



UNIVERSIDAD NACIONAL DE COLOMBIA

# **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

**Fabio Alberto Vargas Agudelo**

Universidad Nacional de Colombia  
Facultad de Minas, Departamento Ciencias de la Computación y de la Decisión  
Medellín, Colombia  
2015

# **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

**Fabio Alberto Vargas Agudelo**

Tesis de investigación presentada como requisito parcial para optar al título de:  
**Doctor en Ingeniería de sistemas**

Director (a):

Ph.D. Carlos Mario Zapata Jaramillo

Grupo de Investigación:

Lenguajes Computacionales

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

2015

## *Dedicatoria*

*A mi Mamá y a mi Papá porque, sin ellos, no hubiese logrado cumplir mis sueños y metas personales y profesionales y que me permitieron convertirme en una persona honesta y responsable. Los quiero y ojalá la vida me permita compartir con ellos muchos éxitos más. Y también a mi sobrino Gerónimo que es mi inspiración y el hijo que aún no he tenido*

*Fabio A Vargas A*

## **Agradecimientos**

Durante mi formación doctoral, son muchas las personas que han contribuido y me han apoyado para sacar adelante este proyecto profesional y de vida. También son muchas las personas que me han apoyado en la construcción de la Tesis Doctoral que presento para mi grado como Doctor en Ingeniería de Sistemas de la Universidad Nacional de Colombia. En especial agradezco a mi tutor Carlos Mario Zapata quien desde mis estudios de Maestría me ha acompañado en mi proceso de formación en el campo investigativo hasta culminar con éxito mi formación doctoral. Todo este tiempo me ha permitido conocer un gran profesional y un gran ser humano y que ha contribuido directamente con mi formación personal y profesional. Gracias por sus enseñanzas y por su apoyo incondicional.

Muchas gracias al Tecnológico de Antioquia, Institución Universitaria, por su apoyo y confianza durante mi tiempo de formación, esperando contribuir a la institución con mi dedicación y compromiso académico e investigativo.

Igualmente, quiero expresar mi gratitud a todas las personas que directa o indirectamente han apoyado estas ideas, a todos mis demás colaboradores, colegas, estudiantes y amigos y a quienes de alguna manera me ayudaron a salir adelante.

## Resumen

Uno de los desafíos de la educación de requisitos de software es garantizar que los requisitos del sistema sean consistentes, pertinentes y contextualizados con las necesidades de la organización. Para llevar a cabo esta tarea, el analista realiza el proceso de forma manual, basado en su experiencia y conocimiento. La educación de requisitos de software requiere, en sus procesos iniciales de análisis organizacional, una alta participación del analista y el interesado, para reconocer el dominio en el que se desplegará el producto de software, incluyendo objetivos de la organización, problemas del dominio y objetivos del sistema a implementar.

Muchos de los trabajos enfocados en esta fase de la ingeniería de software establecen que los objetivos organizacionales son una base importante para generar trazabilidad y consistencia entre los problemas del dominio y los requisitos del sistema. Algunas metodologías de desarrollo de software, de ingeniería de software orientada a objetivos (Goal-Oriented Requirements Engineering, GORE) y de análisis organizacional utilizan esquemas de representación para los problemas del dominio y los objetivos. Por ejemplo, el UNC-Method (Método para el Desarrollo de Aplicaciones de Software de la Universidad Nacional de Colombia) emplea una adaptación del diagrama causa-efecto para especificar los problemas del dominio y relacionarlos con los objetivos y los actores del sistema; el Business Modeling with UML define un esquema de objetivos que relaciona gráficamente objetivos y problemas; el NFR FrameWork especifica los problemas por medio de marcos de problemas en el proceso de educación de requisitos no funcionales. La metodología KAOS y los FrameWorks I\* y TROPOS definen requisitos de software a partir de objetivos organizacionales y la metodología de análisis organizacional marco lógico relaciona objetivos y problemas en la formulación de proyectos para la toma de decisiones. Todos estos trabajos carecen de representaciones formales para la especificación de objetivos organizacionales, problemas de dominio y objetivos del sistema y, por tal motivo, es difícil generar consistencia y trazabilidad en el proceso, porque los analistas suelen especificar objetivos y problemas de forma subjetiva y en muchos casos no se enuncian con información que denote realmente un problema o un objetivo.

Existe un trabajo inicial que vincula objetivos y problemas a partir de la negación de los primeros, generando una relación entre ellos, pero que sólo utiliza reglas sintácticas para relacionar términos comunes enunciados en los objetivos y los problemas.

La Tesis Doctoral que se presenta a continuación define un modelo para la especificación, formalización y relación de los objetivos organizacionales, problemas del dominio y objetivos del sistema en la fase de educación de requisitos de software, permitiendo generar trazabilidad y consistencia. Se estructura un conjunto de reglas sintácticas y semánticas que facilitan una relación de asociación de términos comunes del dominio. La Tesis se valida por medio de casos de estudio donde se aplica el modelo y se respalda con un conjunto de publicaciones generadas de la propuesta doctoral.

**Palabras clave: objetivos organizacionales, objetivos del sistema, requisitos, problemas del dominio, reglas semánticas, reglas sintácticas, trazabilidad, consistencia.**

## **Abstract**

Guaranteeing the systems requirements are consistent, relevant, and in the context of the organizational needs is one of the most important challenges in software requirements elicitation. With this aim in mind, analysts—based on their experience and knowledge—manually execute the process. The early stages of software requirements elicitation requires high involvement of the analyst and the stakeholder for recognizing the domain in which the software product will be displayed. Such a domain comprises the organizations goals, the goals of the system to be implemented, and the domain problems.

Some research has been focused on this stage of software engineering. According to this, organizational goals are important foundations for the generation of traceability and consistency among domain problems and systems requirements. Some methodologies for software development—Goal-Oriented Requirements Engineering (GORE) and organizational analysis—use representation schemes for domain problems and goals. For example, the UNC-Method (software application development method of the Universidad Nacional de Colombia) employs an adaptation of cause-effect diagrams for specifying domain problems and linking them to the systems goals and actors. Business Modelling

with UML creates a diagram of goals for graphically connecting goals and problems. NFR FrameWork specifies problems by means of problem frameworks in the non-functional requirements elicitation process. KAOS methodology, I\*, and TROPOS define software requirements—system goals—based on organizational goals, whereas the logical framework methodology for organizational analysis relates goals to problems in the creation of projects for making decisions. All of this research lacks formal representations to specify goals—both organizational and system goals—and domain problems. Therefore, traceability and consistency are difficult to evaluate in the process. Analysts tend to specify goals and problems in a subjective way, so problems and goals are unrecognized.

An initial study links goals and problems by denying the former and generating a relationship between them. However, only syntactical rules to relate common terms that appear in the goals and problems are used. Thus, low traceability or consistency can be established among organizational goals, domain problems and system goals.

In this doctoral dissertation, we define a model to specify, formalize and relate domain problems, and organizational and system goals in the early stages of software requirements elicitation. Thus, we can evaluate traceability and consistency among them. A set of semantic and syntactic rules is generated for linking common terms belonging to the domain. The dissertation is validated by using case studies in which the model is applied. Also, some publications based on the dissertation proposal are used to validate the model.

**Keywords: organizational goals, system goals, requirements, domain problems, semantic rules, syntactic rules, traceability, consistency.**

# Contenido

	Pág.
Resumen.....	V
Lista de figuras .....	X
Lista de tablas .....	XI
Lista abreviaturas .....	XII
Introducción .....	1
<b>1. MARCO CONCEPTUAL DE LA PROBLEMÁTICA.....</b>	<b>5</b>
1.1 REQUISITO DE SOFTWARE.....	5
1.1.1 Educación de requisitos de software.....	5
1.1.2 Trazabilidad de requisitos de software .....	6
1.1.3 Consistencia .....	6
1.1.4 Definición de Objetivo .....	7
1.1.5 Definición de objetivo organizacional .....	7
1.1.6 Definición de objetivo del sistema .....	8
1.2 PROBLEMAS DEL DOMINIO.....	8
1.3 TÉCNICAS DE GORE ( <i>GOAL-ORIENTED REQUIREMENTS</i> <i>ENGINEERING</i> ).....	9
1.3.1 KAOS .....	9
1.3.2 I*.....	10
1.3.3 TROPOS .....	10
1.3.4 NFR FRAMEWORK.....	10
1.3.5 GBRAM .....	11
1.4 REGLAS SINTÁCTICAS .....	11
1.5 REGLAS SEMÁNTICAS.....	11
1.6 ESQUEMAS PRECONCEPTUALES.....	12
1.7 MÉTRICAS.....	13
<b>2. REVISIÓN DE LA LITERATURA .....</b>	<b>14</b>
<b>3. PLANTEAMIENTO DEL PROBLEMA E HIPÓTESIS.....</b>	<b>24</b>
3.1 PROBLEMA DE INVESTIGACIÓN.....	24
3.1.1 Descripción del problema .....	24
3.1.2 Pregunta de Investigación .....	27
3.2 HIPÓTESIS DE INVESTIGACIÓN.....	28
3.3 OBJETIVOS .....	28
3.3.1 Objetivo general.....	28



---

3.3.2	Objetivos específicos .....	28
3.4	CONTRIBUCIÓN .....	29
<b>4.</b>	<b>SOLUCION .....</b>	<b>31</b>
4.1	ESTRUCTURA SINTÁCTICO-SEMÁNTICA PARA REPRESENTAR Y FORMALIZAR OBJETIVOS EN EL CONTEXTO DE LA EDUCCIÓN TEMPRANA DE REQUISITOS DE SOFTWARE .....	31
4.2	ESTRUCTURA SINTÁCTICO-SEMÁNTICA PARA REPRESENTAR Y FORMALIZAR PROBLEMAS EN EL CONTEXTO DE LA EDUCCIÓN TEMPRANA DE REQUISITOS DE SOFTWARE .....	37
4.3	MODELO PROPUESTO .....	40
4.3.1	Especificación de Objetivos organizacionales .....	41
4.3.2	Representación del dominio .....	42
4.3.3	Especificación de problemas.....	43
4.3.4	Representación de dominio con problemas.....	53
4.3.5	Especificación y formalización de objetivos del sistema (requisitos iniciales)53	
<b>5.</b>	<b>METRICAS DEL MODELO .....</b>	<b>59</b>
<b>6.</b>	<b>CASO DE ESTUDIO .....</b>	<b>64</b>
6.1	CASO DE ESTUDIO NRO. 1 .....	64
6.2	CASO DE ESTUDIO NRO. 2 .....	84
6.3	CASO DE ESTUDIO NRO. 3 .....	88
6.4	PRODUCTOS DE NUEVO CONOCIMIENTO GENERADOS A PARTIR DE LA TESIS DOCTORAL .....	89
<b>7.</b>	<b>CONCLUSIONES y TRABAJO FUTURO .....</b>	<b>97</b>
7.1	Conclusiones .....	97
7.2	Trabajo Futuro .....	99
<b>A.</b>	<b>Anexo: Certificado de Caso de Estudio.....</b>	<b>102</b>
	<b>Referencias Bibliográficas.....</b>	<b>103</b>

## Lista de figuras

	<b>Pág.</b>
Figura 1. Compendio gráfico de términos de la Tesis Doctoral.....	14
Figura 2. Árbol de Objetivos con inconsistencias.....	29
Figura 3. Árbol de objetivos de Marco Lógico con inconsistencias.....	34
Figura 4. Árbol de problemas de Marco Lógico con inconsistencias.....	34
Figura 5. Representación gráfica de problemas (1).....	40
Figura 6. Representación gráfica de problemas (2).....	41
Figura 7. Esquema general para estructurar objetivos.....	47
Figura 8. Esquema general para la representación de problemas del dominio.....	52
Figura 9. Visión general del modelo.....	56
Figura 10. Diagrama de Objetivos definido por el interesado.....	57
Figura 11. Diagrama de Objetivos después de aplicar reglas.....	57
Figura 12. Representación del dominio con objetivos.....	58
Figura 13. Formalización de un problema del dominio.....	67
Figura 14. Representación del dominio con vinculación de los problemas.....	68
Figura 15. Ejemplo de relación semántica (1).....	75
Figura 16. Ejemplo de relación semántica (2).....	75
Figura 17. Diagrama de objetivos con el interesado (caso de estudio 1).....	82
Figura 18. Representación del dominio con objetivos (Caso de estudio 1).....	83
Figura 19. Representación del dominio con los problemas encontrados parte 1 (caso de estudio 1).....	87
Figura 20. Representación del dominio con los problemas encontrados parte 2 (caso de estudio 1).....	87
Figura 21. Diagrama de objetivos (caso de estudio 2), parte 1.....	101
Figura 22. Diagrama de objetivos (caso de estudio 2), parte 2.....	102
Figura 23. Representación del dominio con objetivos (caso de estudio 2).....	103

## Lista de tablas

	<b>Pág.</b>
Tabla 1. Definición de Elementos de los esquemas preconceptuales.....	26
Tabla 2. Estructura gramatical para representar problemas.....	32
Tabla 3. Estructura gramatical para representar objetivos.....	32
Tabla 4. Estructura gramatical para relacionar objetivos y problemas.....	33
Tabla 5. Compendio de revisión de literatura.....	36
Tabla 6. Estructuras para la especificación y formalización de objetivos.....	48
Tabla 7. Estructuras para la especificación y formalización de problemas.....	53
Tabla 8. Proceso para la especificación y formalización de objetivos.....	56
Tabla 9. Proceso para la representación del dominio con objetivos.....	58
Tabla 10. Proceso para la especificación de problemas del dominio.....	59
Tabla 11. Reglas para convertir objetivos en problemas del dominio.....	60
Tabla 12. Proceso para la representación del dominio con problemas.....	68
Tabla 13. Proceso para la especificación de objetivos del sistema.....	69
Tabla 14. Reglas para vincular problemas con objetivos del sistema.....	70
Tabla 15. Métricas base.....	76
Tabla 16. Métricas derivadas.....	77
Tabla 17. Especificación de objetivos (caso de estudio 1).....	80
Tabla 18. Especificación y formalización de problemas (caso de estudio 1).....	84
Tabla 19. Especificación de objetivos del sistema (caso de estudio 1).....	88
Tabla 20. Aplicación de las métricas base (caso de estudio 1).....	89
Tabla 21. Aplicación de las métricas derivadas (caso de estudio 1).....	94
Tabla 22. Especificación de Objetivos (caso de estudio 2).....	100
Tabla 23. Especificación de problemas (caso de estudio 2).....	103
Tabla 24. Productos de nuevo conocimiento generados.....	106

## **Lista abreviaturas**

GORE: Goal-Oriented Requirements Engineering

KAOS: Knowledge Acquisition in Automated Specification of Software

MOQARE: Misuse-oriented Quality Requirements Engineering

UNC-Method: Problem-based software development method

NFR: Non-functional requirement

UML: Unified Modeling Language

KAOS: Knowledge Acquisition in Automated Specification of Software



solucionar con un producto de software surge a partir del conocimiento de la organización, de la participación activa del interesado y el analista y de la buena comunicación y entendimiento entre ellos. Todo este proceso lo realiza de forma manual el analista, con base en su experiencia.

