	7. [Agente de ambiente] Estudiantes.	7. [Agente de ambiente] Estudiantes.
Hallados AND Presentes	6	6	6	6
#Elementos presentes	6	6	6	6
#Elementos hallados	7	7	7	7
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	85,71%	85,71%	85,71%	85,71%

Tabla B.31. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio ENCA. Fuente: elaboración propia.

Entidades reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
1. Finca -área	1. Finca -área	1. Finca -área	1. Finca -área	1. Finca -área
2. Mar -nivel	2. Mar -nivel	2. Mar -nivel	2. Mar -nivel	2. Mar -nivel
3. Educación -sistema -escuela	3. Educación -sistema -escuela	3. Educación -sistema -escuela	3. Educación -sistema -escuela	3. Educación -sistema -escuela
4. Formación -escuela	4. Formación -escuela	4. Formación -escuela	4. Formación -escuela	4. Formación -escuela

	5. Recurso -manejo	5. Recurso -manejo	5. Recurso -manejo	5. Recurso -manejo
	6. Estudio	6. Estudio	6. Estudio	6. Estudio
	-gira	-gira	-gira	-gira
5. Campo -investigación	7. Campo -investigación	7. Campo -investigación	7. Campo -investigación	7. Campo -investigación
6. Investigación -realización	8. Investigación -realización	8. Investigación -realización	8. Investigación -realización	8. Investigación -realización
	-realización	-realización	-realización	-realización
7. Dirección -asistente	9. Dirección -asistente	9. Dirección -asistente	9. Dirección -asistente	9. Dirección -asistente
8. Estudiante -desempeño	10. Estudiante -desempeño	10. Estudiante -desempeño	10. Estudiante -desempeño	10. Estudiante -desempeño
Hallados AND Presentes	17	17	17	17
#Elementos presentes	17	17	17	17
#Elementos hallados	21	21	21	21
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	80,95%	80,95%	80,95%	80,95%

Tabla B.32. Resultados de las operaciones identificadas por el modelo para el caso de estudio ENCA. Fuente: elaboración propia.

Operaciones reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
	1. Comunicar autopista		1. Comunicar autopista	
	2. Trasladar finca.		2. Trasladar finca.	1. Trasladar finca.
Construir proyectos productivos.	3. Construir proyectos.		3. Construir proyectos.	2. Construir proyectos.
	4. Supervisar investigaciones		4. Supervisar investigaciones.	
Medir desempeño.	5. Medir desempeño.	1. Medir desempeño.	5. Medir desempeño.	3. Medir desempeño.

Hallados AND Presentes	2	1	2	2
#Elementos presentes	2	2	2	2
#Elementos hallados	5	1	5	3
Exhaustividad	100,00%	50,00%	100,00%	100,00%
Precisión	40,00%	100,00%	40,00%	66,67%

Caso de estudio Yamaha

Tabla B.33. Resultados de los objetivos identificados por el modelo para el caso de estudio Yamaha. Fuente: elaboración propia.

Objetivos reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Tener una excelente gestión de sus recursos.	1. [Mantener] Tener una excelente gestión de sus recursos.	1. [Mantener] Tener una excelente gestión de sus recursos.	1. [Mantener] Tener una excelente gestión de sus recursos.	1. [Mantener] Tener una excelente gestión de sus recursos.
Brindar una administración de calidad.	2. [Mantener] Brindar una administración de calidad.	2. [Mantener] Brindar una administración de calidad.	2. [Mantener] Brindar una administración de calidad.	2. [Mantener] Brindar una administración de calidad.
Garantizar que la empresa cumpla con sus responsabilidades.	3. [Mantener] Garantizar que la empresa cumpla con sus responsabilidades.	3. [Mantener] Garantizar que la empresa cumpla con sus responsabilidades.	3. [Mantener] Garantizar que la empresa cumpla con sus responsabilidades.	3. [Mantener] Garantizar que la empresa cumpla con sus responsabilidades.
Garantizar el bienestar de los empleados.	4. [Mantener] Garantizar el bienestar de los empleados.	4. [Mantener] Garantizar el bienestar de los empleados.	4. [Mantener] Garantizar el bienestar de los empleados.	4. [Mantener] Garantizar el bienestar de los empleados.

Garantizar la satisfacción de las necesidades de los clientes.	5. [Mantener] Garantizar la satisfacción de las necesidades de los clientes.	5. [Mantener] Garantizar la satisfacción de las necesidades de los clientes.	5. [Mantener] Garantizar la satisfacción de las necesidades de los clientes.	5. [Mantener] Garantizar la satisfacción de las necesidades de los clientes.
Garantizar el bienestar de los empleados y sus familias.	6. [Mantener] Garantizar el bienestar de los empleados y sus familias.	6. [Mantener] Garantizar el bienestar de los empleados y sus familias.	6. [Mantener] Garantizar el bienestar de los empleados y sus familias.	6. [Mantener] Garantizar el bienestar de los empleados y sus familias.
Hallados AND Presentes	6	6	6	6
#Elementos presentes	6	6	6	6
#Elementos hallados	6	6	6	6
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	100,00%	100,00%	100,00%	100,00%

Tabla B.34. Resultados de los agentes identificados por el modelo para el caso de estudio Yamaha. Fuente: elaboración propia.

Agentes reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Yamaha	1. Yamaha	1. Yamaha	1. Yamaha	1. Yamaha
Gerente regional de YAMAHA.	2. [Agente de ambiente] Gerente regional de Yamaha.	2. [Agente de ambiente] Gerente regional de Yamaha.	2. [Agente de ambiente] Gerente regional de Yamaha.	2. [Agente de ambiente] Gerente regional de Yamaha.
	3. empresa	3. empresa	3. empresa	3. empresa
Jefe de recursos humanos.	4. [Agente de ambiente] Jefe de recursos humanos.	4. [Agente de ambiente] Jefe de recursos humanos.	4. [Agente de ambiente] Jefe de recursos humanos.	4. [Agente de ambiente] Jefe de recursos humanos.
Jefe.	5. [Agente de ambiente] Jefe.	5. [Agente de ambiente] Jefe.	5. [Agente de ambiente] Jefe.	5. [Agente de ambiente] Jefe.

Director de ventas.	6. [Agente de ambiente] Director de ventas.	6. [Agente de ambiente] Director de ventas.	6. [Agente de ambiente] Director de ventas.	6. [Agente de ambiente] Director de ventas.
Jefe de personal.	7. [Agente de ambiente] Jefe de personal.	7. [Agente de ambiente] Jefe de personal.	7. [Agente de ambiente] Jefe de personal.	7. [Agente de ambiente] Jefe de personal.
Hallados AND Presentes	6	6	6	6
#Elementos presentes	6	6	6	6
#Elementos hallados	7	7	7	7
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	85,71%	85,71%	85,71%	85,71%

Tabla B.35. Resultados de las entidades y atributos identificadas por el modelo para el caso de estudio Yamaha. Fuente: elaboración propia.

Entidades reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
1. Recurso -gestión -jefe	1. Recurso -gestión -jefe	1. Recurso -gestión -jefe	1. Recurso -gestión -jefe	1. Recurso -gestión -jefe
2. Calidad -administración	2. Calidad -administración	2. Calidad -administración	2. Calidad -administración	2. Calidad -administración
3. Empleado -bienestar	3. Empleado -bienestar -satisfacción	3. Empleado -bienestar -satisfacción	3. Empleado -bienestar -satisfacción	3. Empleado -bienestar -satisfacción
4. Venta -director	4. Venta -director	4. Venta -director	4. Venta -director	4. Venta -director
	-director	-director	-director	-director
	5. Necesidad -satisfacción	5. Necesidad -satisfacción	5. Necesidad -satisfacción	5. Necesidad -satisfacción
5. Cliente -necesidad -satisfacción	6. Cliente -necesidad -satisfacción	6. Cliente -necesidad -satisfacción	6. Cliente -necesidad -satisfacción	6. Cliente -necesidad -satisfacción
6. Personal -jefe	7. Personal -jefe	7. Personal -jefe	7. Personal -jefe	7. Personal -jefe

Hallados AND Presentes	14	14	14	14
#Elementos presentes	14	14	14	14
#Elementos hallados	17	17	17	17
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	82,35%	82,35%	82,35%	82,35%

Tabla B.36. Resultados de las operaciones identificadas por el modelo para el caso de estudio Yamaha. Fuente: elaboración propia.

Operaciones reales	Técnica de desambiguación utilizada			
	Mejor sentido y puntaje más alto		Puntos de grafo y mejor sentido	
	Técnica de similitud semántica utilizada			
	AdaptedLesk	Jiang y Conrath con IC	AdaptedLesk	Jiang y Conrath con IC
Medir satisfacción empleados.	1. Medir satisfacción empleados.	1. Medir satisfacción empleados.	1. Medir satisfacción empleados.	1. Medir satisfacción empleados.
Medir satisfacción clientes.	2. Medir satisfacción clientes.	2. Medir satisfacción clientes.	2. Medir satisfacción clientes.	2. Medir satisfacción clientes.
Hallados AND Presentes	2	2	2	2
#Elementos presentes	2	2	2	2
#Elementos hallados	2	2	2	2
Exhaustividad	100,00%	100,00%	100,00%	100,00%
Precisión	100,00%	100,00%	100,00%	100,00%

Anexo C. Implementación Ontología

Construcción de la Ontología

Para el desarrollo de la Ontología que se realizó en esta tesis, se tomó como punto de partida el metamodelo propuesto por (Matulevičius et al., 2007). Para la implementación, se tomaron cada una de las clases que se exponen en la figura 3.1, las cuales se modelaron teniendo en cuenta las generalizaciones existentes. Dado lo anterior, se logró obtener el modelo que se indica en las figuras C1 y C2, que corresponden a la estructura de las clases y al grafo que las representa respectivamente.



Figura C1. Estructura de las clases del meta-modelo diagramado en la Ontología.

Fuente: elaboración propia.