Martínez, Estrada y Gama (2009) y Chaves (2011) reconocen la importancia de las técnicas de ingeniería de requisitos de software en sus fases iniciales, conocidas como técnicas de modelado organizacional, ya que es la etapa fundamental para la construcción de software de alta calidad que dé respuesta a las necesidades y expectativas de los interesados y que tenga muy en cuenta los objetivos de la organización, garantizando una mayor trazabilidad y consistencia. Algunas metodologías y FrameWorks como KAOS (Knowledge Acquisition in Automated Specification of Software; Dardenne, Lamsweerde & Fickas, 1993), I\* (Yu, 1995) y TROPOS (Castro, Kolp & Mylopoulos, 2002) emplean esquemas para representar objetivos organizacionales como un insumo fundamental para especificar los requisitos de la aplicación de software a desarrollar.

Otros trabajos permiten visualizar una relación entre objetivos y problemas como medio para mejorar la trazabilidad y la consistencia en la especificación de requisitos de software. UNC-Method (Zapata & Arango, 2009), emplea el diagrama causa-efecto (Ishikawa, 1986) para especificar los problemas y relacionarlos con los objetivos y los actores del sistema; en Business Modeling with UML (Eriksson & Penker, 1999) se define un esquema de objetivos que relaciona gráficamente objetivos y problemas; en NFR FrameWork (Chung, Nixon, Yu & Mylopoulos, 1995) se especifican los problemas empleando marcos de problemas en el proceso de educación de requisitos no funcionales; Engelsman y Wieringa (2014) proponen un lenguaje gráfico para el modelado de objetivos organizacionales, como una herramienta para relacionar la organización con los procesos de tecnología de información que se adelanten al interior de la misma.

Todos estos trabajos conservan algunos problemas en el proceso de educación de requisitos: poca vinculación entre los requisitos iniciales y los requisitos tardíos (Martínez, Pastor, Mylopoulos & Giorgini, 2006); los requisitos no satisfacen por lo menos uno de los objetivos organizacionales (Ali, Dalpiaz & Giorgini, 2010); las aplicaciones de software, en muchos casos, no son la correcta representación de los requisitos capturados desde el modelo del negocio (Estrada, Martínez, Pastor, & Sánchez, 2002); no hay una relación entre la funcionalidad esperada de un aplicativo de software y los procesos de negocio a los que éste dará soporte (Ali, Dalpiaz & Giorgini, 2010); la trazabilidad y consistencia entre las expectativas y necesidades y su representación en un diagrama de objetivos es débil

(Burnay, Jureta, Linden & Faulkner, 2014); no se utilizan los objetivos de la organización para determinar la completitud y suficiencia de la especificación de requisitos (Herrmann & Paech, 2008); no se emplea en muchos casos un modelo de organización inicial, que guía el analista en la obtención de la información relevante desde el contexto de la organización (Horkoff, Barone, Jiang, Yu, Amyot, Borgida & Mylopoulos, 2014); los problemas se redactan mal y esto dificulta su relación y trazabilidad (Vargas, 2010); se carece de métodos formales para el planteamiento de objetivos y problemas del dominio (Zapata, Acevedo & Moreno, 2011); es difícil automatizar y validar el proceso porque los analistas suelen construir el diagrama de objetivos y los diagramas de problemas de forma subjetiva (Zapata & Lezcano, 2009); la determinación de objetivos se basa en la experiencia del analista y en el conocimiento del interesado, pero sin validar la consistencia que tenga en cuenta los problemas del dominio (Liu & Jin, 2007).

Vargas (2010) plantea un trabajo inicial basado en un conjunto de reglas gramaticales para enunciar objetivos y problemas, utilizando como referencia el diagrama de objetivos de KAOS y el diagrama causa-efecto. Se establece una relación de consistencia basada en dichas reglas, pero empleando estructuras sintácticas que requieren el uso explícito de algunas palabras comunes entre el objetivo y el problema. En este caso, no se tienen en cuenta relaciones semánticas que determinen los objetivos que se ligan con ciertos problemas en el dominio sobre el cual se trabaja, generando aun poca trazabilidad y consistencia.

En esta Tesis Doctoral se define un modelo para representación y formalización directa de los problemas y los objetivos durante la fase de educación temprana de requisitos de software, que permita generar una mayor trazabilidad en la relación entre objetivos organizacionales, problemas del dominio y requisitos (objetivos del sistema). Esto se logra con la definición de un conjunto de reglas sintácticas y semánticas que permiten una relación de asociación de términos comunes del dominio sobre el cual se trabaja para lograr con ello una especificación de requisitos contextualizados y pertinentes.

La Tesis se organiza de la siguiente manera: en el Capítulo 1 se describen conceptos importantes que se emplean para dar claridad del área del conocimiento en el cual se centra el trabajo; en el Capítulo 2 se presenta una revisión de literatura que muestra los procesos de educación de requisitos que utilizan objetivos y problemas en su especificación; en el Capítulo 3 se realiza el planteamiento del problema, la hipótesis, los objetivos y la contribución; en el Capítulo 4 la solución; en el Capítulo 5 la definición de las métricas

propuestas; en el Capítulo 6 los casos de estudio y los productos de nuevo conocimiento generados y, por último, las conclusiones y el trabajo futuro.



# **1.MARCO CONCEPTUAL DE LA PROBLEMATICA**

## **1.1 REQUISITO DE SOFTWARE**

Los requisitos de software constituyen un insumo fundamental de la ingeniería de software. Entre sus principales definiciones se tienen (IEEE, 1990):

- Una necesidad de un interesado para dar solución a un problema o alcanzar un objetivo.
- Una característica que debe estar presente en un sistema de software para responder a un contrato, estándar, especificación u otro documento formal.
- Un objetivo que el sistema de software debe cumplir.

Los requisitos de software se dividen en dos categorías (Grillo & La Rosa, 2009):

- Funcionales: Son los que definen las funciones u objetivos que el sistema será capaz de realizar y describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el qué y no el cómo se deben hacer esas transformaciones.
- No funcionales: Son las características que, de una u otra forma, limitan el sistema. Algunos de ellos son: rendimiento (en tiempo y espacio), fiabilidad (robustez del sistema y disponibilidad de equipo), mantenimiento, seguridad y portabilidad.

### **1.1.1 Educción de requisitos de software**

Se enmarca en el proceso de levantamiento y especificación de requisitos de software que ejecuta el analista de sistemas en compañía del interesado. Es una actividad que depende directamente de la capacidad y conocimiento del analista. En esta fase se identifican y analizan aspectos relevantes del negocio y del dominio sobre el que se piensa desarrollar el aplicativo de software. Este proceso permite reconocer los objetivos organizacionales, los problemas del dominio, los requisitos funcionales y no funcionales y los actores que

intervienen en los procesos de la organización, entre otros. La educación de requisitos de software utiliza métodos, técnicas y herramientas para descubrir, documentar y administrar los requisitos del software, de forma sistemática y repetible (Adriano, 2006).

Durante la educación de requisitos se pretende una especificación completa, consistente y no ambigua, la cual sirve como base para lograr acuerdos comunes entre todas las partes involucradas y se describen las funciones que realizará el sistema (Adriano, 2006). Es la fase en la cual se intercambian puntos de vista para recopilar y especificar lo que el sistema va a realizar. El entregable es un documento de requisitos de software (Leite & Freeman, 1991).

### **1.1.2 Trazabilidad de requisitos de software**

Es un proceso de revisión que busca un producto en el dominio de la solución lo más exacto y fiable a las necesidades del interesado. Una aplicación de software es flexible y cambia con el tiempo y es aquí donde la trazabilidad adquiere gran importancia (Tabares, Moreira, Anaya, Arango & Araujo, 2007).

La trazabilidad se refiere a la “capacidad para describir y seguir la vida de un requisito en ambas direcciones, hacia delante (forward) y hacia atrás (backward), esto es, desde su origen, durante su desarrollo y especificación, hasta su desarrollo y uso y a lo largo de todos los períodos de refinamiento en curso e iteración en alguna de estas fases” (Lindvall & Sandahl, 1996).

La práctica de la trazabilidad es una actividad necesaria e inseparable durante el proceso de desarrollo (Tabares, Moreira, Anaya, Arango & Araujo, 2007).

### **1.1.3 Consistencia**

Es el estado coherente en la información o datos que contiene y que relaciona, en el cual la información cumple las necesidades o expectativas de quien la requiera. Zowghi y Gervasi (2002) enuncian algunas razones de la importancia de la consistencia en los procesos de educación de requisitos de software: la evolución de los requisitos de los interesados, que hace que los diferentes artefactos referentes a la conceptualización de

una solución, deben permanecer correctos para poder soportar los múltiples cambios que plantean los interesados, y el hecho de que diferentes personas, introduciendo sus propias apreciaciones en el proceso de desarrollo de software, pueden levantar los requisitos, generando diferentes puntos de vista del mismo problema.

### 1.1.4 Definición de Objetivo

Un propósito o meta que se propone cumplir en un lapso definido (Navarro, Valero-García, Sanchez & Tubella, 2000). Cuando se habla de los procesos de educación de requisitos de software, se plantean dos clases de objetivos: organizacionales y del sistema.

### 1.1.5 Definición de objetivo organizacional

Situación deseada que la organización intenta lograr, como una guía para la ejecución de sus acciones y la toma de decisiones. Los objetivos justifican las actividades de una empresa y la ejecución de los procesos (Martínez, Muñoz, Monserrat, Muñoz & Serafín, 2014).

Los objetivos organizacionales sirven para evaluar las acciones y la eficacia de la organización. Son muy importantes para una organización teniendo en cuenta lo siguiente (Martínez, Muñoz, Monserrat, Muñoz & Serafín, 2014):

**Guían la toma de decisiones:** una parte importante en la responsabilidad de los gerentes es tomar decisiones que influyen en la operación diaria y en la existencia de la organización y del personal de la misma. Una vez que los gerentes formulan los objetivos organizacionales, saben en qué dirección debe apuntar la operación. Su responsabilidad se convierte en la toma de decisiones que lleven a la empresa hacia el logro de sus objetivos.

**Guían la eficiencia de la organización:** dado que la ineficiencia se convierte en un costoso desperdicio del esfuerzo humano y de los recursos, los gerentes luchan por aumentar la eficiencia de la organización cuando sea posible. La eficiencia se define en términos de la calidad total del esfuerzo humano y de recursos que una empresa invierte para alcanzar sus objetivos. Por lo tanto, antes de que puedan mejorar la eficiencia de una empresa, los gerentes deben lograr una clara comprensión de los objetivos

organizacionales. Sólo entonces los gerentes podrán utilizar los recursos limitados a su disposición tan eficientemente como les es posible.

**Guían la coherencia de una organización:** el personal de una organización necesita una orientación relacionada con su trabajo. Los objetivos de la empresa se usan como base para la actividad productiva, la toma de decisiones de calidad y la planeación efectiva.

### **1.1.6 Definición de objetivo del sistema**

Propósito y fin para lo que se diseña, desarrolla, construye o piensa un sistema. Estos objetivos constituyen el fundamento del sistema y, usualmente, se definen como las metas que el sistema y su entorno deben cumplir. Durante el proceso de educación de requisitos de software se definen los objetivos del sistema, los cuales guían el desarrollo del aplicativo de software para la organización (Alrajeh, Russo, Lockerbie, Maiden, Mavin & Novak, 2013).

## **1.2 PROBLEMAS DEL DOMINIO**

Un problema es un determinado asunto o cuestión que requiere una solución en un dominio específico. A nivel social, se trata de alguna situación en concreto que, en el momento en que se logra solucionar, aporta beneficios a la sociedad. Perales (1993) establece que “un problema, de forma genérica, es cualquier situación prevista o espontánea que produce, por un lado, un cierto grado de incertidumbre y, por el otro, una conducta tendiente a la búsqueda de su solución”.

Según Rupérez (1989) y Rittel y Webber (1973), los problemas, de acuerdo con su naturaleza, se clasifican en problemas cerrados (que son aquellos que contienen toda la información precisa y se resuelven empleando un cierto algoritmo) y problemas abiertos (que, por el contrario, implican la existencia de una o varias respuestas en su solución y cuyo encargado aporta mediante una acción de pensamiento productivo).

## **1.3 TÉCNICAS DE GORE (GOAL-ORIENTED REQUIREMENTS ENGINEERING)**

Son técnicas que promueven la utilización de objetivos como base para educir, elaborar, estructurar, especificar, analizar, negociar, documentar y modificar los requisitos de software (Van Lamsweerde, 2004). Se incorpora, de esta forma, un punto de vista intencional, es decir, el propósito del sistema. La introducción de este punto de vista permite que los interesados expresen sus necesidades de una manera más natural, centrándose en lo que quieren (sus objetivos) frente a la manera de alcanzarlos (los requisitos convencionales). A partir de los objetivos, los requisitos se pueden derivar como maneras de alcanzar esos objetivos. (González-Baixauli, Laguna & Prado Leite, 2004).

Existen diversos enfoques dentro de la Goal-Oriented Requirements Engineering que se diferencian básicamente en dos factores: la actividad de ingeniería de requisitos en la que se enfocan (edución, modelado o análisis de requisitos) y el grado de formalismo que soportan.

### **1.3.1 KAOS**

Plantea la representación de un árbol de objetivos, el cual se enfoca en realizar un proceso de análisis formal de los requisitos. El proceso para el trazado del diagrama de objetivos de KAOS (Dardenne, Lamsweerde & Fickas, 1993) requiere que se definan los objetivos secundarios que subrogan los objetivos generales, para luego presentarlos en objetivos más elementales que los subrogan y así sucesivamente hasta llegar a objetivos que se consideren elementales o atómicos o hasta que se alcancen expectativas, requisitos o propiedades del dominio. Con el diagrama de objetivos se busca señalar cuáles de los objetivos subrogados satisfacen actualmente los procesos del área. Se debe notar que la incapacidad de dar satisfacción a los objetivos generales se asocia con la incapacidad de dar satisfacción a los objetivos subrogados. El diagrama de objetivos se realiza para el área problemática y para la organización que la rodea, porque se requiere que la aplicación de software del área se contextualice y se pueda determinar su influencia en la organización, evaluando la viabilidad de cumplimiento de los objetivos subrogados pertenecientes a esa área.

### **1.3.2 I\***

Yu (1995) plantea que I\* es un framework orientado a objetivos que incluye nodos que representan actores, objetivos, tareas y recursos, además de un conjunto de relaciones entre ellos. Es un enfoque que utiliza la idea de softgoal. La principal particularidad del modelado de negocio sobre otros campos de la ingeniería de requisitos es la importancia de los agentes. Un agente se define como una entidad que existe en la organización, que tiene objetivos y que puede realizar tareas o utilizar recursos para alcanzar dichos objetivos o ayudar a otros agentes a alcanzar sus objetivos.

### **1.3.3 TROPOS**

Castro, Kolp y Mylopoulos (2002) plantean TROPOS como un framework que se deriva de I\*, muy utilizado en los procesos de educación temprana de requisitos de software. Esta metodología permite capturar el qué, el cómo y el porqué del desarrollo del software en la organización. Esta metodología, además, realiza una descripción detallada de las dependencias del sistema a desarrollar y logra una adecuada especificación de requisitos funcionales y no funcionales. TROPOS utiliza dos diagramas para la representación del ambiente organizacional, vitales en la educación temprana de requisitos que plantea la metodología: el diagrama de actores, el cual describe los actores, sus objetivos y las dependencias entre ellos, y el diagrama de objetivos, el cual detalla la relación entre los objetivos y los actores.

### **1.3.4 NFR FRAMEWORK**

Según Chung, Nixon, Yu y Mylopoulos (1995) existe un acoplamiento bidireccional entre objetivos y escenarios que otorga mayor importancia a los requisitos no funcionales (NFR), basados en softgoals, que a los basados en hardgoals. Utiliza una aproximación a la representación sintáctica de objetivos, en la cual por cada objetivo se genera un escenario para identificar nuevos objetivos. El NFR realiza una representación de objetivos, los cuales se descomponen en una jerarquía AND/OR de objetivos menos abstractos.

### **1.3.5 GBRAM**

Es una metodología basada en objetivos que permite identificar, refinar y organizar objetivos, para luego transformarlos en requisitos (Antón & Potts, 1998). Dicha metodología busca atacar las dificultades que se presentan en el proceso de educación de requisitos de software y pretende generar una alta comprensión de parte de los interesados. GBRAM busca que los requisitos del software tengan una verdadera justificación y pertinencia respecto de los niveles organizacionales.

El enfoque de Antón y Potts (1998) propone atacar los problemas que se presentan en el proceso de educación y especificación de requisitos. Para resolver estos problemas, se propone que los requisitos se desarrollen mediante un modelo que los interesados comprendan fácilmente. Provee mecanismos de representación adecuados para la comprensión de los interesados, facilita la comunicación de los interesados con los analistas, empleando un lenguaje entendible, y produce requisitos relativamente fáciles de validar. Como metodología basada en objetivos, se concentra en establecer los fundamentos que justifican los requisitos de software. El propósito de GBRAM es desarrollar, validar y afirmar un método basado en metas u objetivos, otorgando soporte procedural para la identificación, elaboración, refinamiento y organización de objetivos, en la especificación de sistemas de información basados en software.

## **1.4 REGLAS SINTÁCTICAS**

Manera de construir las oraciones de forma adecuada, incluyendo la combinación u orden de los vocablos. Consiste en determinar las funciones de las palabras o grupos de palabras dentro de la oración (Van Dijk, 2005). Las reglas sintácticas permiten reconocer si una cadena o serie de símbolos es correcta gramaticalmente.

## **1.5 REGLAS SEMÁNTICAS**

Se ocupan del significado de las palabras dentro de una oración, texto o enunciado. La semántica consiste en asignar significados a las estructuras generadas, es decir se

establecen correspondencias entre las estructuras sintácticas y cada palabra dentro de un dominio (Van Dijk, 2005).

## 1.6 ESQUEMAS PRECONCEPTUALES

Son esquemas de representación que utilizan una notación gráfica para la expresión de los diferentes elementos del discurso de un interesado y constituyen un paso intermedio en la obtención automática de los diagramas de UML a partir de un lenguaje controlado (Zapata, 2007).

En la Tabla 1 se definen los elementos de los esquemas preconceptuales (Zapata, 2007)

Tabla 1. Definición de elementos de los esquemas preconceptuales

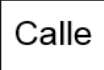


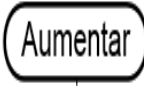
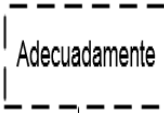




Elemento	Definición	Ejemplo
Concepto	Es un sustantivo o frase nominal del discurso del interesado.	
Relación estructural	Se asocia con los verbos “ser” y “tener”. Es una relación permanente entre conceptos.	
Relación Dinámica	Se asocia con los denominados “verbos de actividad”, que generan relaciones de tipo temporal entre los conceptos.	
Relación de logro	Representa un verbo de realización, mantenimiento o mejoramiento.	
Nota	Cuando se liga con relaciones dinámicas, representa los adverbios que se ligan con los verbos. Cuando se liga con conceptos, se pueden considerar adjetivos que se describen en los problemas y objetivos.	



Tabla 1. Definición de elementos de los esquemas preconceptuales (Continuación)

Negación (elemento nuevo que se propone en esta Tesis)	Se puede ligar con cualquiera de los elementos del esquema preconceptual para negar su contenido.	
Conexión	Es la forma en que se unen los conceptos con las relaciones y las relaciones con los conceptos.	
Unión	Es la forma de relacionar las notas y las relaciones de logro con los conceptos, relaciones estructurales y relaciones dinámicas.	
Restricción	Representa una restricción que el dominio puede tener en alguno de sus procesos.	

Para el desarrollo que se propone en esta Tesis, la negación, que se describe en la tabla anterior, es uno de los elementos nuevos que se propone para los esquemas preconceptuales.

## 1.7 MÉTRICAS

Apoyan un conjunto de técnicas basadas en medidas de procesos de ingeniería de software (aplicables a todo el ciclo de vida), para producir información de gestión significativa y a tiempo. Esta información se utiliza para mejorar esos procesos y los productos que se obtienen de ellos. Las métricas se pueden utilizar para medir aspectos como productividad, calidad, complejidad, portabilidad, trazabilidad, consistencia, etc. (Fenton & Bieman, 2014).

## 2. REVISIÓN DE LA LITERATURA

Algunas metodologías y frameworks de ingeniería de requisitos orientadas a objetivos como I\* (Yu, 1995), KAOS (Dardenne, Lamsweerde & Fickas, 1993) y TROPOS (Castro, Kolp & Mylopoulos, 2002), se centran en los objetivos organizacionales y los requisitos, pero no realizan una especificación directa de los problemas que permita una trazabilidad entre objetivos organizacionales, problemas del dominio y objetivos del sistema.

El framework I\* (Yu, 1995) tiene una alta dependencia del analista en la elaboración de los diagramas y no cuenta con una estructura sintáctico-semántica que ayude a estructurar adecuadamente los objetivos, para su fácil relación con otros componentes del sistema.

En KAOS (Dardenne, Lamsweerde & Fickas, 1993) el analista construye de manera subjetiva el diagrama, garantizando manualmente la consistencia entre sus elementos. Además, no se plantean estructuras claras para definir objetivos y, en algunos casos, los objetivos dan mayor cuenta de operaciones y no realmente de objetivos. Zapata y Vargas (2009) anotan que la estructura y definición de objetivos y requisitos dentro del diagrama de objetivos de KAOS es una tarea del analista, quien lo define de acuerdo con la interpretación y el conocimiento del área y de la organización. En la metodología KAOS, no se especifica de forma directa la representación de los problemas del dominio. En la Figura 2 se muestra un ejemplo de un diagrama de objetivos, el cual refleja que algunos de los objetivos no se redactan de forma apropiada y no son claros en la forma de expresar la idea que se pretende.

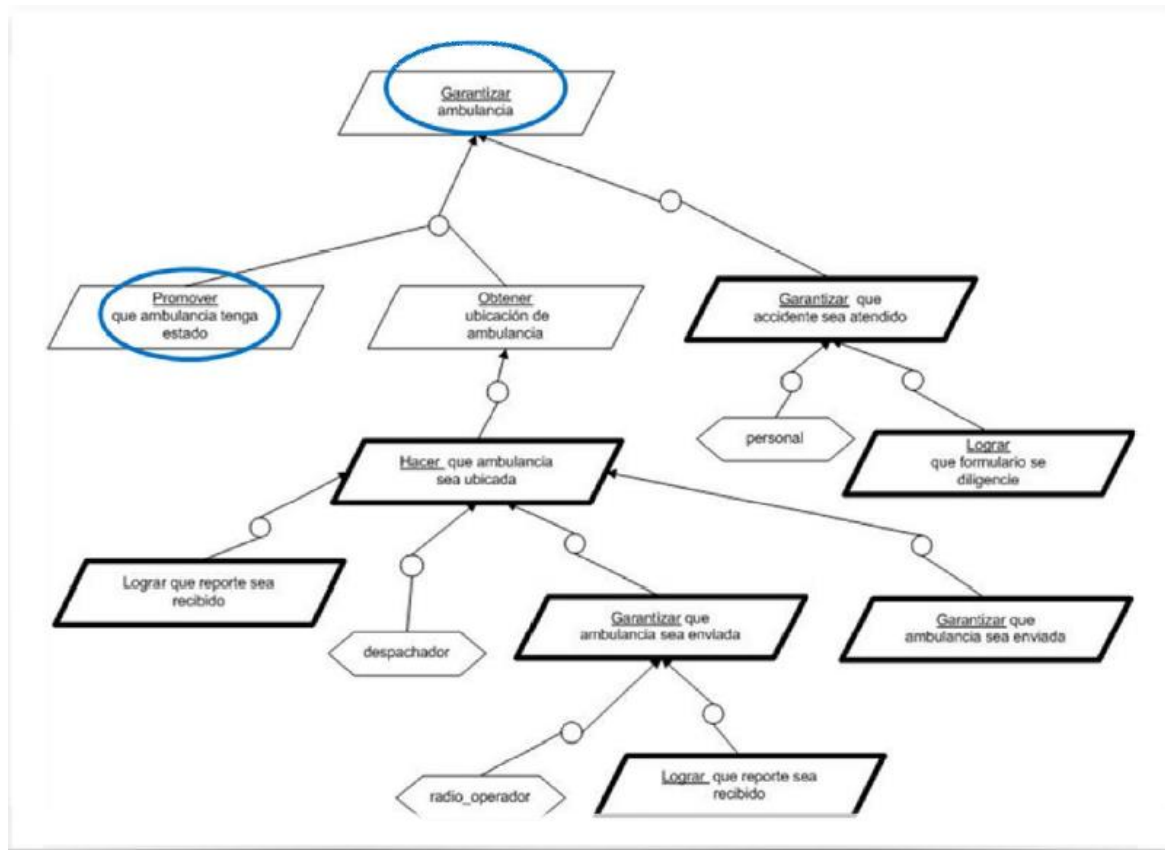


Figura 2. Árbol de Objetivos con inconsistencias (Adaptación Zapata, Lezcano & Tamayo, 2007)

Morandini, Dalpiaz, Nguyen & Siena (2014) plantean que en TROPOS no existe una representación de problemas del dominio que permita visualizar de forma directa la trazabilidad entre objetivos organizacionales, problemas del dominio y requisitos.

UNC-Method (Zapata & Arango, 2009) combina artefactos tradicionales del desarrollo de software (como los diagramas de UML y las interfaces gráficas de usuario) con enfoques no tradicionales en dicha disciplina, como los diagramas causa-efecto (Ishikawa, 1986), los diagramas de objetivos de KAOS y los esquemas preconceptuales (Zapata, 2007). La verificación de consistencia para determinar objetivos a partir del diagrama causa-efecto, es un proceso manual que lleva a cabo el analista, recurriendo a su experiencia y al conocimiento de la organización, que refleja, en algunos casos, objetivos poco claros y con problemas estructurales que no se ligan directamente con los problemas a solucionar.

Eriksson y Penker (1999) estructuran un goal/problem model que relaciona los objetivos de la organización con los problemas encontrados, para dar solución a dichos problemas. El

goal/problem model utiliza el diagrama de objetos de UML. No se especifican estructuras sintácticas ni semánticas para representar objetivos ni problemas. El analista realiza el modelo con base en su experiencia y conocimiento.

En NFR framework (Chung, Nixon, Yu & Mylopoulos, 1995), todo el proceso lo realiza el analista de forma manual. No existe una representación de problemas del dominio que permita generar una relación de trazabilidad y consistencia con los objetivos. No se plantean estructuras sintácticas ni semánticas para representar objetivos y tampoco problemas.

Horkoff, Barone, Jiang, Yu, Amyot, Borgida & Mylopoulos (2014) especifican la poca vinculación que existe entre las necesidades iniciales de los interesados y los objetivos que debe cumplir el sistema de software. Plantean la importancia de los modelos de objetivos en la validación y pertinencia final del producto.

Ali, Dalpiaz y Giorgini (2010) plantean que los requisitos de software, en muchos casos, no satisfacen por lo menos uno de los objetivos organizacionales, impidiendo con ello determinar la pertinencia de la aplicación de software que se implementa.

Burnay, Jureta, Linden y Faulkner (2014) plantean que la trazabilidad entre las expectativas y necesidades que recopilan el analista y el interesado durante el proceso de educación de requisitos de software y su representación en un diagrama de objetivos es débil y no representa en muchos casos la información recopilada.

Estrada, Martínez, Pastor y Sánchez (2002) plantean la generación de especificaciones de requisitos de software a partir de modelos de negocio, utilizando para ello objetivos organizacionales y teniendo en cuenta que las aplicaciones de software, en muchos casos, no son la correcta representación de los requisitos capturados desde el modelo del negocio.

Herrmann y Paech (2008), en su trabajo MOQARE (Misuse-oriented Quality Requirements Engineering), plantean que los objetivos organizacionales no se tienen en cuenta para determinar completitud y suficiencia en la especificación de requisitos de software.

Zapata, Acevedo y Moreno (2011) realizan una representación de relaciones semánticas entre problemas y objetivos mediante lógica de predicados como una aproximación inicial a la generación de métodos formales para el planteamiento de objetivos y problemas del dominio. El trabajo todavía se queda corto en la relación entre objetivos y problemas que no tengan palabras comunes en sus especificaciones y en la representación de los mismos.

Estrada, Martínez, Santillán y Pérez (2014) plantean la carencia de mecanismos de representación de objetivos que permita identificar la forma como se relacionan con los actores y servicios del negocio.

Liu y Jin (2007) establecen una sinergia entre en el enfoque de análisis basado en objetivos y el enfoque de análisis de problemas, procurando tratamiento integral para el proceso de obtención de los requisitos de software. No se plantean estructuras para la especificación de objetivos ni de problemas.