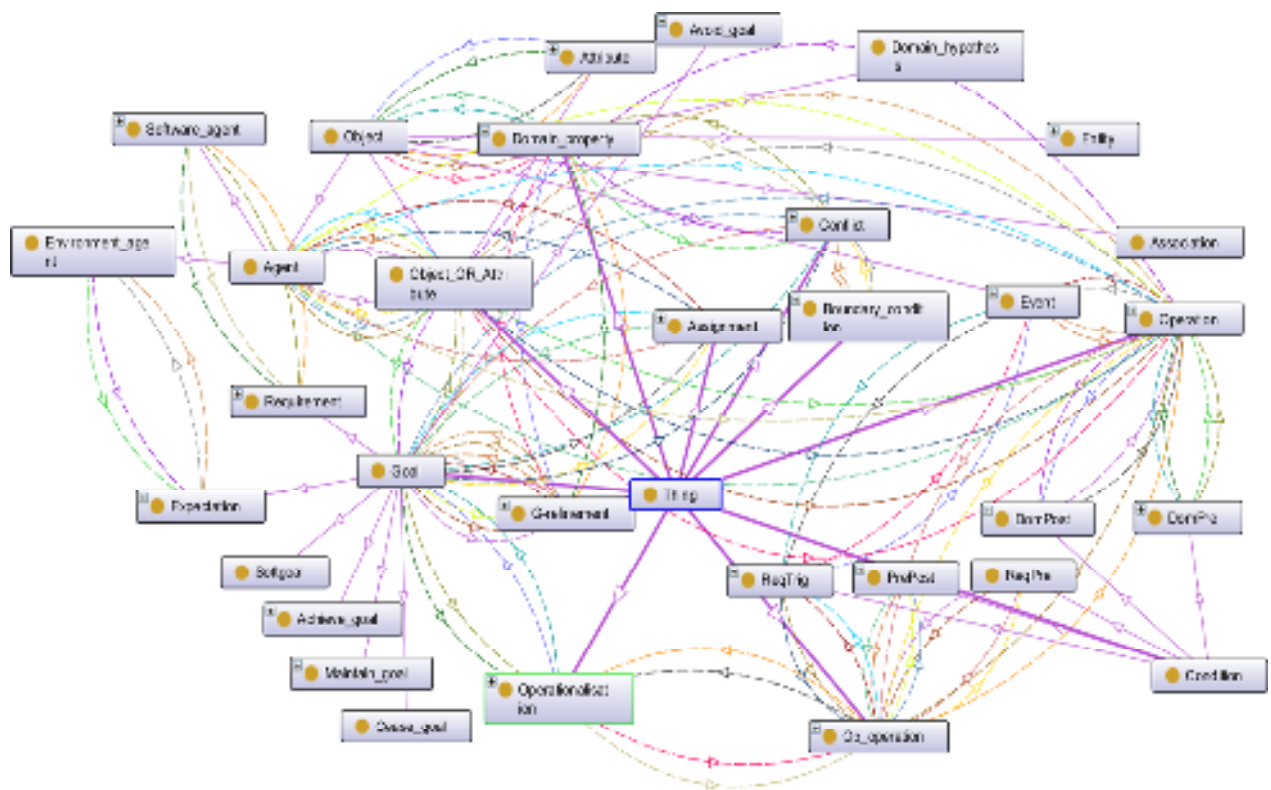


Figura C2. Grafo de la Ontología construida. Fuente: elaboración propia.

Propiedades de los datos (Data Properties)

Con los data properties se modelan los atributos de las clases. Dado lo anterior, y a manera de ejemplo, un objetivo tendría nombre, definición, prioridad, dueño, categoría y especificación formal. Sin embargo, y debido a que varias clases también tienen nombre, definición y especificación formal, se hizo necesario evitar inferencias incorrectas por parte del razonador, es por ello, que se crearon propiedades diferentes para cada atributo de cada entidad, de manera que, el concepto objetivo tenga un nombre del objetivo (nameG), la condición de límite, un nombre de condición de límite (nameBC), y análogamente para las demás entidades.

Si no se hace la distinción entre atributos, es decir, si se dejan varios elementos en el dominio (e.g. goal, boundary condition y domain property para formalSpec), el razonador infiere que cualquier instancia de estas clases es equivalente a las demás. Por ejemplo, en la figura C3, se puede observar que el razonador obtuvo que un objetivo (AccurateSpeed/LocationEstimates) es también una propiedad de dominio, lo cual no es cierto. La explicación del porqué concluyó eso, obedece a que la propiedad de dominio es parte del dominio de la data property de especificaciones formales (formalSpec).

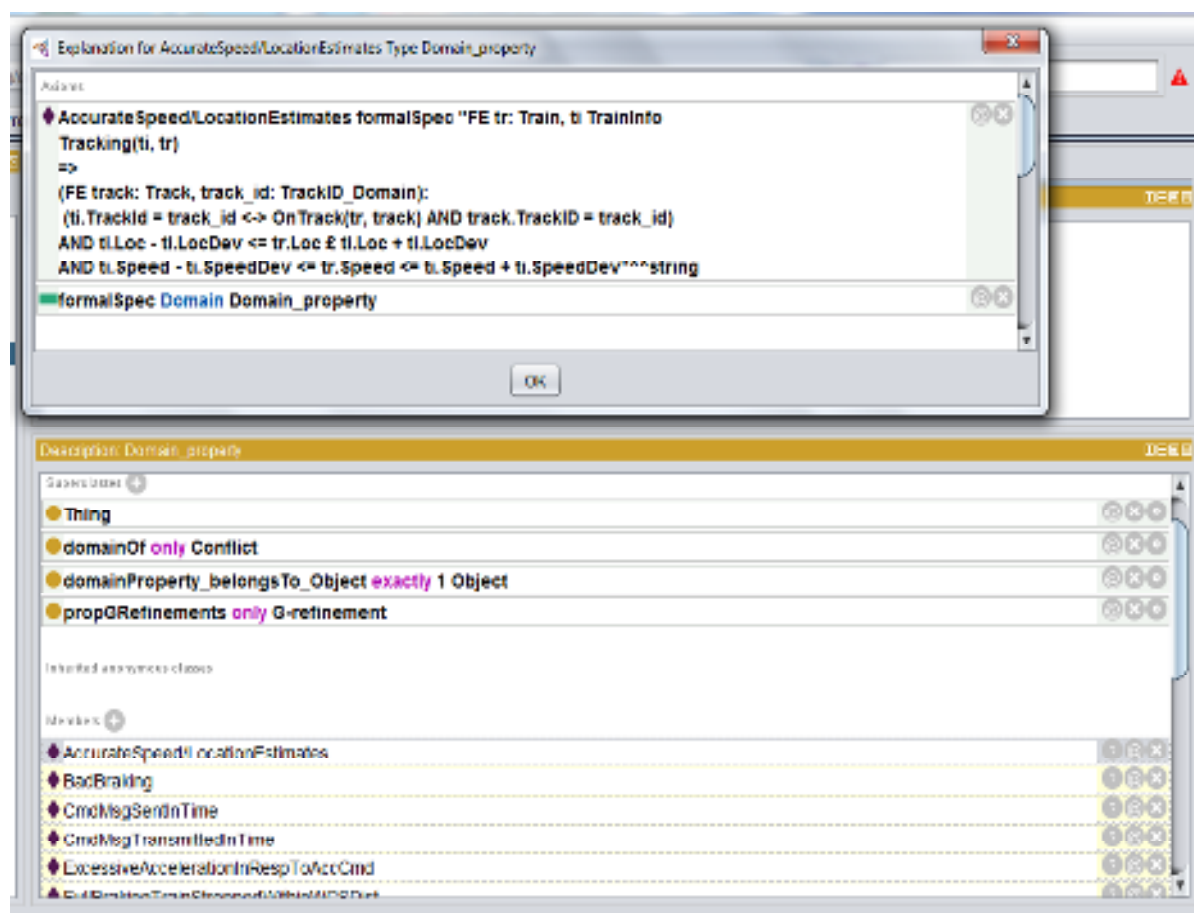


Figura C3. Inferencia incorrecta por parte del razonador, por tener varios elementos en el dominio de una propiedad de datos. Fuente: elaboración propia.

Propiedades de los objetos (Object Properties)

Los object properties suelen asociarse a relaciones entre las clases. Para cada relación deben salir dos propiedades de objetos. Por ejemplo, se tiene la relación `Software_agent isResponsibleFor Requirement` (ver figura 3.1), también hay que definir una relación `Requirement responsible_SoftwareAgent Software_agent`. De esta forma se procede para todas las relaciones entre clases.

Con el propósito de evitar el mismo problema que se presentaba con los data properties, cada una de las propiedades de los objetos se instanció de forma tal que tuviera sólo un tipo de dato para el dominio y otro para el rango. Adicionalmente, como habían propiedades cuyo nombre se repetía (e.g. `belongsTo`), los nombres de esas propiedades se completaron añadiendo el rango o el dominio. Por ejemplo, dado que existen varias propiedades que pertenecen a `Op_operation`, quedaron definidas como `reqPre_belongsTo_Op_operation` (nótese el `Op_operation` al final). Los conceptos se separan a través del guion bajo (`_`) para facilitar posibles tareas posteriores de lectura del XML de la Ontología.

Una de las ventajas de tener un metamodelo, es que permite visualizar relaciones binarias, entidad a entidad, de forma tal que, se puede detectar qué entidad es inversa a otra. De hecho, como se indicó anteriormente, todas las propiedades de objeto tienen una propiedad inversa, pues, si existe una relación de la entidad “A” a la “B”, esa misma relación se puede ver de “B” hacia “A”. Por ejemplo, hay una relación `attribute_belongsTo_Object`, que mapea `Attribute` a `Object`, entonces existe una relación inversa que mapea de `Object` a `Attribute` que es precisamente `hasAttributes`. Asimismo, con relación a las características de las propiedades, se indicó que si la relación es de: (i) uno a uno, es tanto funcional (`Functional`) como inversamente funcional (`Inverse functional`), (ii) muchos a uno, tan sólo es funcional, y (iii) uno a muchos, es inversamente funcional.

Las propiedades simétricas implican que tanto el dominio como el rango deben ser del mismo tipo, de lo contrario, el razonador asumirá que ambas son iguales. Además, significan que si “A” tiene una relación “P” con “B”, entonces “B” tiene también esa relación “P” con “A” (Horridge, 2011). La relación es de antisimetría, si no puede existir esa relación “P” de “B” hacia “A”. Por la forma en que se presenta el metamodelo, en principio, todas las relaciones son de antisimetría, pues están claramente diferenciadas y dadas entre entidades diferentes. Por ejemplo, “responsible_EnvironmentAgent” es asimétrica porque una expectativa no es agente de ambiente de un agente de ambiente.

Sin embargo, puede darse el caso en que un evento pueda ser causa de otro evento. Por tanto, tiene propiedades de transitividad. Para ello, se debe definir al evento como parte del dominio de esa relación, lo cual no resulta ser cierto. En consecuencia, simplemente se definen como un mismo evento.

En el caso de “controlledObjects” y “controllingAgents”, como un agente es un objeto, es posible que se dé la situación en la que un agente controle o monitoree a otro agente, por lo tanto, es más lógico que tenga control sobre sí mismo. Así que no sólo serían propiedades transitivas, sino también reflexivas. Sin embargo, aunque un agente es un objeto, la relación está dada de objeto a agente, y no de agente a agente. Es decir, si decimos que son transitivas y reflexivas, el razonador etiquetaría objetos, que no son agentes, como agentes.

Por otro lado, si un objetivo tiene conflictos con otros objetivos, el metamodelo explícitamente define un “conflicto”. En este caso no estaría bien señalar que existe simetría. Esta situación estaría bien si se hubiese definido directamente esta propiedad entre objetivos. Pero como la relación se da entre “objetivo” y “conflicto”, es necesario definirla como asimétrica e irreflexiva. Lo mismo ocurre con G-Refinement. Si fuera una relación entre objetivos, estaría bien decir que es transitiva. Pues si un objetivo se subroga en varios, y éstos en otros, el primero se

estaría definiendo también a través de la definición de los últimos. Sin embargo, la relación que se establece no es entre objetivos, sino con un elemento nuevo que hace parte del metamodelo “G-Refinement”.