Para Choi, Park y Sugumaran (2012) los requisitos en lenguaje natural se expresan mediante objetivos y escenarios, utilizando una estructura semántica para representarlos. Los objetivos, en general, son oraciones que se representan mediante un verbo, un objeto conceptual o físico, una dirección de origen o destino y la forma o camino en que se van a lograr. Los escenarios se representan mediante una oración que contiene los siguientes componentes: un agente o actor, un verbo, un objeto conceptual o físico, una dirección de origen o destino y una forma o camino. Los problemas del dominio no se representan en esta propuesta.

Engelsman y Wieringa (2014) proponen el modelado de los objetivos organizacionales, como una herramienta para relacionar la organización con los procesos de tecnología de información que se adelanten al interior de la misma. El objetivo es determinar el impacto y la pertinencia de las aplicaciones de software que se desarrollan en la organización. En este lenguaje no se plantean estructuras para la especificación de los objetivos y tampoco se vinculan los problemas de dominio.

Vargas (2010) plantea reglas gramaticales para la especificación de objetivos y problemas, utilizando como referencia el diagrama de objetivos de KAOS y el diagrama causa-efecto. Establece una aproximación a una relación de consistencia basada en dichas reglas, pero empleando estructuras sintácticas que requieren el uso directo de algunas palabras comunes entre el objetivo y el problema. Esta propuesta no tiene en cuenta relaciones semánticas que determinen los objetivos que se ligan con ciertos problemas.

En la Tabla 2 se presenta un ejemplo de las estructuras para enunciar problemas, en la Tabla 3 un ejemplo para enunciar objetivos y en la Tabla 4 un ejemplo de la relación. Las abreviaturas que se emplean en las tablas, son las siguientes: Ad=Adjetivo, Ad1=Adjetivo, Ad2=Adjetivo, Adv=Adverbio, Adv1=Adverbio, Adv2=Adverbio, C=Conjunción, O=Oración, SN=Sintagma Nominal, SN1=Sintagma Nominal, SN2=Sintagma Nominal, S=Sustantivo, V=Verbo, V1=Verbo, V2=Verbo.

Tabla 2. Estructura gramatical para representar problemas

Caso	Descripción	Restricciones	Ejemplos
1	O→V+Ad+SN	<p>V→{en forma reflexiva impersonal o voz pasiva refleja, por ejemplo: "hay", "existe", "presenta"}</p> <p>Ad→{debe tener una connotación negativa; se sugieren palabras como "bajo", "poco", "malo", etc.}</p>	<p>Se presenta baja producción de materia prima en la fábrica de telares.</p> <p>Existen malas condiciones habitacionales en el conjunto residencial.</p>

Tabla 3. Estructura gramatical para representar objetivos

Caso	Descripción	Restricciones	Ejemplos
1	O→V1+Ad+SN	<p>V1→{Verbo de logro}</p> <p>Ad→{Debe tener connotación positiva. Por ejemplo: "Alta", "buena", "Adecuada"}</p>	<p>Lograr alta producción de materia prima en la fábrica de telares.</p> <p>Lograr buenas condiciones habitacionales en el conjunto residencial.</p>

Tabla 4. Estructura gramatical para relacionar objetivos y problemas

Caso	Descripción	Restricciones	Ejemplos
1	O→V+Ad1+SN  P→V1+Ad2+SN	V→{Verbo de logro}  V1→{en forma reflexiva impersonal o voz pasiva refleja}  SN→{Se usa el mismo sintagma nominal tanto para el objetivo como para el problema.}  Ad1 y Ad2→{Los adjetivos deben poseer connotaciones contrarias.}	Objetivo: Lograr alta producción de materia prima en la fábrica de telares.  Problema: Existe baja producción de materia prima en la fábrica de telares.  Objetivo: Lograr buenas condiciones habitacionales en el conjunto residencial.  Problema: Existen malas condiciones habitacionales en el conjunto residencial.

A nivel de análisis organizacional, Saravia (2004) plantea que la metodología de Marco Lógico utiliza un árbol de objetivos y un árbol de problemas, para la formulación de proyectos y la toma de decisiones. En el árbol de objetivos se representan los objetivos como las situaciones futuras a la que se desea llegar una vez se resuelvan los problemas. Para establecer la relación entre el árbol de objetivos y el árbol de problemas, se toman en consideración algunas reglas como: los estados negativos encontrados en los problemas se convierten en estados positivos alcanzados; los problemas se reformulan convirtiéndolos en objetivos y el problema central detectado se convierte también en un objetivo central del proceso. Todo este proceso lo lleva a cabo en forma manual el analista organizacional. En la Figura 3 se muestra un ejemplo de un árbol de objetivos que presenta problemas de redacción y de claridad y en la Figura 4 un ejemplo de un árbol de problemas que visualiza los mismos problemas del árbol de objetivos.

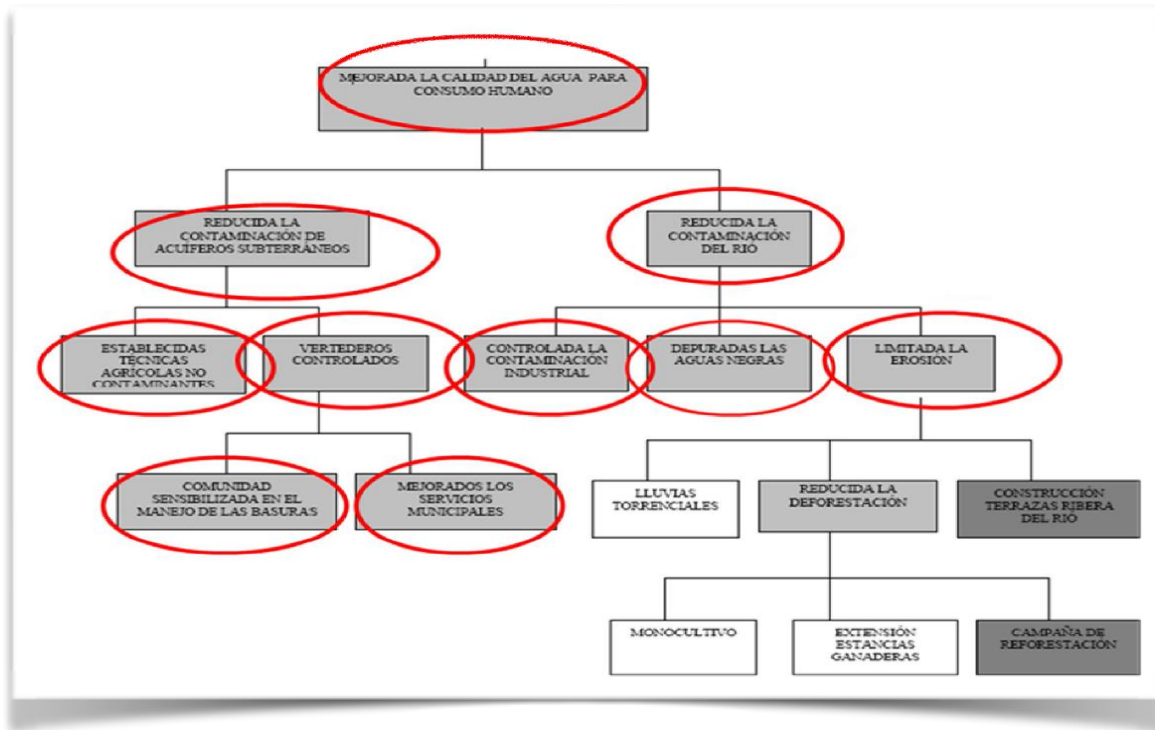


Figura 3. Árbol de objetivos de Marco Lógico con inconsistencias (Vargas 2010, Adaptado de Camacho, Cámara, Cascante & Sainz, 2001)

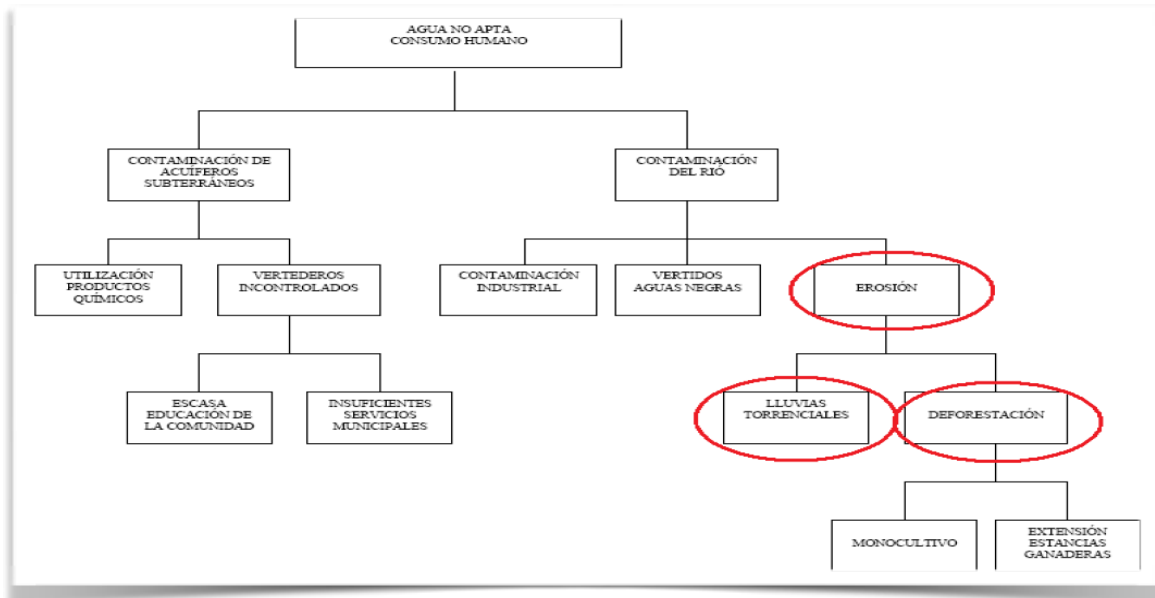


Figura 4. Árbol de problemas de Marco Lógico con inconsistencias (Vargas 2010, Adaptado de Camacho, Cámara, Cascante & Sainz, 2001)



En la metodología Kepner-Tregoe (Kepner & Tregoe, 1976) se implementa un procedimiento para la solución de problemas que facilita la toma de decisiones al interior de la organización. Para ello, se realiza una serie de etapas que van desde análisis de situaciones, el análisis de problemas, el análisis de objetivos de la decisión a tomar y el análisis de problemas potenciales a ocurrir después de la decisión tomada. La metodología exige que los objetivos describan los aspectos que se van a lograr en forma precisa y situarlos en un tiempo y un lugar definido, pero no plantea una estructura precisa para realizar esa descripción.

En la Tabla 5 se muestra un compendio de la revisión de la literatura realizada para esta Tesis Doctoral, especificando algunos de los aspectos relevantes trabajados.





# **3. PLANTEAMIENTO DEL PROBLEMA E HIPÓTESIS**

## **3.1 PROBLEMA DE INVESTIGACIÓN**

### **3.1.1 Descripción del problema**

Teniendo en cuenta la revisión realizada, el problema que se detecta es la poca trazabilidad y consistencia que existe en la relación entre objetivos organizacionales, problemas del dominio y requisitos del sistema dentro de la fase de educación temprana de requisitos de software. Todo esto se debe a dificultades que aún se conservan en las diferentes metodologías y frameworks existentes. I\*, TROPOS y KAOS no permiten establecer de forma directa y consistente una estructura para la descripción de problemas y objetivos y mucho menos la relación que existe entre los objetivos organizacionales, los problemas del dominio y los requisitos. Depende directamente del analista la elaboración de los diagramas, lo cual se hace de forma manual, no existe una estructura sintáctico-semántica que ayude a estructurar los objetivos ni los problemas, para su fácil relación con otros componentes del sistema. Además no cuentan con una representación explícita y formal para la especificación de los problemas, lo que dificulta la trazabilidad y consistencia entre objetivos organizacionales, problemas del dominio y requisitos. Algunos de los objetivos que se plantean en los diagramas no especifican claramente un objetivo, sino que en muchos casos reflejan operaciones y su redacción no es clara en expresar el objetivo que se pretende.

UNC-Method (Zapata & Arango, 2009) y el Business Modeling with UML (Eriksson & Penker, 1999) dependen directamente de la experiencia y conocimiento del analista,

quedando reflejado, en algunos casos, en objetivos poco claros y con problemas estructurales que no se ligan directamente con los problemas del dominio que se desea solucionar. Estos modelos no especifican estructuras sintácticas ni semánticas para representar objetivos y tampoco problemas del dominio.

39

En el NFR framework (Wieringa, Maiden, Mead & Rolland, 2006), todo el proceso lo realiza el analista de forma manual. No existe una representación de problemas del dominio que permita generar trazabilidad y consistencia con los objetivos y los requisitos.

Otros de los problemas que se detectan en el proceso de educación temprana de requisitos son los siguientes: la poca vinculación de los requisitos iniciales (objetivos del sistema) con los requisitos tardíos (funcionalidad final del sistema; Martínez, Pastor, Mylopoulos & Giorgini, 2006); falta una relación entre la funcionalidad esperada de un sistema de información y los procesos de negocio a los que éste dará soporte (Ali, Dalpiaz & Giorgini, 2010); existe poca trazabilidad entre los modelos conceptuales dentro de la educación de requisitos de software (Aguilar, Garrigós & Mazón, 2011); los objetivos del negocio se desligan de la construcción del sistema de información; los requisitos no satisfacen por lo menos uno de los objetivos del negocio (Ali, Dalpiaz & Giorgini, 2010); no se utilizan los objetivos del negocio para determinar la completitud y suficiencia de la especificación de requisitos (Herrmann & Paech, 2008); existe poca relación entre la organización con los procesos de tecnología de información que se desarrollan (Engelsman & Wieringa, 2014); la carencia de una notación gráfica para representar objetivos no permite identificar la forma en la cual se relacionan los actores del negocio, cómo dependen entre ellos, qué tipo de información se intercambia, ni cómo se afecta el proceso de negocio por el cumplimiento o incumplimiento de las acciones de los actores del negocio (Martínez, Pastor & Estrada, 2005); no se utilizan estructuras semánticas en la elaboración de requisitos que faciliten su trazabilidad con otros actores del dominio (Nguyen, Vo, Lumpe & Grundy, 2014); los objetivos del sistema a desarrollar no tienen en cuenta las necesidades prioritarias de la organización (Waseem, 2014); en muchos casos, los problemas detectados en la educación de requisitos no se enuncian con información que denote realmente el problema; los problemas y los objetivos se redactan mal y esto dificulta su relación, trazabilidad y consistencia (Vargas, 2010); no se emplea, en muchos casos, un modelo de organización inicial, que guíe al analista en la obtención de la información relevante desde el contexto de la organización; la representación en un diagrama de objetivos es débil, ya que no satisface las necesidades y expectativas recopiladas en el

## 26 Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas

proceso de educación de requisitos de software (Burnay, Jureta, Linden & Faulkner, 2014); los sistemas de información, en muchos casos, no son la correcta representación de los requisitos capturados desde el modelo del negocio (Estrada, Martínez, Pastor & Sánchez, 2002); se carece de métodos formales para el planteamiento de los enunciados de los problemas y los objetivos, lo cual casi siempre se hace por medio del lenguaje natural (Zapata, Acevedo & Moreno, 2011).