Con respecto a la cardinalidad, ésta permite describir la cantidad de entidades que puede haber en el rango de una relación. Además, esa descripción se convierte en una definición, puesto que, establece condiciones necesarias para que se dé dicha relación. Por ejemplo, es necesario que un Object sea Object_OR_Attribute, y también es necesario que tenga al menos un atributo que sea miembro de la clase Attribute y que, si tiene hasDomainProperty, sea con un elemento de la clase Domain_property. Además, como hereda características de Object_OR_Attributes, también hereda sus clases anónimas (condiciones necesarias o necesarias y suficientes). Este ejemplo se puede apreciar mejor en la figura C4.

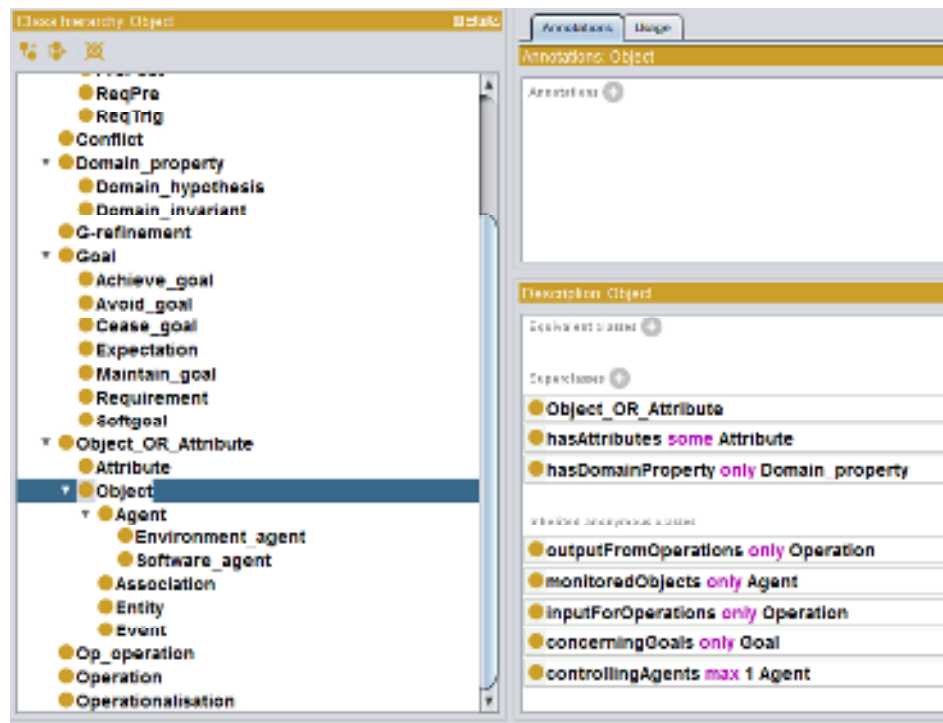


Figura C4. Clases anónimas para Object que definen condiciones necesarias a través de la cardinalidad. Fuente elaboración propia.

Anexo D. Casos de uso y diagrama de secuencias

En este anexo se pueden observar los casos de uso correspondientes a la arquitectura del modelo propuesto en esta tesis. Asimismo, se presenta un fragmento del diagrama de secuencia que representa la interacción que debe cumplir la aplicación desarrollada.

En las figuras D1 – D11, se presentan los diagramas de casos de uso que establecen las interacciones de los usuarios con el sistema, el sistema con los módulos internos y externos. Igualmente, en las figuras D12 y D13 se presentan los diagramas de secuencias.

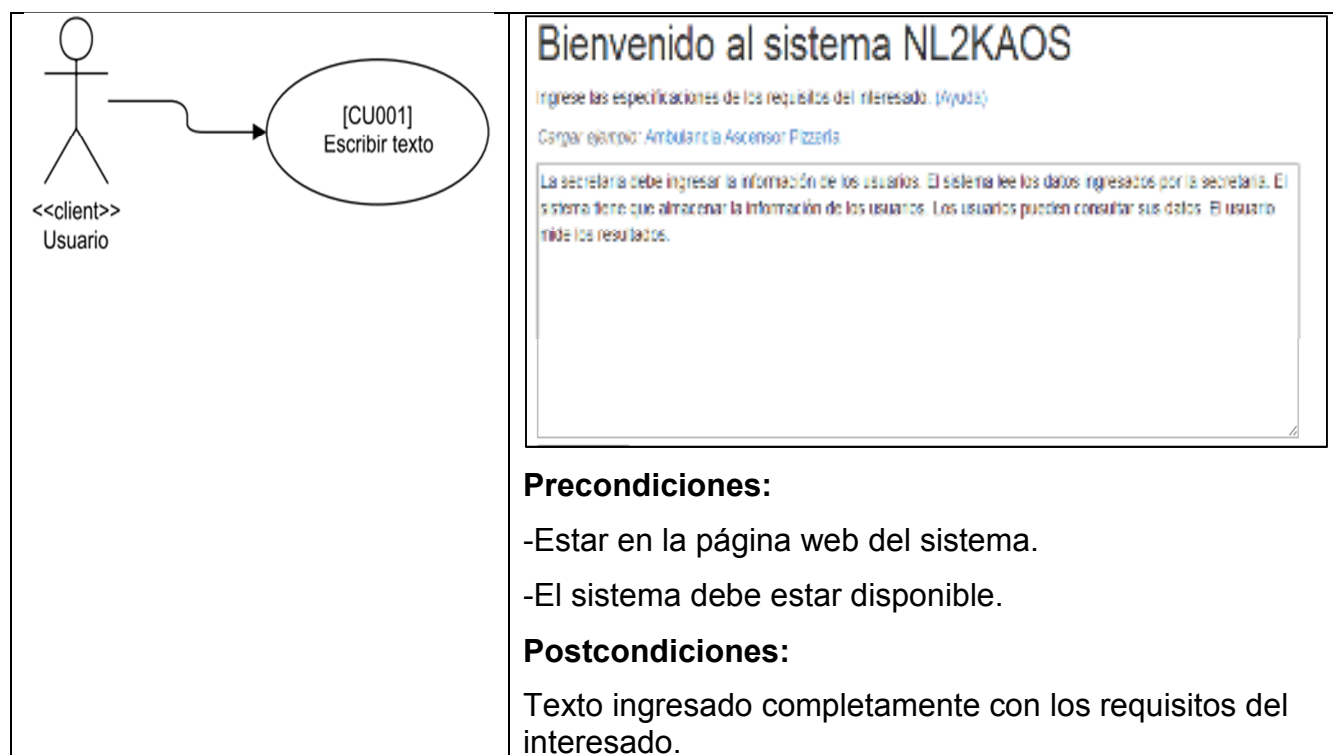
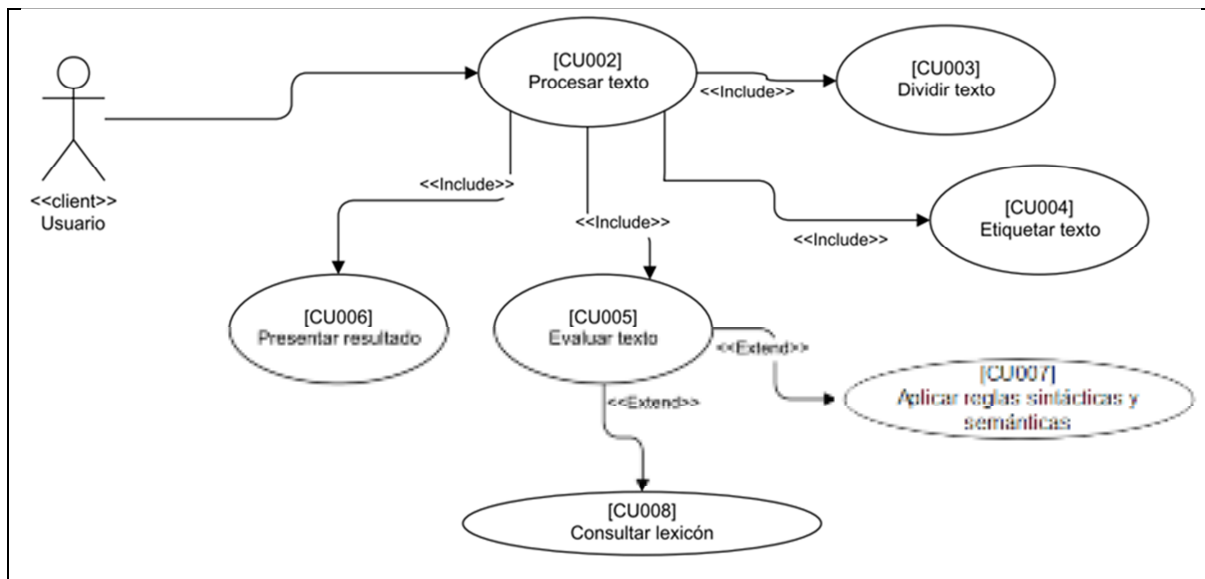


Figura D1. Caso de uso "Escribir texto". Fuente: elaboración propia.



gruposintelweb.tk:8081/KAOS/#

Bienvenido al sistema NL2KAOS

Ingrese las especificaciones de los requisitos del interesado. [\(Ayuda\)](#)

Cargar ejemplo: [Ambulancia](#) [Ascensor](#) [Pizzeria](#)

operador logístico llame a los auxiliares de la ambulancia. Por lo tanto, el operador logístico realiza llamadas telefónicas o por radio. El operador logístico debe darle los datos al auxiliar de la ambulancia para llegar al lugar del incidente. Así, el operador logístico comunica los datos.

Es menester que el operador de radio guíe al personal de ambulancia hacia el lugar del incidente. Se debe garantizar que el operador de radio reciba la información de la ambulancia seleccionada. El operador de radio debe dar instrucciones de localización al auxiliar de la ambulancia constantemente. El auxiliar de la ambulancia tiene que reportar el estado de la operación al operador de radio.

Seleccione el algoritmo que desea utilizar como método de desambiguación semántica:

☒ MFS
☐ UKB

Indique el paquete que desea utilizar para medir la similitud semántica:

☒ javasimlib
☐ CALPS

Precondiciones:

- El usuario debe de haber ingresado el texto correctamente. El texto debe estar escrito en voz activa, sin errores ortográficos, y deben ser explícitos todos los

sujetos de la oración.

- El usuario selecciona el algoritmo de desambiguación semántica (MFS o UKB)
- El usuario selecciona el paquete para medir la similitud semántica (javasimlib o CALPS).

Una vez el usuario presiona el botón “Procesar texto”, el sistema aplica las postcondiciones.

Postcondiciones:

El sistema realiza el procesamiento siguiendo el proceso indicado en el capítulo cuatro de esta tesis. El sistema divide el texto, etiqueta el texto morfosintácticamente. El sistema evalúa el texto aplicando reglas sintácticas y morfosintácticas (detalladas en el capítulo cuatro), consultando WordNet y el paquete de similitud semántica indicado por el usuario.

Figura D2. Caso de uso “Procesar texto”. Fuente: elaboración propia.

En la misma interfaz de la figura D2, el usuario da clic en el botón “Procesar texto” para realizar el procesamiento con todas las reglas descritas en esta tesis.

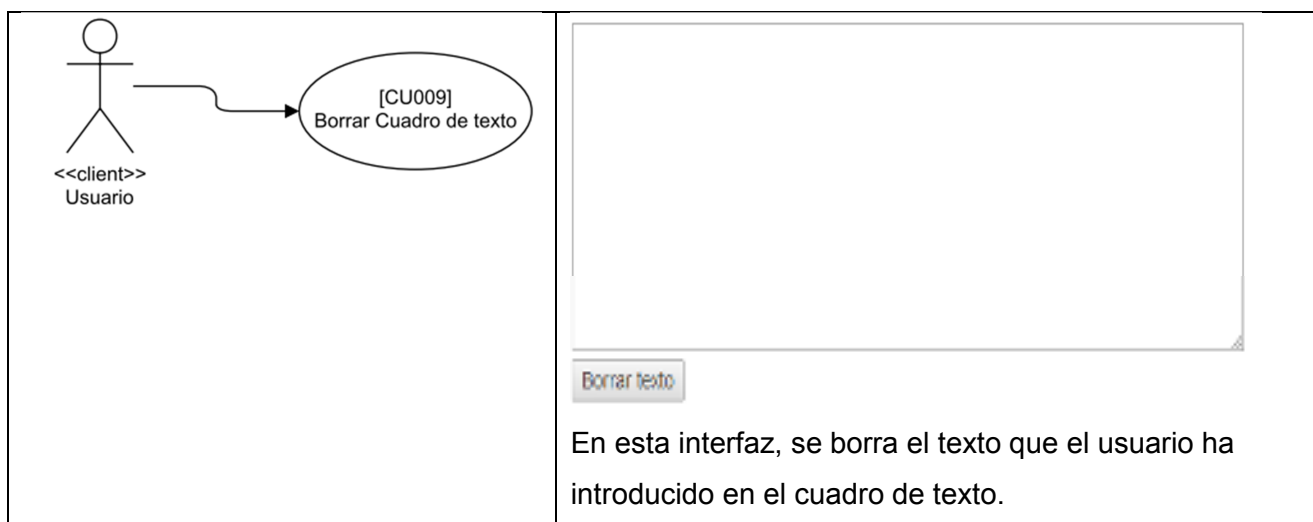


Figura D3. Caso de uso “Borrar Cuadro de texto”. Fuente: elaboración propia.

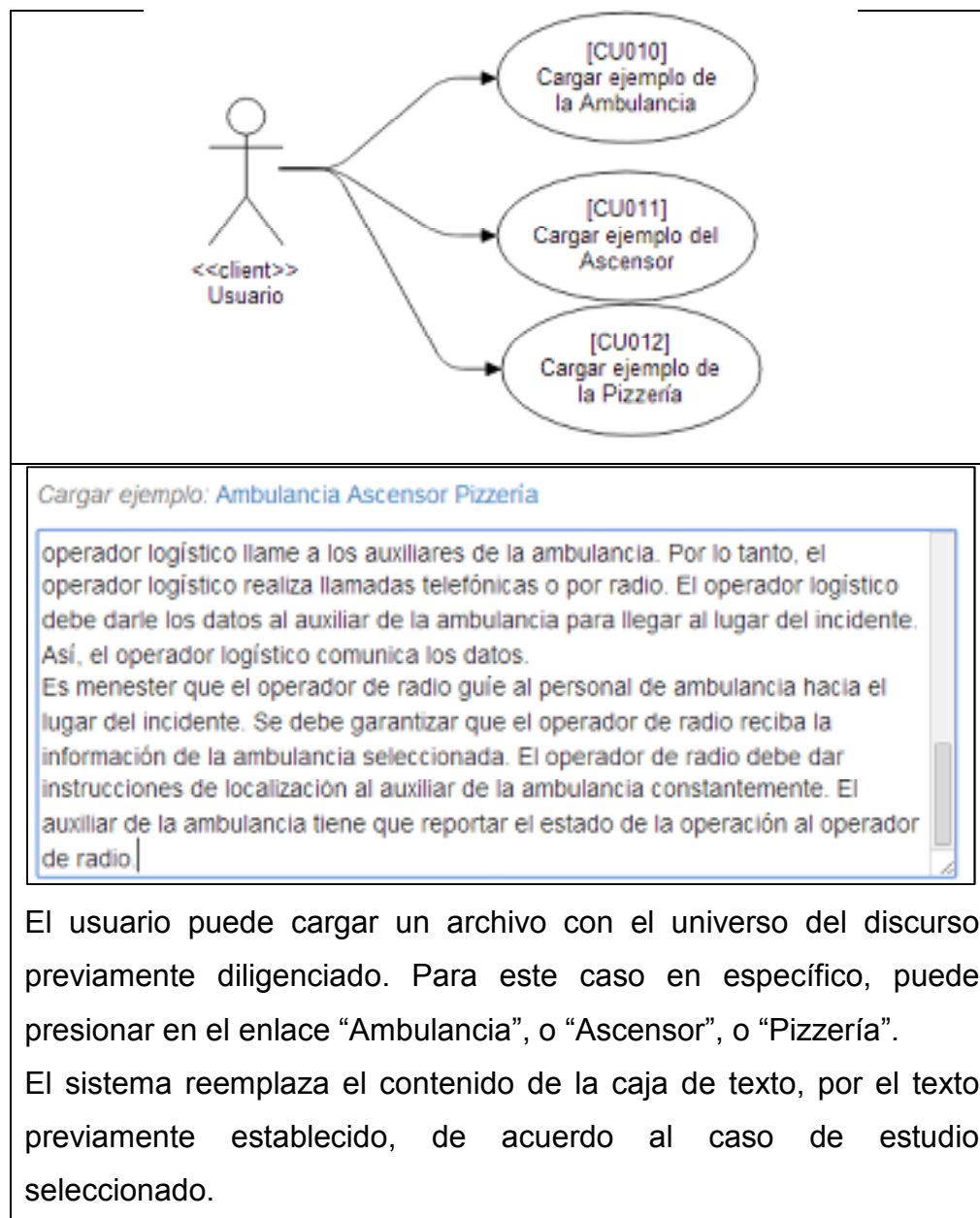


Figura D4. Casos de uso “Cargar ejemplo”. Fuente: elaboración propia.

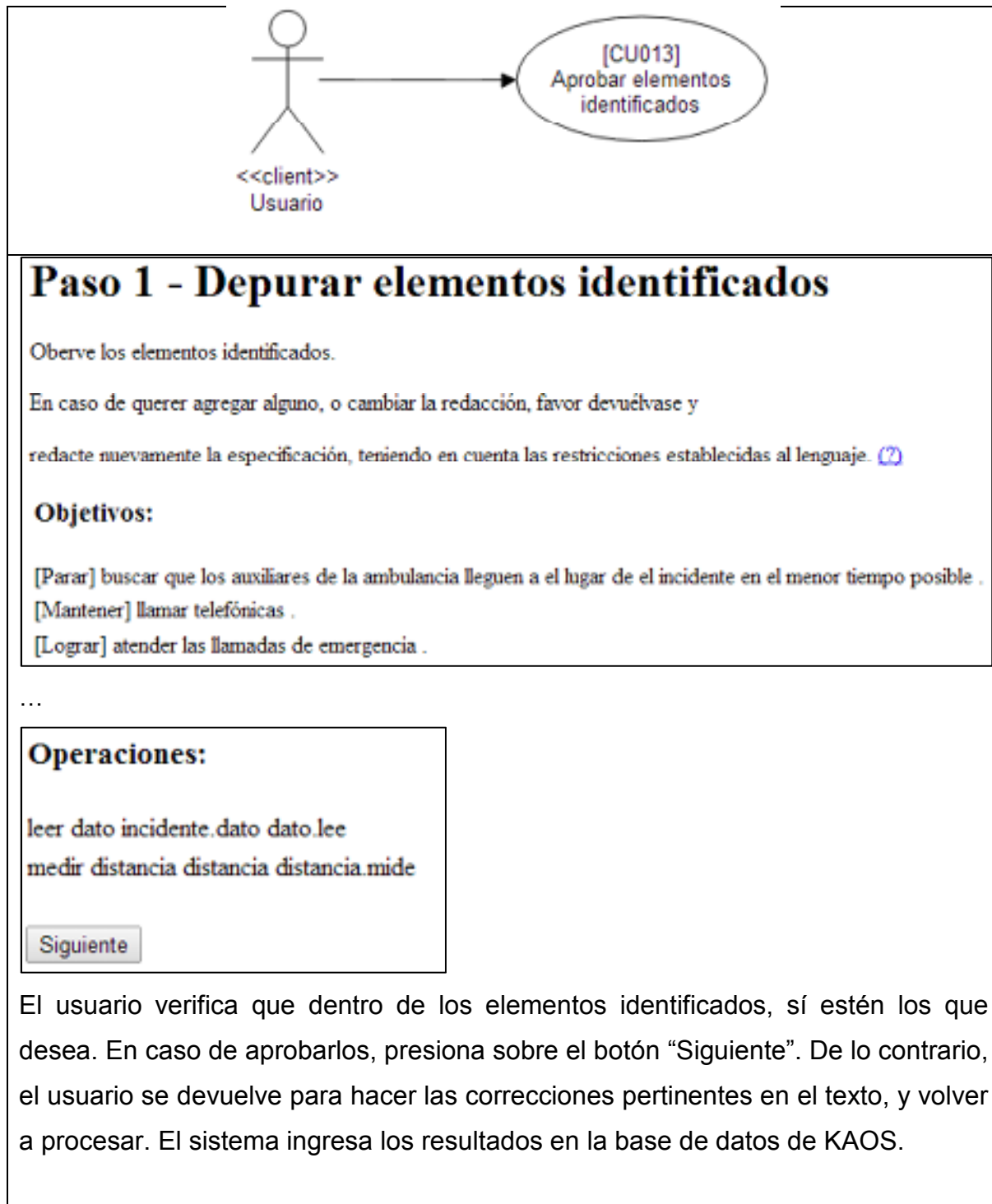
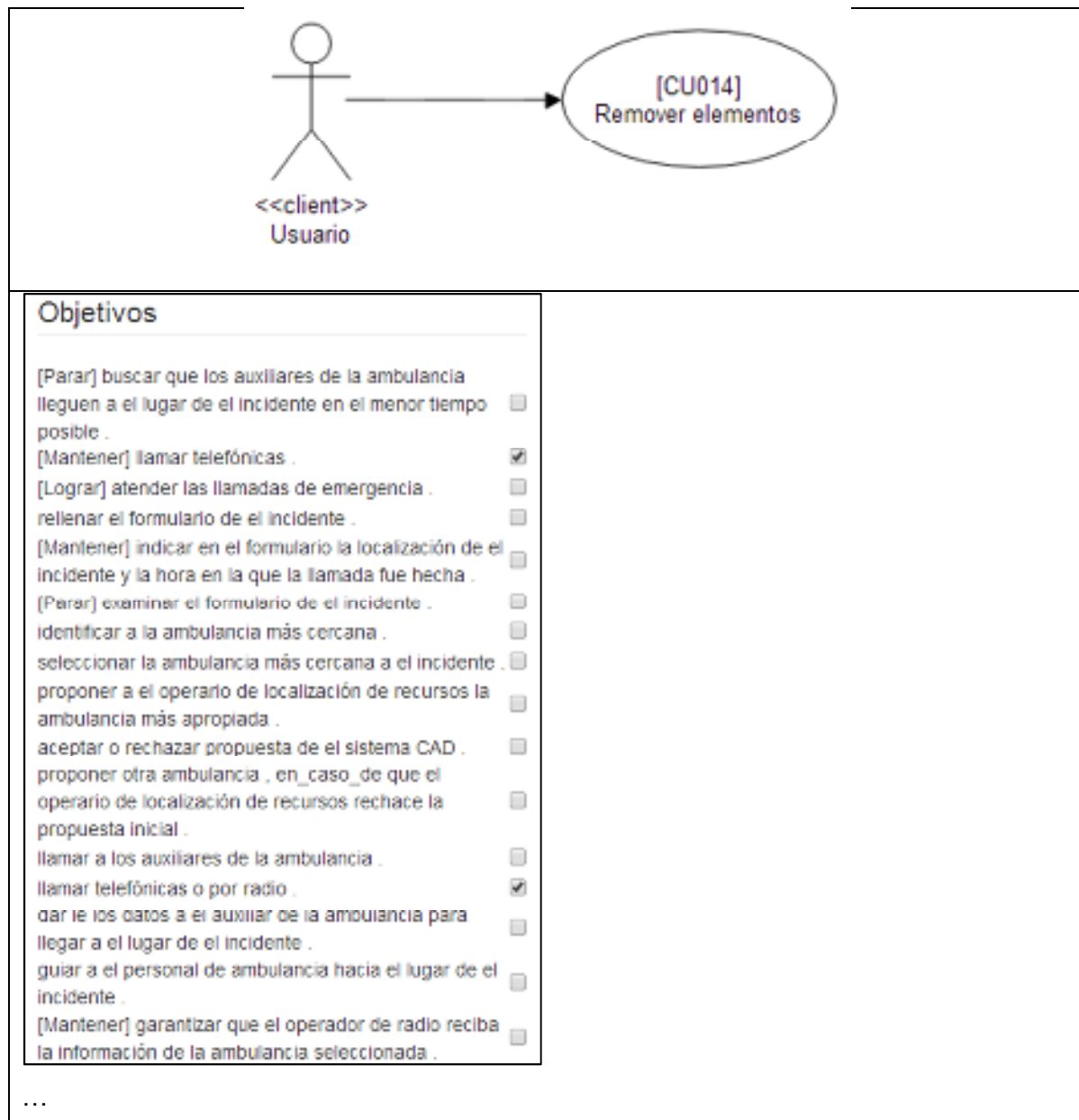