En las Figuras 5 y 6 se muestran los problemas de forma gráfica utilizando el diagrama causa-efecto.

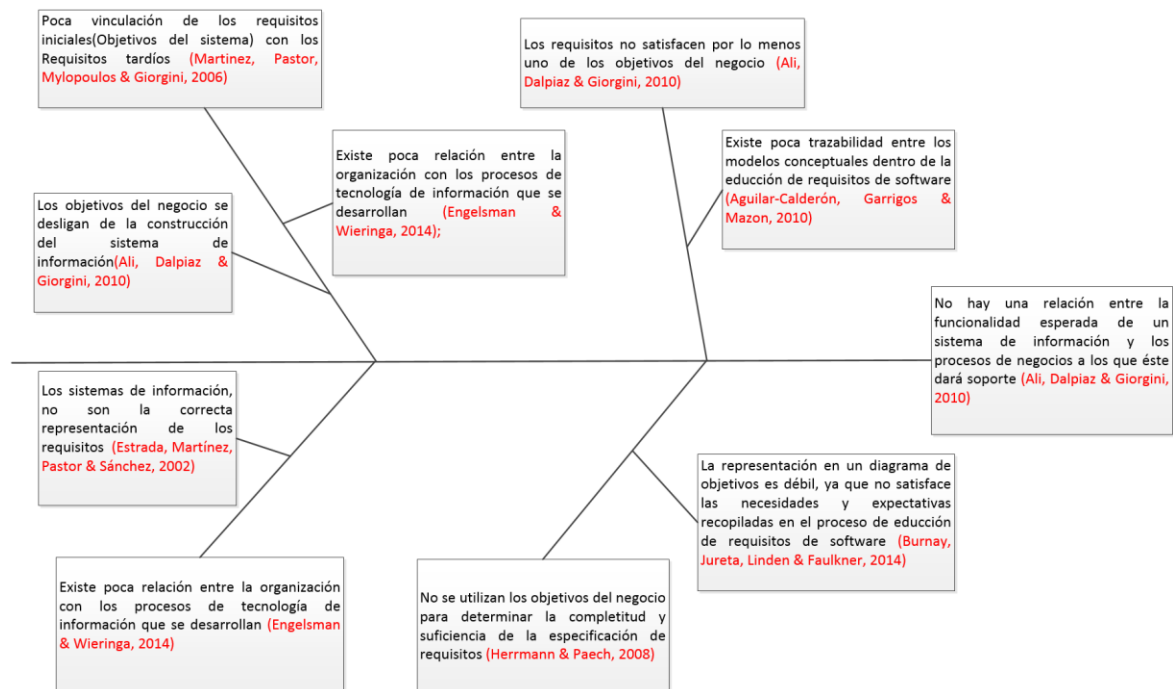


Figura 5. Representación gráfica de problemas (1)

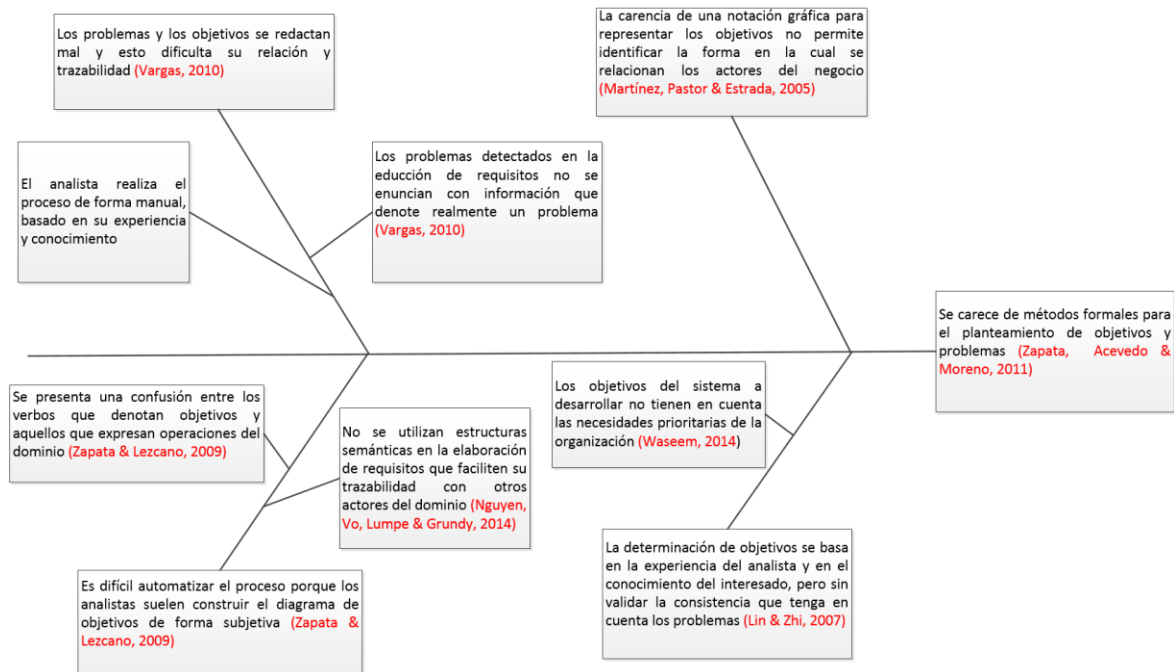


Figura 6. Representación gráfica de problemas (2)

### 3.1.2 Pregunta de Investigación

Debido a las dificultades que se presentan para generar mayor trazabilidad y consistencia en la relación entre objetivos organizacionales, problemas del dominio y requisitos iniciales del sistema que permitan una especificación de requisitos de software contextualizado y pertinente, se plantea la siguiente pregunta de investigación:

¿Es posible desarrollar un modelo para la especificación temprana de requisitos de software, a partir de la relación sintáctica y semántica entre objetivos y problemas?

A continuación, se presenta la sistematización del problema de investigación, la cual consta de un conjunto de preguntas que permitirán descomponer el problema principal en problemas menos complejos:

¿Cómo definir una estructura sintáctica y semántica para la formalización y representación de problemas del dominio en el contexto de la educación temprana de requisitos de software?

¿Cómo definir una estructura sintáctica y semántica para la formalización y representación de objetivos en el contexto de la educación temprana de requisitos de software?

¿Cómo establecer una relación de consistencia y trazabilidad entre objetivos organizacionales, problemas del dominio y requisitos iniciales (objetivos del sistema)?

¿Cómo validar la trazabilidad en la especificación de requisitos de software, a partir de la relación entre objetivos organizacionales, problemas del dominio y requisitos?

## **3.2 HIPÓTESIS DE INVESTIGACIÓN**

A continuación se presenta la hipótesis de investigación que se propuso como opción de solución a la problemática descrita:

HI1:

Es posible establecer una relación sintáctica y semántica entre objetivos organizacionales, problemas del dominio y objetivos del sistema que permita mejorar la trazabilidad en la especificación de requisitos iniciales de software. Esto permitirá reducir la inconsistencia de la información, generando mayor trazabilidad en la especificación de requisitos de software.

## **3.3 OBJETIVOS**

### **3.3.1 Objetivo general**

Diseñar un modelo para la relación sintáctica y semántica entre objetivos organizacionales, problemas del dominio y objetivos del sistema que mejoren la trazabilidad en la especificación temprana de requisitos de software.

### **3.3.2 Objetivos específicos**

- Identificar y analizar los diferentes modelos de especificación de requisitos de software que utilizan problemas y objetivos y relaciones entre ellos.
- Definir una estructura sintáctica y semántica para la formalización y representación de problemas del dominio en el contexto de la educación temprana de requisitos.



- Definir una estructura sintáctica y semántica para la formalización y representación de objetivos organizacionales y del sistema en el contexto de la educación temprana de requisitos.
- Establecer un conjunto de reglas sintácticas y semánticas que permitan establecer una relación consistente y que genere trazabilidad entre objetivos organizacionales, problemas del dominio y objetivos del sistema.
- Definir un conjunto de métricas para la consistencia y trazabilidad del modelo.
- Validar el modelo con un caso de estudio dado en el proceso de educación de requisitos.

### 3.4 CONTRIBUCIÓN

La principal contribución de esta Tesis de Doctorado es un modelo para la relación sintáctica y semántica entre objetivos organizacionales, problemas del dominio y objetivos del sistema, que mejore la consistencia y trazabilidad en la especificación temprana de requisitos de software.

El aporte es innovador, debido a que los modelos y metodologías existentes todavía no cuentan con herramientas que garanticen una relación consistente entre objetivos organizacionales, problemas del dominio y objetivos del sistema en los procesos de educación temprana de requisitos de software. Los problemas no se representan explícitamente lo cual dificulta la trazabilidad. De este modo, el modelo apoyará a los analistas de sistemas y a los interesados en su proceso de educación de requisitos, proporcionándoles una herramienta que facilitará la elaboración y especificación de objetivos y problemas y, por ende, un mejor seguimiento a la especificación inicial de requisitos.

Los principales logros resultantes de la elaboración de este modelo basado en la relación sintáctica y semántica entre objetivos organizacionales, problemas del dominio y objetivos del sistema, se pueden resumir como sigue:

Del trabajo teórico:

- Una formalización y especificación de problemas de dominio aplicables al proceso de educación de requisitos de software, fundamentadas en una estructura sintáctica y semántica.

### 30 **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

---

- Una formalización y especificación de objetivos organizacionales y de objetivos del sistema aplicables al proceso de educación de requisitos de software, basada en una estructura sintáctica y semántica.
- Un conjunto de reglas sintácticas y semánticas para relacionar objetivos organizacionales, problemas del dominio y objetivos del sistema como insumo para la especificación de requisitos consistentes y trazables.

Del trabajo práctico:

Un modelo aplicable en la fase de educación de requisitos de software que relacione objetivos organizacionales, problemas del dominio y objetivos del sistema por medio de un conjunto de reglas sintácticas y semánticas que generen una trazabilidad y consistencia en los requisitos iniciales.

Del trabajo empírico:

La eficiencia del modelo a partir de un conjunto de métricas para la consistencia y trazabilidad. La validación se realiza en un caso de estudio dado en el proceso de educación de requisitos de software.

## **4.SOLUCION**

### **4.1 ESTRUCTURA SINTÁCTICO-SEMÁNTICA PARA REPRESENTAR Y FORMALIZAR OBJETIVOS EN EL CONTEXTO DE LA EDUCACIÓN TEMPRANA DE REQUISITOS DE SOFTWARE**

Se define un conjunto de estructuras sintáctico-semánticas para la representación y formalización de objetivos organizacionales y objetivos del sistema (requisitos iniciales) en el contexto de la educación temprana de requisitos de software, que facilite la relación directa con los problemas del dominio que detecta el analista y valida el interesado.

Las estructuras definidas procuran minimizar los errores encontrados en la especificación de objetivos (organizacionales y del sistema) que se presentan en los diferentes métodos de análisis organizacional e ingeniería de requisitos orientada a objetivos. Entre los errores encontrados, se desatacan los siguientes: mala redacción de los objetivos, algunos de los objetivos describen acciones u operaciones en vez de objetivos, no se utilizan los verbos adecuados para su especificación y, en muchos casos, no es claro lo que se busca con el objetivo.

Para la representación de los objetivos, se tiene en cuenta la estructura general que se plantea en la Figura 7, que muestra la descripción utilizada para plantear las diferentes estructuras que especifican objetivos.

## 32 Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas

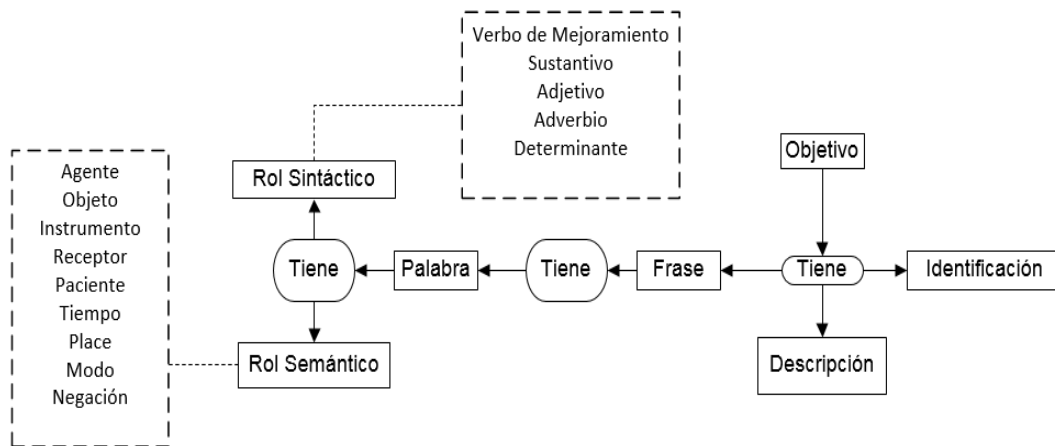


Figura 7. Esquema general para estructurar objetivos

En la Tabla 6 (adaptada de Vargas, 2010) se proponen las estructuras para la especificación y formalización de los objetivos. Las abreviaturas empleadas en dicha tabla, son las siguientes: Ad=Adjetivo, Adv=Adverbio, Adv1=Adverbio, Adv2=Adverbio, C=Conjunción, O=Oración, SN=Sintagma Nominal, SN1=Sintagma Nominal, SN2=Sintagma Nominal, S=Sustantivo, V=Verbo, V1=Verbo, V2=Verbo.

Tabla 6. Estructuras para la especificación y formalización de objetivos

Caso	Descripción	Restricciones	Ejemplos
1	O→V1+SN1+C+SN2	V1→{Verbo de Mejoramiento} C→ {"Para", "de", "del"}	Aumentar las ventas de la empresa
2	O→V1+C+V2+SN+Adv	V1→{Verbo de logro o mejoramiento} C→ {"que"} V2 → {en forma reflexiva impersonal o voz pasiva refleja; se sugieren algunos verbos como: "presentar", "cometer", "realizar", "generar", "utilizar".} Adv→ {debe tener una connotación positiva, se sugieren palabras como: "a tiempo", "correctamente", etc.}	Lograr que se realicen las actualizaciones correctamente.
3	O→V1+C+V2+SN	V1→{Verbo de logro} C→ {"que"} V2 → {Verbo conjugado}	Lograr que se desarrollen mecanismos de control a documentos.