Figura D5. Aprobar elementos identificados. Fuente: elaboración propia



Agentes

[Agente de ambiente] auxiliares de ambulancia	<input checked="" type="checkbox"/>
[Agente de ambiente] asistente de control	<input type="checkbox"/>
[Agente de ambiente] operario de localización de recursos	<input type="checkbox"/>
sistema CAD	<input type="checkbox"/>
[Agente de ambiente] operador logístico	<input type="checkbox"/>
[Agente de ambiente] operador de radio	<input type="checkbox"/>
[Agente de ambiente] auxiliar de ambulancia	<input type="checkbox"/>

Operaciones

leer dato	<input type="checkbox"/>
medir distancia	<input type="checkbox"/>

Enviar

El usuario selecciona aquellos elementos que considera no son pertinentes. Luego presiona el botón “Enviar”. El sistema almacena esta información en la base de datos.

Figura D6. Remover elementos. Fuente: elaboración propia

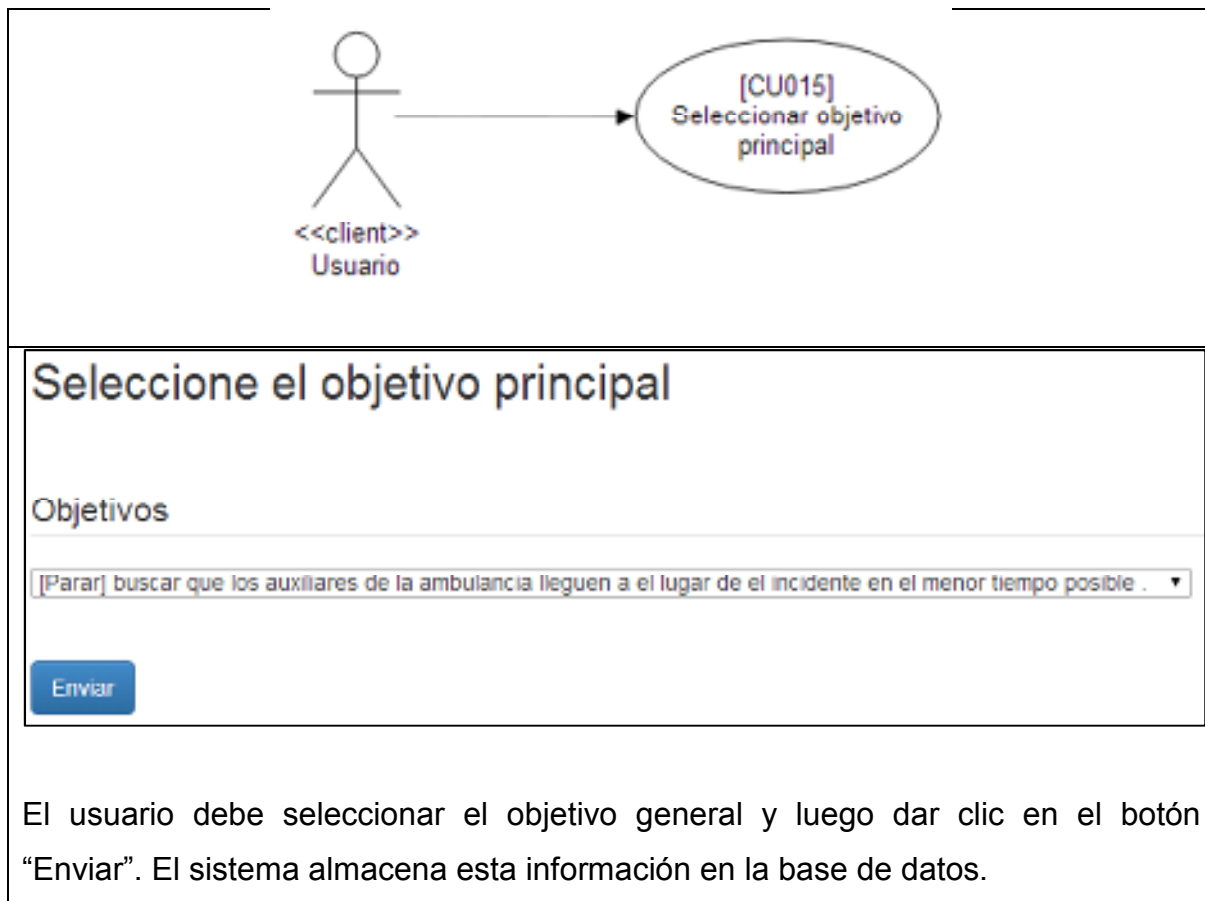


Figura D7. Seleccionar el objetivo principal. Fuente: elaboración propia



Seleccione el padre para cada objetivo

Objetivos

Objetivo	Padre
[lograr] eliminar los formularios de emergencia.	[reflexionar] formularios de emergencia.
[reflexionar] formularios de emergencia.	[seleccionar] la información más relevante del incidente.
[planificar] analizar el formato y la localización del incidente y la hora en la que se formulará el formulario.	[reflexionar] formularios de emergencia.
[planificar] analizar el formato de emergencia.	[identificar] a quienes más les interesa.
[planificar] analizar el formato de emergencia.	[seleccionar] la información más relevante del incidente.
[planificar] analizar el formato de emergencia.	[planificar] analizar que los formularios de emergencia se registren al lugar de emergencia en el momento.
[planificar] analizar el formato de emergencia.	[aceptar] recibir el formulario de emergencia.
[planificar] analizar el formato de emergencia.	[identificar] a quienes más les interesa.
[planificar] analizar el formato de emergencia.	[aceptar] recibir el formulario de emergencia.
[planificar] analizar el formato de emergencia.	[planificar] analizar que los formularios de emergencia se registren al lugar de emergencia en el momento.

...

[planificar] analizar el formato de emergencia.	[planificar] analizar que los formularios de emergencia se registren al lugar de emergencia en el momento.
[planificar] analizar el formato de emergencia.	[planificar] analizar que los formularios de emergencia se registren al lugar de emergencia en el momento.
[planificar] analizar el formato de emergencia.	[planificar] analizar que los formularios de emergencia se registren al lugar de emergencia en el momento.
[planificar] analizar el formato de emergencia.	[planificar] analizar que los formularios de emergencia se registren al lugar de emergencia en el momento.

Enviar

El usuario indica para cada objetivo, cuál es el padre. Como no tiene porque conocer la estructura del diagrama, el sistema (software) lo orienta para realizar esta tarea, proponiéndole que se pregunte: “¿por qué necesita cumplir ese objetivo?”, “¿cómo se puede alcanzar ese objetivo?”. Una vez el usuario haya establecido la jerarquía de todos los objetivos, presiona el botón “Enviar” y el sistema almacena estos datos en la base de datos.

Figura D8. Establecer jerarquía entre objetivos. Fuente: elaboración propia

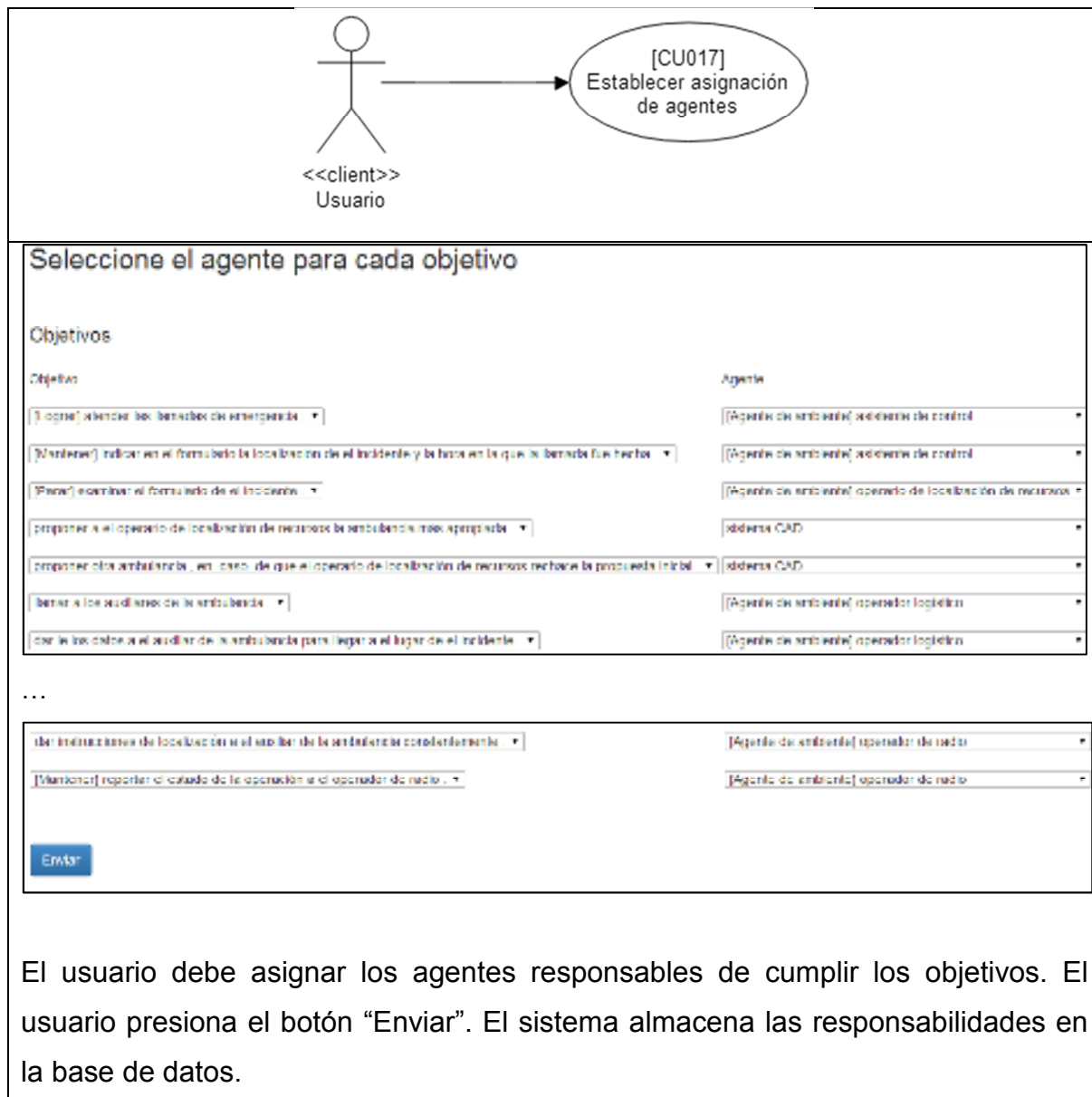
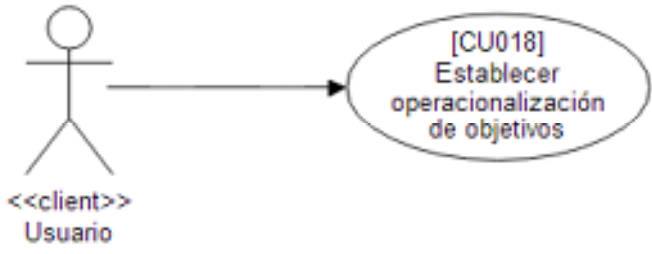


Figura D9. Establecer asignación de agentes. Fuente: elaboración propia



Seleccione la operación para cada objetivo

Objetivos

Objetivo	Operation
[Lograr] atender las llamadas de emergencia . ▾	▾
rellenar el formulario de el incidente . ▾	▾
[Mantener] indicar en el formulario la localización de el incidente y la hora en la que la llamada fue hecha . ▾	▾
[Parar] examinar el formulario de el incidente . ▾	leer dato ▾
identificar a la ambulancia más cercana . ▾	▾
seleccionar la ambulancia más cercana a el incidente . ▾	▾
proponer a el operario de localización de recursos la ambulancia más apropiada . ▾	medir distancia ▾

...

[Mantener] reportar el estado de la operación a el operador de radio . ▾ ▾

+ Agregar Campo

Enviar

El usuario indica la operación que corresponde a cada objetivo operacionalizable. En el caso de que un objetivo posea más de una operación, el usuario da clic en el botón “+ Agregar Campo”, seleccionando la operación y el objetivo respectivo. Cuando finalice, el usuario da clic en el botón “Enviar” y el sistema almacena la información procesada en la base de datos.

Figura D10. Establecer operacionalización de objetivos. Fuente: elaboración propia.

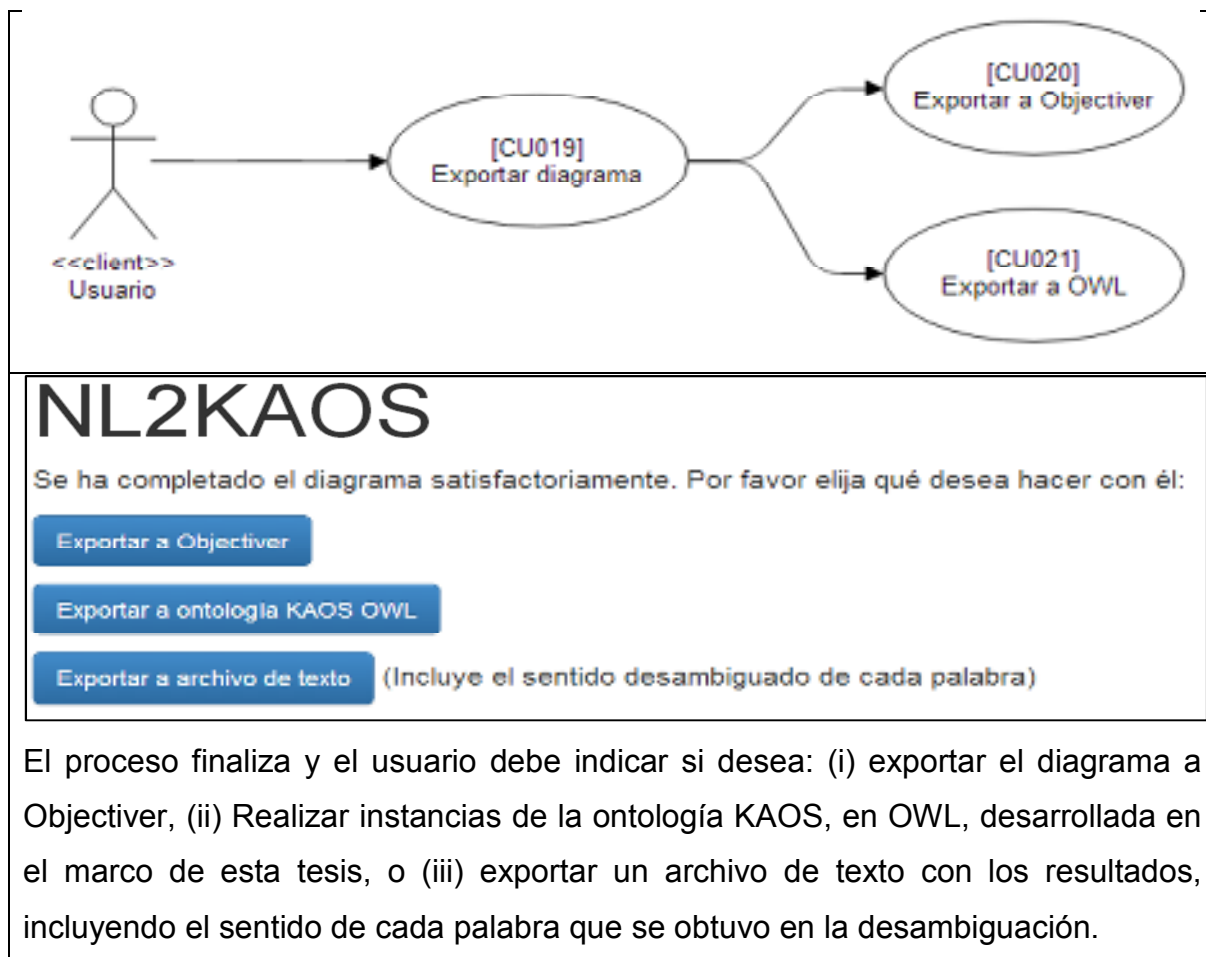


Figura D11. Exportar diagrama. Fuente: elaboración propia.