34 **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

Tabla 6. Estructuras para la especificación y formalización de objetivos (Continuación)

Caso	Descripción	Restricciones	Ejemplos
4	O→V+SN	<p>V → {Verbo de Mejoramiento con connotación negativa, por ejemplo “reducir”, “disminuir”, “decrementar”, etc.}</p> <p>SN → {Debe tener connotación negativa; por ejemplo: “demora”, “retraso”, “deserción”, “accidentalidad”, etc.}</p>	<p>Reducir la accidentalidad en la vías de la ciudad</p>
5	O→V1+C+SN1 + V2+SN2+Adv	<p>V1→{Verbo de logro}</p> <p>C→ {"que"}</p> <p>V2→ {verbo conjugado}</p> <p>Adv→ {debe tener una connotación positiva; se sugieren palabras como: “a tiempo”, “correctamente”, etc.}</p>	<p>Lograr que el laboratorista entregue las muestras correctamente.</p>

Tabla 6. Estructuras para la especificación y formalización de objetivos (Continuación)

Caso	Descripción	Restricciones	Ejemplos
6	O→V1+C+SN1 +Adv+V2+SN2	V1→{Verbo de logro} C→ {"que"} Adv→{"no"} V2 → {Verbo conjugado} SN2 → {Debe tener una connotación negativa; por ejemplo: "demoras", "retrasos", "errores", etc.}	Lograr que los datos no tengan errores.
7	O→V1+C+SN+ V2+SN	V1→{Verbo de logro} C→ {"que"} V2→{Verbo conjugado}	Lograr que los equipos de cómputo se conecten a la red.

36 **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

Tabla 6. Estructuras para la especificación y formalización de objetivos (Continuación)

Caso	Descripción	Restricciones	Ejemplos
8	O→V1+C+V2+ SN + Ad	V1→{Verbo de logro} C→ {"que"} V2 → {Verbo conjugado} Ad → {Connotación positiva}	Lograr que se registre el inventario a tiempo
9	O→V1+C+SN+ V2+SN + Ad	V1→{Verbo de logro} C→ {"que"} V2→{Verbo conjugado} Ad → {Connotación positiva}	Garantizar que el empleado reciba el inventario a tiempo
10	O→V1+C+SN+ V2	V1→{Verbo de logro} C→ {"que"} V2→{Verbo conjugado}	Lograr que los informes de inventario se registren



## 4.2 ESTRUCTURA SINTÁCTICO-SEMÁNTICA PARA REPRESENTAR Y FORMALIZAR PROBLEMAS EN EL CONTEXTO DE LA EDUCACIÓN TEMPRANA DE REQUISITOS DE SOFTWARE

Se define un conjunto de estructuras sintáctico-semánticas para la representación de problemas de dominio en el contexto de la educación temprana de requisitos de software, que permita la especificación y formalización de los problemas del dominio, procurando apoyar al analista de sistemas en el análisis organizacional y en el reconocimiento del dominio sobre el cual se requiere desarrollar una aplicación de software.

Las estructuras definidas buscan minimizar algunos errores que se cometen en el momento de especificar un problema detectado durante la fase de educación temprana de requisitos de software. Los errores típicos incluyen: la mala redacción de los problemas, algunos de los problemas no expresan realmente un problema, no son claros de entender para el interesado y, además, se descontextualizan del dominio.

Para la representación de los problemas, se tuvo en cuenta la Figura 8, que muestra el esquema general utilizado para plantear las especificaciones que definen los problemas.

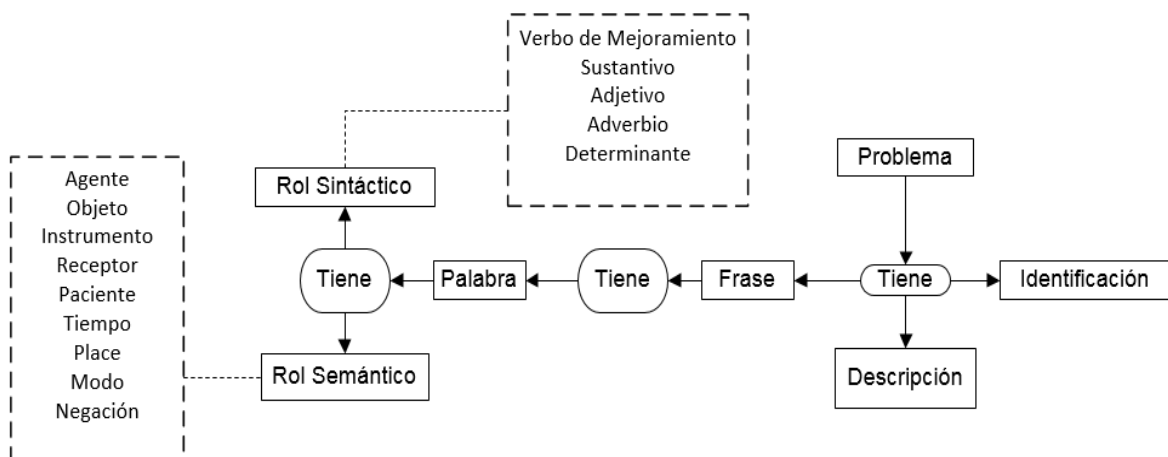


Figura 8. Esquema general para la representación de problemas del dominio.

En la Tabla 7 se especifican las reglas para representar y formalizar los problemas del dominio en el contexto de la educación de requisitos de software. Las abreviaturas utilizadas en las reglas son: Ad=Adjetivo, Adv=Adverbio, Adv1=Adverbio, Adv2=Adverbio, C=Conjunción, O=Oración, SN=Sintagma Nominal, SN1=Sintagma Nominal, SN2=Sintagma Nominal, S=Sustantivo, V=Verbo, V1=Verbo, V2=Verbo.

38 **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

Tabla 7. Estructuras para la especificación y formalización de problemas del dominio

Caso	Descripción	Restricciones	Ejemplos
1	O=SN1+Adv1+V+SN2+Adv2	Adv1=No V=Verbo de Acción Adv2=Adverbio con connotación positiva	La Secretaria no genera facturas eficientemente
Caso	Descripción	Restricciones	Ejemplos
2	O=SN1+V+Adv + +SN2	V=Verbo de Acción Adv=Adverbio con connotación negativa	El profesor entrega inadecuadamente los reportes de las notas
Caso	Descripción	Restricciones	Ejemplos
3	O=SN1+Adv+V +SN2	Adv1=No V = Verbo de Acción	El Despachador no envía el servicio de ambulancia

Tabla 7. Estructuras para la especificación y formalización de problemas del dominio  
(Continuación)

Caso	Descripción	Restricciones	Ejemplos
4	O=SN1+Adv+V+SN2	Adv = No V= Tener	El Proveedor no tiene disponibilidad
Caso	Descripción	Restricciones	Ejemplos
5	O=SN+V+Ad	V= ser o estar Ad= Connotación negativa	Los recursos tecnológicos son limitados
Caso	Descripción	Restricciones	Ejemplos
6	O=SN+Adv+V+Ad	Adv=No V=Verbo ser o estar Ad=Connotación positiva	La Garantía del equipo no está disponible

40 **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

Tabla 7. Estructuras para la especificación y formalización de problemas del dominio  
(Continuación)

Caso	Descripción	Restricciones	Ejemplos
7	O=SN+V1+V2	V1=Verbo ser o estar (Negado) V2= Verbo en voz pasiva refleja	La demanda del producto no se logra
Caso	Descripción	Restricciones	Ejemplos
8	O=SN1+V+SN 2	V=Verbo estructural SN2 = Connotación negativa	La administración de documentos tiene demoras

### 4.3 MODELO PROPUESTO

El modelo se propone para apoyar la especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos (organizacionales y del sistema) y problemas del dominio. Para el modelo, se utilizan como insumos principales las reglas descritas anteriormente para la especificación y formalización de objetivos y de problemas. En la Figura 9 se plantea una visión general del modelo.

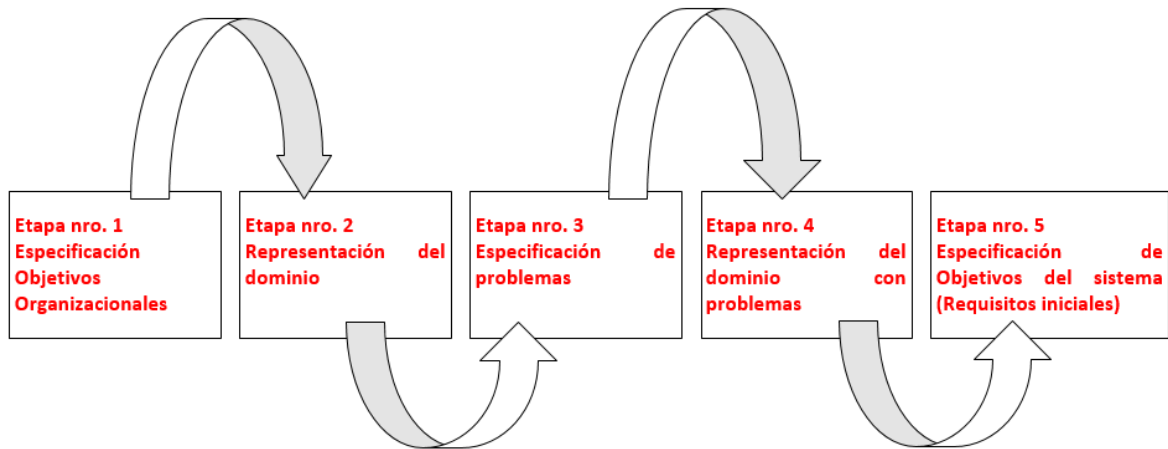


Figura 9. Visión general del modelo

### 4.3.1 Especificación de Objetivos organizacionales

Se define un diagrama de objetivos (se adopta la propuesta de KAOS), donde cada objetivo (organizacional) se estructura de acuerdo con las reglas establecidas en la Sección anterior. El modelo se plantea para definir o actualizar un diagrama de objetivos de acuerdo con el dominio sobre el que se realiza el modelado organizacional. Esta tarea recae directamente en el analista y el interesado. En la Tabla 8 se muestra el proceso de la especificación de objetivos organizacionales.

Tabla 8. Proceso para la especificación y formalización de objetivos organizacionales

Entrada	Actor	Salida
Objetivos organizacionales	Analista	Diagrama de objetivos
Información del interesado sobre el dominio	Interesado	
Estructuras para especificar y formalizar objetivos	Analista	

**Ejemplo:** En Figura 10 se muestra un diagrama de objetivos de KAOS que define un interesado antes de aplicarle las reglas.

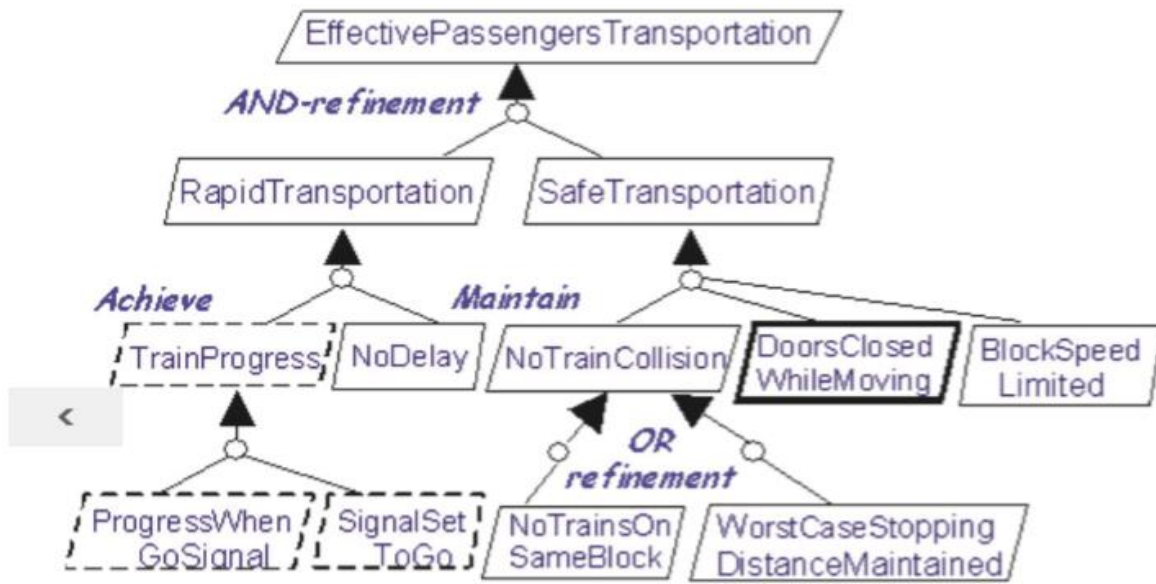


Figura. 10 Diagrama de Objetivos que define el interesado (Zapata y Vargas, 2014)

En la Figura 11 se muestra el mismo diagrama luego de aplicarle las reglas definidas en la Sección anterior.

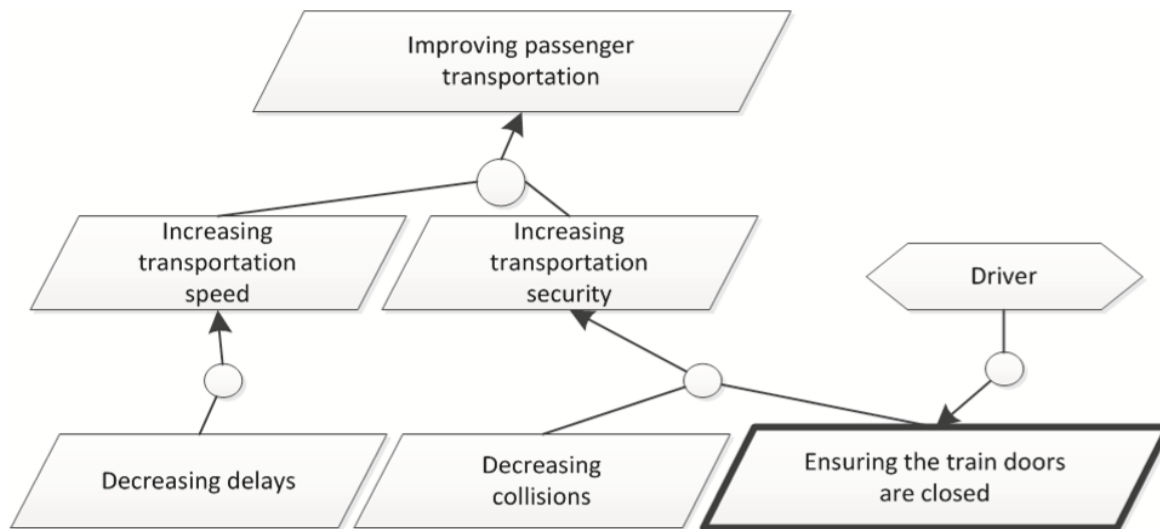


Figura. 11 Diagrama de Objetivos luego de aplicar las reglas (Zapata y Vargas, 2014)

### 4.3.2 Representación del dominio

Se realiza una representación del dominio del interesado, donde se reflejen los objetivos descritos en la etapa anterior. Esto se hace para formalizar ante el interesado el dominio y los objetivos organizacionales del área. Para esta representación del dominio se utilizan

los esquemas preconceptuales. En la Tabla 9 se muestra el proceso de representación del dominio.