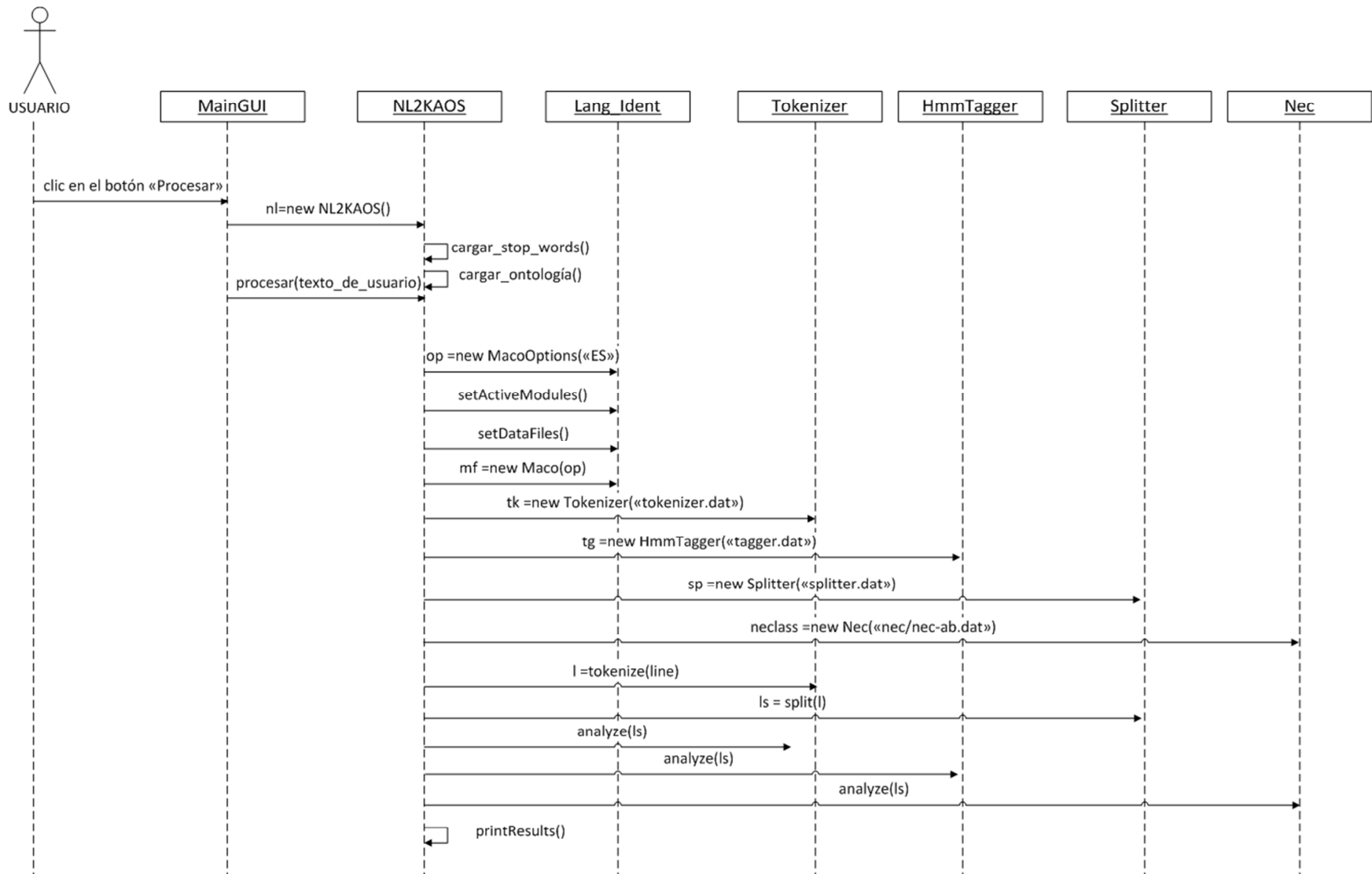


Figura D12. Fragmento 1 del diagrama de secuencias de la aplicación desarrollada. Fuente: elaboración propia.

REFERENCIAS

Agirre, E., y Soria, A. (2009). Personalizing PageRank for Word Sense Disambiguation. En: Proceedings of the 12th conference of the European chapter of the Association for Computational Linguistics (EACL-2009). Athens, Greece.

Agirre, E., y Rigau, G. (1996). Word sense disambiguation using conceptual density. En Proceedings of the 16th conference on Computational linguistics. Volume 1, pp. 16–22.

Agirre, E., y Soria, A. (2009). Personalizing pagerank for word sense disambiguation. En: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, pp. 33–41.

Almished, F., y Keppens, J. (2010). Requirements Analysis: Evaluating KAOS Models. En: 2nd International Workshop for Requirements Analysis, pp. 869-874. Londres.

Alrajeh, D., Russo, A., Uchitel, S. (2006). Inferring operational requirements from scenarios and goal models using inductive learning. En: Intl. Workshop on Scenarios and State Machines: Models, Algorithms, and Tools, ACM, Shanghai, China.

Antón, A. (1996). Goal-Based Requirements Analysis. En: Proceedings of the Second IEEE International Conference on Requirements Engineering. Colorado Springs, USA, pp. 136-144.

Antón, A. (1997). Goal Identification and Refinement in the Specification of Software-Based Information Systems, PhD Thesis. En: Georgia Institute of Technology, Atlanta, USA.

Antón, A. (1998). The Use of Goals to surface Requirements for Evolving System. Software Engineering. En: Proceedings of the 1998 (20th) International Conference. Kyoto, Japan, pp 157-166.

Antón, A., Potts, C. (1998). The use of goals to surface requirements for evolving systems. En: 20th Intl. Conf. on Software Eng. (ICSE'98), IEEE Computer Society, Kyoto, Japan.

Arango, F., y Zapata, C. (2006). UN-MÉTODO para la elicitación de requisitos de software. En: Escuela de Sistemas de la Universidad Nacional de Colombia, Medellín, Colombia. 113 p. ISBN: 958-33-9717-2.

Atserias, J., Comelles, E., and Mayor A. (2005). Txala un analizador libre de dependencias para el castellano. En: Procesamiento del Lenguaje Natural, (35):455–456.

Artiles, J., Gonzalo, J., y Sekine, S. (2007). The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. En: Proceedings of the 4th International Workshop on Semantic Evaluations, 2007, pp. 64–69.

Atserias, J., Villarejo, L., Rigau, G., Agirre, E., Carroll, J., Magnini, B., y Vossen, P. (2004). The meaning Multilingual Central Repository. En: Proceedings of GWC, Brno, Czech Republic.

Atserias, J., y Rodríguez, H. (1998). Tacat: Tagged corpus analyzer tool. En: Technical report Isi-98-2-t, Departament de LSI. Universitat Politècnica de Catalunya.

Baeza, Y., y Berthier, R. (1999). Modern information retrieval. En: ACM Publisher, Ed. New York.

Brants, T. (2000). Tnt - a statistical part- of-speech tagger. En: Proceedings of the 6th Conference on Applied Natural Language Processing, ANLP. ACL.

Bunge, M. (1977). Treatise on Basic Philosophy: Ontology I. En: The Furniture of the World (Vol. 3). Boston: D. Reidel.

Booch, G., Rumbaugh, J., Jacobson, I., Martínez, J., y Molina, J. (2000). El lenguaje unificado de modelado. En: Addison Wesley.

Byars, L. (1984). Strategic management: planning and implementation concepts and cases. En: Harper and Row, Publishers. New York, NY

Carreras, X., Márquez, L., y Padró, L. (2002). Named entity extraction using adaboost. En: Proceedings of CoNLL Shared Task, pages 167–170, Taipei, Taiwan.

Cueto, A. (2004). Resolución de la Ambigüedad Semántica de las palabras mediante Modelos de Probabilidad de Máxima Entropía. PhD Tesis. En: Universidad de Alicante, España.

Cvitas, A. (2011). Relation Extraction from Text Documents. En: MIPRO, 2011 Proc. 34th International Convention, pp. 1565-1570. Opatija, Croacia: IEEE.

Davis, A. (1993). *Software Requirements, Objetos, Funciones y Estados*. En: Editorial Prentice Hall, ISBN 0-13-805763-X.

De Landtsheer, R., Letier, E., y Lamsweerde, A. (2004). Deriving tabular event-based specifications from goal-oriented requirements models, *Requirements Eng.* 9 (2), 104–120.

Demonte, V., y Bosque, I. (1999). *Gramática descriptiva de la lengua española*. En: Editorial: Espasa C. España.

DRAE: Diccionario de la Real Academia de la Lengua Española. [En línea], acceso 28 de junio, 2013; Disponible: <http://www.rae.es/rae.html>.

Dorr, B. (1997). Large-scale acquisition of LCS based lexicons for foreign language tutoring. En: *Proceedings of the fifth conference on Applied natural language processing*, Washington, USA, pp 139-146.

Eriksson, H. (2007). The semantic-document approach to combining documents and ontologies, *Int. J. Human–Comput. Stud.* 65 (7), 624–639.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. En: Cambridge, MA: MIT Press.

Fensel, D. (2001). *Ontologies : Silver Bullet for Knowledge Management and Electronic Commerce*. En: Springer Berlin Heidelberg.

Fillmore, C. (1968). *Universals in Linguistics Theory (Vol. 1)*. En: E. B. Harms, Ed. Holt, Rinehart and Winston Publishing Company.

Gale, W., Kenneth, W., Church, Y., y David, Y. (1992b). One sense per discourse. En: Proceedings of the ARPA Workshop on Speech and Natural Language Processing, pp 233–237.

Genta, F. (2008). Perífrasis Verbales en español: Focalización aspectual, restricción temporal y rendimiento discursivo. Tesis Doctoral. En: Universidad de Granada.165-250.

Gómez, P., Corcho, O., y Fernández, L. (2004). Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and Semantic Web, Advanced Information and Knowledge Processing. En: Springer-Verlag, Berlin, Germany.

González, A., Laparra, E., y Rigau, G. (2012). Multilingual Central Repository version 3.0. En: 8th international conference on Language Resources and Evaluation (LREC'12). Istambul, Turkey. 2012. Disponible [En Línea] <http://adimen.si.ehu.es/web/MCR/> Consultado el 5 de julio de 2013.

Guzmán, J., Lezcano, L., y Gómez, A. (2013). Characterization of the elements the goal diagram KAOS from natural language. En: Revista Colombiana de Tecnologías de Avanzada Volumen 1, Número 21, pp.138-144

Grishman, R. (1991). Computational Linguistics. En: Cambridge University Press, Cambridge. Versión castellana de Antonio Moreno Sandoval: Introducción a la lingüística computacional. En: Visor Distribuciones Madrid, 1991. Colección Lingüística y Conocimiento.

Gruber, T. (1993). A Translation Approach to Portable Ontology Specifications. En: Knowledge Acquisition, 5(2) pp. 199-220.

Hope, D. (2008). Java WordNet: Similarity. Falmer: Cognitive and Language Processing Systems Group (CALPS). En: University of Sussex. Retrieved from <http://www.sussex.ac.uk/Users/drh21/>

Heaven, W. Finkelstein, A. (2004). UML profile to support requirements engineering with KAOS. En: IEEE proceedings software. Vol 151; Part 1. pp. 10-27.

Horridge, M. (2011). A Practical Guid To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools. En: The University of Manchester.

Ibrahim, M., y Ahmad, R. (2010). Class diagram extraction from textual requirements using Natural Language Processing techniques. En: Second International Conference on Computer Research and Development, pp. 200-2004. Kuala Lumpur, Malaysia: IEEE.

Jacobson, I., Booch, G., y Rumbaugh, J. (2001). El Proceso Unificado de Desarrollo de Software. En: Addison Wesley. p 438.

Jackson, P., Moulinier, I. (2002). Natural language processing for online applications : text retrieval, extraction, and categorization. En: Natural Language Processing, ISSN 1567-8202.

Jaramillo, A., Zapata, C., y Arango, F. (2005). “Una propuesta para el reconocimiento semiautomático de operaciones utilizando un enfoque lingüístico”. En: Revista Facultad de Ingeniería Universidad de Antioquia. N.o 34. pp. 42-51.

Jiang, J., y Conrath, D. (1998). ‘Semantic similarity based on corpus statistics and lexical taxonomy’. En: Proceedings of the International Conference on Research in Computational Linguistics.

Joshi, S., y Deshpande, D. (2012). Textual Requirement Analysis for UML Diagram Extraction by using NLP. En: International Journal of Computer Applications, 50(8), pp. 42-46.

Lamsweerde, A., y Letier, E. (2004). From Object Orientation to Goal Orientation: A Paradigm Shift for Requirements Engineering. En: Radical Innovations of Software and Systems Engineering in the Future, pp. 153-166. Venecia: Springer.

Lamsweerde, A. (2004). Goal-oriented requirements engineering: a roundtrip from research to practice. En: 12th Requirements Eng. Conf. 2004 (RE'04), IEEE Publishers, Kyoto, Japan.

Lamsweerde, A. (2000). Requirements Engineering in the Year 2000: A Research Perspective. En: 22nd International Conference on Software Engineering.