Tabla 9. Proceso para la representación del dominio con objetivos

Entrada	Actor	Salida
Diagrama de objetivos	Analista	Representación del dominio con objetivos
Sintaxis de los esquemas preconceptuales	Analista	
Información del interesado sobre el dominio	Interesado	

**Ejemplo.** En la Figura 12 se muestra la representación del dominio del interesado, donde se visualizan los objetivos establecidos en el proceso de especificación de objetivos organizacionales.

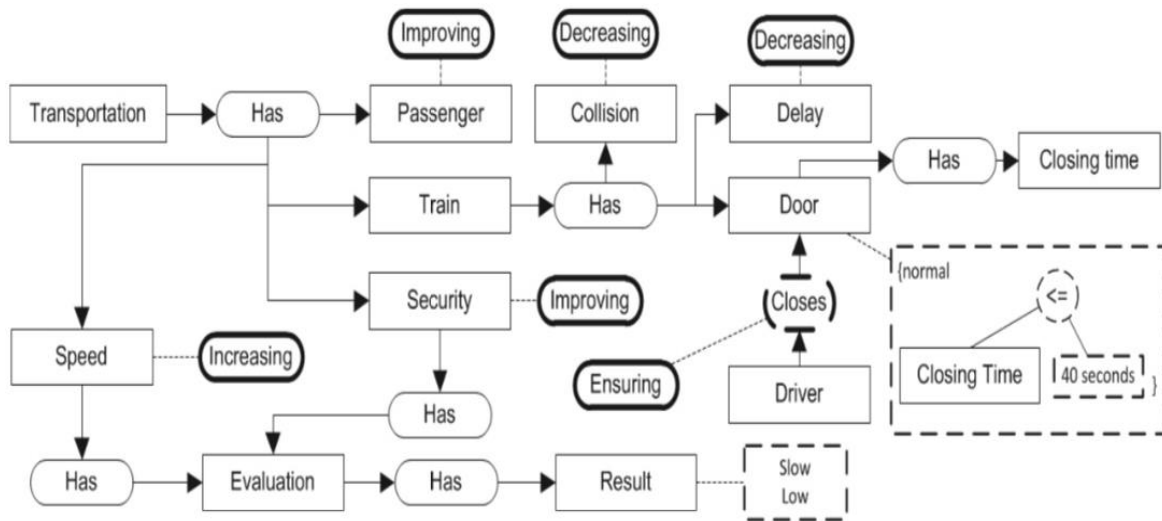


Figura. 12 Representación del dominio con objetivos (Zapata y Vargas, 2014)

### 4.3.3 Especificación de problemas

Se aplican las reglas descritas anteriormente para especificar y formalizar los problemas del dominio que se pueden generar a partir de los objetivos organizacionales descritos y plasmados en la representación del dominio. Todo esto permite clarificar a los interesados

#### 44 Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas

---

los posibles problemas que se pueden presentar a partir de los objetivos del área. En la Tabla 10 se muestra el proceso para la especificación de problemas del dominio.

Tabla 10. Proceso para la especificación de problemas del dominio

<b>Entrada</b>	<b>Actor</b>	<b>Salida</b>
Representación del dominio con objetivos	Analista e Interesado	Especificación y formalización de problemas del dominio.
Reglas para la especificación de problemas (descritas anteriormente)	Analista	
Reglas para vincular objetivos con problemas del dominio (véase la Tabla 11)	Analista	

En la Tabla 11 se especifican las reglas para vincular objetivos con problemas del dominio.



Tabla. 11. Reglas para convertir objetivos en problemas del dominio

Nro.	Antecedente	Consecuente	Ejemplo
1	Especificación de un objetivo donde se utilice un verbo de mejoramiento de connotación positiva para calificar el sustantivo  Estructura de objetivos (regla: 1)	Enunciar el problema con un adjetivo de connotación negativa que califique al sustantivo, donde se continúe con el sustantivo descrito en el objetivo.	<b>Objetivo</b> Aumentar las ventas de la empresa  <b>Problema (regla 5)</b> Las ventas de la empresa son pobres
		Enunciar el problema con un adjetivo de connotación negativa que califique al sustantivo, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<b>Objetivo</b> Aumentar las ventas de la empresa  <b>(regla 5)</b> La atención al cliente es pobre
		Enunciar el problema con un adverbio de connotación negativa, donde se continúe con el sustantivo descrito en el objetivo.	<b>Objetivo</b> Aumentar las ventas de la empresa  <b>(regla 2)</b> El vendedor realiza inadecuadamente las ventas
		Enunciar el problema con un adverbio de connotación negativa, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio	<b>Objetivo</b> Aumentar las ventas de la empresa  <b>(regla 4)</b> La empresa no tiene clientes
		Enunciar el problema con un sustantivo de connotación negativa, que se relacione con el descrito en el objetivo y que se refleje en el contexto del dominio.	<b>Objetivo</b> Aumentar las ventas de la empresa  <b>(regla 8)</b> El servicio al cliente tiene demoras
		Enunciar el problema con la negación de un verbo que califique al sustantivo, donde se continúe con el sustantivo descrito en el objetivo.	<b>Objetivo</b> Aumentar las ventas de la empresa  <b>(regla 3)</b> El vendedor no realiza ventas
		Enunciar el problema con la negación de un verbo que califique al sustantivo, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<b>Objetivo</b> Aumentar las ventas de la empresa  <b>(regla 7)</b> Las ventas de la empresa no se logran

46 **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

Tabla. 11. Reglas para convertir objetivos en problemas del dominio (Continuación)

Nro.	Antecedente	Consecuente	Ejemplo
2	Especificación de un objetivo donde se utilice un verbo de logro para calificar el sustantivo y/o se emplee un adverbio de connotación positiva.  Estructura de objetivos (reglas: 2,3)	Enunciar el problema con un adjetivo de connotación negativa que califique al sustantivo, donde se continúe con el sustantivo descrito en el problema.	<b>Objetivo</b> Lograr que se entreguen las muestras a tiempo.  <b>(regla 5)</b> La demanda de las muestras es limitada
		Enunciar el problema con un adjetivo de connotación negativa que califique al sustantivo, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<b>Objetivo</b> Lograr que se entreguen las muestras a tiempo.  <b>(regla 5)</b> Los recursos tecnológicos del Laboratorio son pobres.
		Enunciar el problema con un adverbio de connotación negativa, donde se continúe con el sustantivo descrito en el objetivo.	<b>Objetivo</b> Lograr que se entreguen las muestras a tiempo.  <b>(regla 2)</b> El Laboratorio entrega a destiempo las muestras.
		Enunciar el problema con un adverbio de connotación negativa, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<b>Objetivo</b> Lograr que se entreguen las muestras a tiempo  <b>(regla 1)</b> El técnico del laboratorio no entrega las muestras rápidamente.
		Enunciar el problema con un sustantivo de connotación negativa relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<b>Objetivo</b> Lograr que se entreguen las muestras a tiempo  <b>(regla 8)</b> La producción de muestras tiene demoras

Tabla. 11. Reglas para convertir objetivos en problemas del dominio (Continuación)

Nro.	Antecedente	Consecuente	Ejemplo
2	Especificación de un objetivo donde se utilice un verbo de logro para calificar el sustantivo y/o se emplee un adverbio de connotación positiva.	Enunciar el problema con la negación de un verbo que califique al sustantivo, donde se continúe con el sustantivo descrito en el objetivo.	<b>Objetivo</b> Lograr que se entreguen las muestras a tiempo  <b>(regla 3)</b> El técnico no entrega las muestras.
	Estructura de objetivos (reglas: 2,3)	Enunciar el problema con la negación de un verbo que califique al sustantivo, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<b>Objetivo</b> Lograr que se entreguen las muestras a tiempo  <b>(regla 6)</b> Las muestras no están disponibles.
Nro.	Antecedente	Consecuente	Ejemplo
3	Especificación de un objetivo donde se utilice un verbo de mejoramiento de connotación negativa para calificar el sustantivo.  Estructura de objetivos (Regla: 4)	Enunciar el problema con un adjetivo de connotación negativa que califique al sustantivo, donde se continúe con el sustantivo descrito en el problema.	<b>Objetivo</b> Disminuir la deserción escolar en la Universidad. <b>(regla 5)</b> La tasa de retención en la Universidad es pobre
		Enunciar el problema con un adjetivo de connotación negativa que califique al sustantivo, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<b>Objetivo</b> Disminuir la deserción escolar en la Universidad. <b>(regla 5)</b> La Matrícula del estudiante es limitada
		Enunciar el problema con un adverbio de connotación negativa, donde se continúe con el sustantivo descrito en el problema.	<b>Objetivo</b> Disminuir la deserción escolar en la Universidad. <b>(regla 3)</b> La Universidad no entrega ayudas educativas
		Enunciar el problema con un adverbio de connotación negativa, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio expresado en el esquema preconceptual.	<b>Objetivo</b> Disminuir la deserción escolar en la Universidad. <b>(regla 2)</b> El Sistema académico entrega inadecuadamente los recibos de pago

48 **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

Tabla. 11. Reglas para convertir objetivos en problemas del dominio (Continuación)

Nro.	Antecedente	Consecuente	Ejemplo
3	Especificación de un objetivo donde se utilice un verbo de mejoramiento de connotación negativa para calificar el sustantivo.  Estructura de objetivos (Regla: 4)	Enunciar el problema con un sustantivo de connotación negativa relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio expresado en el esquema preconceptual.	<b>Objetivo</b> Disminuir la deserción escolar en la Universidad.  <b>(regla 8)</b> El sistema académico tiene demoras.
		Enunciar el problema con la negación de un verbo que califique al sustantivo, donde se continúe con el sustantivo descrito en el objetivo	<b>Objetivo</b> Disminuir la deserción escolar en la Universidad.  <b>(regla 7)</b> La deserción escolar no se registra
		Enunciar el problema con la negación de un verbo que califique al sustantivo, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio expresado en el esquema preconceptual.	<b>Objetivo</b> Disminuir la deserción escolar en la Universidad.  <b>(regla 6)</b> El Sistema de pago no está disponible
Nro.	Antecedente	Consecuente	Ejemplo
4	Especificación de un objetivo donde se utilice un verbo de logro para calificar el sustantivo, un verbo de acción para completar la frase y se emplee un adverbio de connotación positiva.  Estructura de objetivos (Regla: 5)	Enunciar el problema con un adjetivo de connotación negativa que califique al sustantivo, donde se continúe con el sustantivo descrito en el problema.	<b>Objetivo</b> Lograr que el proveedor despache las solicitudes eficientemente  <b>(regla 5)</b> El despacho de solicitudes el limitado
		Enunciar el problema con un adjetivo de connotación negativa que califique al sustantivo, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio	<b>Objetivo</b> Lograr que el proveedor despache las solicitudes eficientemente  <b>(regla 5)</b> La Capacidad instalada del proveedor es limitada

Tabla. 11. Reglas para convertir objetivos en problemas del dominio (Continuación)

Nro.	Antecedente	Consecuente	Ejemplo
4	Especificación de un objetivo donde se utilice un verbo de logro para calificar el sustantivo, un verbo de acción para completar la frase y se emplee un adverbio de connotación positiva.  Estructura de objetivos (Regla: 5)	Enunciar el problema con un adverbio de connotación negativa, donde se continúe con el sustantivo descrito en el problema.	<p><b>Objetivo</b> Lograr que el proveedor despache las solicitudes eficientemente</p> <p><b>(regla 2)</b> El proveedor despacha lentamente las solicitudes.</p>
		Enunciar el problema con un adverbio de connotación negativa, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<p><b>Objetivo</b> Lograr que el proveedor despache las solicitudes eficientemente</p> <p><b>(regla 6)</b> El stock de la solicitud no está disponible</p>
		Enunciar el problema con un sustantivo de connotación negativa, relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<p><b>Objetivo</b> Lograr que el proveedor despache las solicitudes eficientemente</p> <p><b>(regla 8)</b> El despacho de solicitudes tiene demoras</p>
		Enunciar el problema con la negación de un verbo que califique al sustantivo. Donde se continúe con el sustantivo descrito en el objetivo.	<p><b>Objetivo</b> Lograr que el proveedor despache las solicitudes eficientemente.</p> <p><b>(regla 3)</b> El proveedor no despacha las solicitudes.</p>
		Enunciar el problema con la negación de un verbo que califique al sustantivo, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<p><b>Objetivo</b> Lograr que el proveedor despache las solicitudes eficientemente.</p> <p><b>(regla 4)</b> El proveedor no tiene insumos</p>

50 **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

Tabla. 11. Reglas para convertir objetivos en problemas del dominio (Continuación)

Nro.	Antecedente	Consecuente	Ejemplo
5	Especificación de un objetivo donde se utilice un verbo de logro para calificar el sustantivo, un verbo de acción para completar la frase y se emplee un segundo sustantivo de connotación negativa.	Enunciar el problema con un adjetivo de connotación negativa que califique al sustantivo, donde se continúe con el sustantivo descrito en el problema.	<p><b>Objetivo</b> Lograr que la gestión de documentos no tenga demoras.</p> <p><b>(regla 5)</b> La gestión de documentos es limitada.</p>
	Estructura de objetivos (Regla: 6)	Enunciar el problema con un adjetivo de connotación negativa que califique al sustantivo, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<p><b>Objetivo</b> Lograr que la gestión de documentos no tenga demoras.</p> <p><b>(regla 5)</b> Los recursos tecnológicos son limitados</p>
		Enunciar el problema con un adverbio de connotación negativa, donde se continúe con el sustantivo descrito en el problema.	<p><b>Objetivo</b> Lograr que la gestión de documentos no tenga demoras.</p> <p><b>(regla 3)</b> El archivista no entrega los documentos</p>
		Enunciar el problema con un adverbio de connotación negativa, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<p><b>Objetivo</b> Lograr la gestión de documentos</p> <p><b>(regla 2)</b> El archivista gestiona inadecuadamente los documentos</p>
		Enunciar el problema con un sustantivo de connotación negativa relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<p><b>Objetivo</b> Lograr la gestión de documentos</p> <p><b>(regla 8)</b> La gestión de documentos tiene retrasos</p>