Lamsweerde, A., y Willemet, L. (1998). Inferring declarative requirements specifications from operational scenarios. En: IEEE Trans. Software Eng. 24 (12), 1089–1114.

Lamsweerde, A., Dardenne, A., y Fickas, S. (1991). Goal-directed Concept Acquisition in Requirements Elicitation. En: Proceedings of the 6th international workshop on Software specification and design, pp. 14-21. IEEE Computer Society Press.

Lamsweerde, A., Dardenne, A., Delcourt, B., y Dubisy, F. (1991). The KAOS Project: Knowledge Acquisition in Automated Specification of Software. En:

Proceedings AAAI Spring Symposium Series, Stanford University, American Association for Artificial Intelligence, pp. 59-63.

Lapouchnian, A. (2005). Goal-Oriented Requirements Engineering: An Overview of the Current Research. En: University of Toronto. Technical Report: <http://www.cs.toronto.edu/~alexsei/pub/Lapouchnian-Depth.pdf>.

Leacock, C., y Chodorow, M. (1998). Combining local context and WordNet similarity for word sense identification. En: WordNet Electron. Lex. Database, vol. 49, n.o 2, pp. 265–283.

Leite, J., Rossi, G., Maiorana, V., Balaguer, F., Kaplan, G., Hadad, G., y Oliveros, A. (1997). Enhancing a Requirements Baseline with Scenarios. En: Proceedings of RE 97', IEEE 3rd International Requirements Engineering Symposium, Annapolis, USA, pp. 44–53.

Leite, J., Hadad, G., Doorn, J., y Kaplan, G. (2000). A Scenario Construction Process. En: Requirements Engineering Journal Vol.5.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. En: Proceedings of the 5th annual international conference on Systems documentation, pp. 24–26.

Letier, E. (2001). Reasoning about Agents in Goal-Oriented Requirements Engineering. PhD Thesis. En: Université catholique de Louvain, Département d'Ingénierie Informatique.

Letier, E., y Lamsweerde, A. (2002). Deriving operational software specifications from system goals. En: 10th Symposium on Foundations of Software Eng. (FSE'02), ACM Press, Charleston, South Carolina, USA.

Lezcano, L., Herrera, L., y Londoño, A. (2013). Definición de un método para realizar la educación de requisitos de software a partir de la articulación del diagrama causa efecto con el diagrama de objetivos de KAOS. En: Revista Politécnica, Vol. 9, número 16 pp, 37-50.

Lezcano, L. (2007). Elaboración semiautomática del diagrama de objetivos a partir de lenguaje natural restringido. M.Sc. Tesis. En: Universidad Nacional de Colombia, Medellín.

Loucopoulos, P., y Karakostas, V. (1995). System Requirements Engineering. En: McGraw-Hill International series. En: Software Engineering, ISBN 0-07-707843-8.

Maiden, N., Manning, S., Jones, S., y Greenwood, J. (2005). Generating requirements from systems models using patterns: a case study. En: Requirements Eng. 10 (4), 276–288.

Matulevičius, R., Heymans, P., y Opdahl, A. (2007). Ontological Analysis of KAOS Using Separation of Reference. En: Proc. IGIPublishing Vol. 6, pp 37-54.

Miller, G. (1995). WordNet: a lexical database for English. En: ACM, vol. 38, número 11, pp. 39–41.

Montoyo, A. (2002). Desambiguación léxica mediante Marcas de Especificidad. PhD Tesis. En: Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante.

Moreno, B., y Molina, M. (1999): Preliminares y tendencias en el Procesamiento del Lenguaje Natural. En: Inteligencia Artificial, 7, Primavera, pp. 65-76.

Mylopoulos J., Chung L., Liao S., Wang H., y Yu, E.(2001). Exploring Alternatives During Requirements Analysis. En: IEEE Software.

Navigli, R., y Lapata, M. (2007). Graph Connectivity Measures for Unsupervised Word Sense Disambiguation. En: IJCAI, 2007, pp. 1683–1688.

Navigli, R. (2009). Word sense disambiguation: A survey. En: ACM Comput. Surv. CSUR, vol. 41, n.o 2, p. 10.

Navigli, R., y Vannella, D. (2013). SemEval-2013 task 11: Evaluating word sense induction & disambiguation within an end-user application. En: Proceedings of the 7th International Workshop on Semantic Evaluation, in conjunction with the Second Joint Conference on Lexical and Computational Semantics. Atlanta, USA.

Nicolás, J., Toval, A. (2009). On the generation of requirements specifications from software engineering models: A systematic literature review. En: Information and Software Technology. 51(9), 1291-1307.

Ng, H., y Lee, H. (1996). Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. En: Proceedings of

the 34th annual meeting on Association for Computational Linguistics, pp. 40–47.

Samper, J. (2005). Ontologías para servicios web semánticos de información de tráfico: descripción y herramientas de explotación. PhD Tesis. En: Universidad de Valencia. España.

Seco, N. (2004). *javasimlib - Java WordNet Similarity*. Dublin: University College Dublin. Retrieved from <http://fviveros.gelbukh.com/wsd.html>

Sousa, B. (2008). *Aspect-Oriented Requirements Analysis*. PhD Thesis. En: Universidade Nova de Lisboa, Faculdade de Ciência e Tecnologia, Lisboa.

Tartarus. Snowball. [En línea], acceso 05 de enero, 2013; disponible en: <http://snowball.tartarus.org/algorithms/spanish/stop.txt>.

Thomas, P., y Oliveros, A. (2004). Proceso de Elicitación de Objetivos. En: Congreso Argentino de Ciencias de la Computación CACIC2004, Universidad Nacional de La Matanza.

Thomas, P., y Oliveros, A. (2005). *Workshop em Engenharia de Requisitos*. En: Porto, Portugal, ISBN 972-752-079-0, pp. 50-62.

Oshiro, K., Watahiki K., Saeki M. (2003). Goal-Oriented Idea Generation Method for Requirements Elicitation, 11th IEEE International Requirements Engineering Conference, 2003, pp. 363-364.

Park, S., Kim, M., y Sugumaran, V. (2004). A scenario, goal and feature oriented domain analysis approach for developing software product lines. En: *Ind. Manage. Data Syst. (IMDS)* J. 104 (4), pp. 296–308.

Pfleeger, S. (2002). Ingeniería de Software, Teoría y Práctica. En: Editorial Prentice Hall. ISBN 987-9460-71-5.

Padró, L., y Stanilovsky, E. (2012). Freeling 3.1: Towards wider multilinguality. "An open source suite of language analyzers" Disponible [En Línea] <http://nlp.lsi.upc.edu/freeling/> Consultado el 4 de julio de 2013

Pressman, R. (1998). Software Engineering. En: McGraw-Hill. ISBN 84-481-1186-9. Cuarta Edición.

Respect It. A KAOS Tutorial. Objectiver (2007). [En línea], acceso 04 de julio, 2013; disponible en:
<http://www.objectiver.com/fileadmin/download/documents/KaosTutorial.pdf>

Resnik, P., y Yarowsk, D. (1999). Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation. J. Nat. Lang. Eng. 5, 2, 113–133.

Rios, M. (2012). Word Sense Disambiguation And Recognizing Textual Entailment Wint Statistical Methods. M.Sc Tesis. En: Instituto Politécnico Nacional. México.

Rigau, G. (2012). Resolución automática de la ambigüedad semántica de palabras. En: Fundación Duques de Soria. España.

Rodríguez, H. (2011). Cálculo de la visibilidad de conceptos en ontologías. M.Sc. Tesis. En: Instituto Politécnico Nacional. México.

Rumbaugh, J., Blaha, M., Premerlany, W., Eddy, F., y Lorensen W. (1991). Object Oriented Modelling and Design. En: Prentice-Hall.

Saporiti, E. (2006). Gramática elemental de la lengua española. En: ISBN 978-987-05-2221-8. Mar del Plata, Buenos Aires, Argentina.

Snasel, V., Moravec, P., y Pokorny, J. (2005). WordNet Ontology Based Model for Web Retrieval. International Workshop on Challenges. En: Web Information Retrieval and Integration, 220–225. doi:10.1109/WIRI.2005.38

UML. (2004). Object Management Group (OMG). Unified Modeling Language Specification.

Verdejo, M. (1994). Comprensión del lenguaje natural: avances, aplicaciones y tendencias. En: Procesamiento del lenguaje natural: fundamentos y aplicaciones, pp. 5-29.

Wand, Y., y Weber, R. (1995). On the Deep Structure of Information Systems. En: W. O. Library, Ed. Information Systems Journal, 5(3), pp. 203-223.

Wu, Z., y Palmer, M. (1994). Verb semantics and lexical selection. En: Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico.

Yu, E. (1995) Modelling Strategic Relationships for Process Reengineering. PhD Thesis. En: University of Toronto. Department of Computer Science.

Yu, E., Dubois, P., Dubois, E., y Mylopoulos, J. (1995). From organization models to system requirements. A cooperating agents approach, in: 3rd Intl. Conf. on Cooperative Inf. Sys. (CoopIS-95), Vienna, Austria.

Zapata, C., y Arango, F. (2005). Los modelos verbales en Lenguaje Natural y su utilización en la elaboración de esquemas conceptuales para el desarrollo

de software: Una revisión crítica. En: Revista EAFIT. Vol 41. No 137, pp 77 – 95.

Zapata, C., Gelbukh, A., y Arango, F. (2006c). Pre-conceptual Schema: A Conceptual-Graph-Like Knowledge Representation for Requirements Elicitation. En: Lecture Notes in Computer Science, Vol. 4293, pp. 17–27.

Zapata, C., y Lezcano, L. (2009). Caracterización de los verbos usados en el diagrama de objetivos. En: Revista Dyna, Nro. 158, pp. 219-228.

Zapata, C., Villegas, S., y Arango, F. (2006). Reglas de consistencia entre modelos de requisitos de UN-Método. En: Revista Universidad Eafit. Vol 42. No 141, pp. 40–59.

Zapata, C., Lezcano, L., y Tamayo, P. (2011). "Preconceptual-schema-based representation of KAOS goal diagram. En IEE: "Computing Congress (CCC) 6th Colombian", pp. 1-6.

Zapata, C., Lezcano, L., Tamayo, P. (2007). Validación del método para la obtención automática del diagrama de objetivos desde esquemas preconceptuales En: Revista Eia Vol 8. pp.221 – 235.

Zapata, C., y Vargas, F. (2011). Innovación en el diseño y evaluación de proyectos: establecimiento de las relaciones lingüísticas entre objetivos y problemas. En: Lámpsakos, Vol. 3, No. 6, pp. 46-55.

Zapata, C., Giraldo, G., y Urrego, G. (2010). Las ontologías en la ingeniería de software: un acercamiento de dos grandes áreas del conocimiento. En: Revista Ingenierías Universidad de Medellín, Vol. 9, No. 16 pp. 91-99.

Zowghi, D., y Gervasi, V. (2004). Erratum to “On the interplay between consistency, completeness, and correctness in requirements evolution”. En: Information and Software Technology, 46(11), 763-779.