Tabla. 11. Reglas para convertir objetivos en problemas del dominio (Continuación)

Nro.	Antecedente	Consecuente	Ejemplo
5	Especificación de un objetivo donde se utilice un verbo de logro para calificar el sustantivo, un verbo de acción para completar la frase y se emplee un segundo sustantivo de connotación negativa.  Estructura de objetivos (Regla: 6)	Enunciar el problema con la negación de un verbo que califique al sustantivo, donde se continúe con el sustantivo descrito en el objetivo.	<b>Objetivo</b> Lograr la gestión de documentos  <b>(regla 3)</b> El archivista no gestiona los documentos
		Enunciar el problema con la negación de un verbo que califique al sustantivo, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<b>Objetivo</b> Lograr la gestión de documentos  <b>(regla 6)</b> El documento no está disponible
Nro.	Antecedente	Consecuente	Ejemplo
6	Especificación de un objetivo donde se utilice un verbo de logro para calificar el sustantivo, un verbo de acción para completar la frase.  Estructura de objetivos (Regla: 7)	Enunciar el problema con un adjetivo de connotación negativa que califique al sustantivo, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio expresado en el esquema preconceptual.	<b>Objetivo</b> Lograr que los equipos de cómputo se conecten a la red.  <b>(regla 5)</b> La conexión de los equipos es limitada.
		Enunciar el problema con un adjetivo de connotación negativa que califique al sustantivo, donde se continúe con el sustantivo descrito en el problema.	<b>Objetivo</b> Lograr que los equipos de cómputo se conecten a la red. <b>(regla 5)</b> Los equipos de cómputo son obsoletos.
		Enunciar el problema con un adverbio de connotación negativa, donde se continúe con el sustantivo descrito en el problema.	<b>Objetivo</b> Lograr que los equipos de cómputo se conecten a la red. <b>(regla 2)</b> Los equipos de cómputo se conectan inadecuadamente a la red.
		Enunciar el problema con un adverbio de connotación negativa, especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio expresado en el esquema preconceptual.	<b>Objetivo</b> Lograr que los equipos de cómputo se conecten a la red. <b>(regla 3)</b> Los equipos de cómputo no se conectan a la red.

52 **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

Tabla. 11. Reglas para convertir objetivos en problemas del dominio (Continuación)

Nro.	Antecedente	Consecuente	Ejemplo
6	Especificación de un objetivo donde se utilice un verbo de logro para calificar el sustantivo, un verbo de acción para completar la frase.  Estructura de objetivos (Regla: 7)	Enunciar el problema con un sustantivo de connotación negativa. Relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio.	<b>Objetivo</b> Lograr que los equipos de cómputo se conecten a la red  <b>(regla 8)</b> La trasmisión de la red tiene demoras.
		Enunciar el problema con la negación de un verbo que califique al sustantivo. Donde se continúe con el sustantivo descrito en el objetivo.	<b>Objetivo</b> Lograr que los equipos de cómputo se conecten a la red  <b>(regla 3)</b> El ingeniero no realiza la conexión
		Enunciar el problema con la negación de un verbo que califique al sustantivo. Especificando un sustantivo, un verbo, un adjetivo o un adverbio relacionado con el descrito en el objetivo y que se refleje en el contexto del dominio expresado en el esquema preconceptual.	<b>Objetivo</b> Lograr que los equipos de cómputo se conecten a la red  <b>(regla 7)</b> La velocidad trasmisión de la red no se cumple

En la Figura 13 se visualiza la formalización de un problema del dominio teniendo en cuenta los objetivos planteados y las reglas definidas.

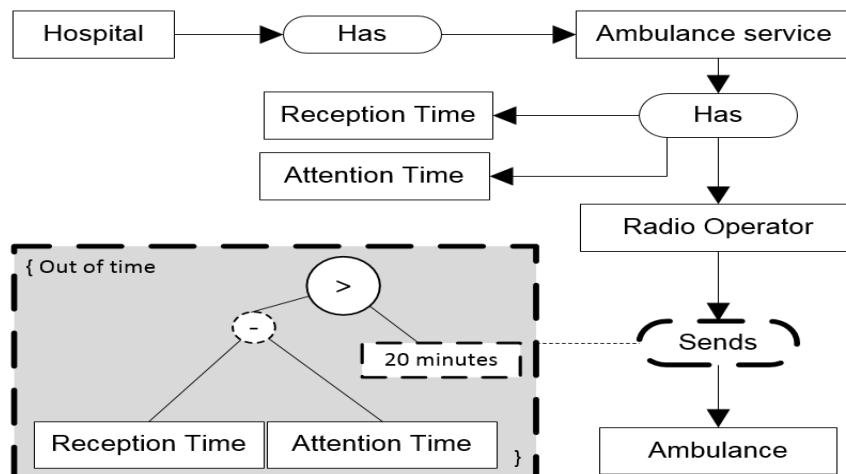


Figura 13. Formalización de un problema de dominio (Zapata y Vargas, 2014)



### 4.3.4 Representación de dominio con problemas

Se genera la representación del dominio actualizada con la especificación de los problemas detectados y formalizados en la etapa anterior. Todo esto permite visualizar los posibles problemas que se pueden presentar a partir de los objetivos del área. En la Tabla 12 se muestra el proceso para la representación del dominio con problemas asociados.

Tabla 12. Proceso para la representación del dominio con vinculación de problemas

Entrada	Actor	Salida
Representación del dominio	Analista e Interesado	Representación del dominio vinculando los problemas detectados.
Problemas especificados y formalizados	Analista	

En la Figura 14 se muestra la representación del dominio con la incorporación de los problemas detectados y formalizados.

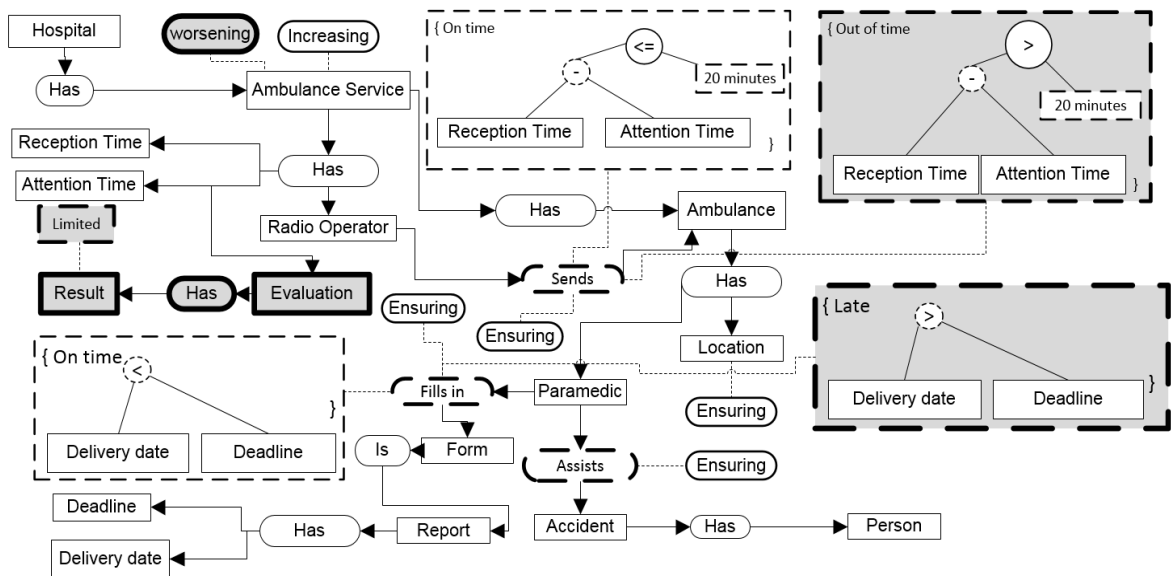


Figura 14. Representación del dominio con la incorporación de los problemas

### 4.3.5 Especificación y formalización de objetivos del sistema (requisitos iniciales)

## 54 Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas

---


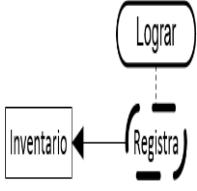
Se especifican y formalizan los objetivos del sistema a partir de los problemas del dominio que se detectan, utilizando para ello las estructuras definidas en las Secciones anteriores. Todo esto permite realizar una especificación de objetivos del sistema (requisitos iniciales del software). En la Tabla 13 se muestra el proceso para la especificación y formalización de objetivos del sistema como requisitos iniciales para el desarrollo del software.

Tabla 13. Proceso para la especificación y formalización de objetivos del sistema

<b>Entrada</b>	<b>Actor</b>	<b>Salida</b>
Representación del dominio con problemas detectados	Analista e Interesado	Especificación y formalización de Objetivos del sistema.
Reglas para la especificación de problemas y objetivos (descritas anteriormente)	Analista	
Reglas para vincular problemas con objetivos del sistema (véase la Tabla 14)	Analista	

En la Tabla 14 se especifican las reglas para vincular problemas del dominio con objetivos del sistema (requisitos iniciales).

Tabla 14. Reglas para vincular problemas del dominio con objetivos del sistema

Nro.	Antecedente	Consecuencia	Ejemplo
1	<p>Especificación de problemas en los cuales se utiliza la negación (no) para expresar la connotación negativa del problema o se utiliza un adverbio de connotación negativa para expresar el problema</p>	<p>Enunciar el objetivo con un verbo de logro y eliminado la negación descrita en el problema, haciendo referencia al sustantivo, verbo y adverbio descrito en el problema.</p>	<p><b>Problema</b> El jefe de inventario no registra el inventario rápidamente</p> <p><b>(regla 2)</b> Lograr que se registre el inventario rápidamente</p> 
	<p>(Problemas caso 1, 2, 3, 5, 6)</p>	<p>Enunciar el objetivo con un verbo de logro y eliminado la negación descrita en el problema, haciendo referencia a un sustantivo, verbo o adverbio descrito en el dominio y que tenga relación con el problema enunciado.</p>	<p><b>Problema</b> El jefe de inventario no registra el inventario</p> <p><b>(regla 3)</b> Lograr que se registre el inventario</p> 

56 **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

Tabla 14. Reglas para vincular problemas del dominio con objetivos del sistema  
(Continuación)

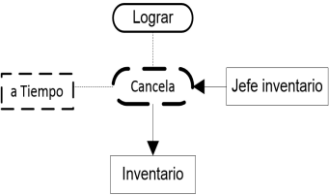
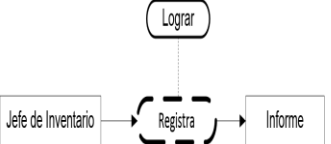
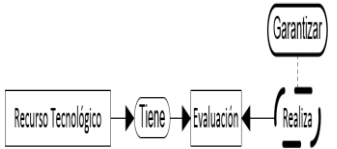
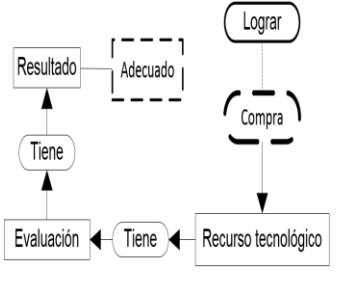
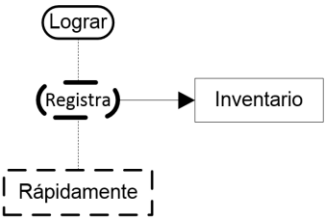
Nro.	Antecedente	Consecuencia	Ejemplo
1	<p>Especificación de problemas en los cuales se utiliza la negación (no) para expresar la connotación negativa del problema o se utiliza un adverbio de connotación negativa para expresar el problema</p> <p>(Problemas caso 1, 2, 3, 5, 6)</p>	<p>Enunciar el objetivo con un verbo de logro, empleando un adverbio contrario al descrito en el problema, haciendo referencia al sustantivo o al verbo descrito en el problema.</p> <p>Enunciar el objetivo con un verbo de logro, eliminando el adverbio de connotación negativa descrito en el problema y haciendo referencia a un sustantivo o verbo descrito en el dominio y que tenga relación con el problema enunciado.</p>	<p><b>Problema</b> El jefe de inventario cancela a destiempo el inventario</p> <p><b>(regla 5)</b> Lograr que el jefe de inventario cancele el inventario a tiempo.</p>  <p><b>Problema</b> Los informes del inventario no se registran</p> <p><b>(regla 7)</b> Lograr que el jefe de inventario registre los informes</p> 

Tabla 14. Reglas para vincular problemas del dominio con objetivos del sistema  
(Continuación)

Nro.	Antecedente	Consecuencia	Ejemplo
2	<p data-bbox="380 411 633 674">Especificación de problemas en los cuales el adjetivo constituye la connotación negativa de la frase que expresa el problema.</p> <p data-bbox="380 709 633 806">Estructuras de problemas (Caso 4)</p>	<p data-bbox="659 411 1065 705">Enunciar el objetivo con un verbo de logro que permita eliminar la connotación negativa descrita en el problema, especificando un sustantivo relacionado con el descrito en el problema y que se refleje en el contexto del dominio.</p>	<p data-bbox="1091 411 1425 508"><b>Problema</b> Los recursos tecnológicos son limitados</p> <p data-bbox="1091 541 1425 674"><b>(regla 3)</b> Garantizar que se realice la evaluación de los recursos tecnológicos</p> 
		<p data-bbox="659 867 1065 1161">Enunciar el objetivo con un verbo de logro, reemplazando el adjetivo de connotación negativa con uno de connotación positiva, especificando un sustantivo relacionado con el descrito en el problema y que se refleje en el contexto del dominio.</p>	<p data-bbox="1091 867 1425 963"><b>Problema</b> Los recursos tecnológicos son limitados</p> <p data-bbox="1091 997 1425 1129"><b>(regla 8)</b> Lograr que se compren recursos tecnológicos adecuados</p> 

58 **Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas**

Tabla 14. Reglas para vincular problemas del dominio con objetivos del sistema  
(Continuación)

Nro.	Antecedente	Consecuencia	Ejemplo
3	<p>Especificación de problemas en los cuales el sustantivo constituye la connotación negativa de la frase que expresa el problema presente en la oración.</p> <p>Estructuras de problemas (Caso 7)</p>	<p>Enunciar el objetivo con un verbo de logro que permita eliminar la connotación negativa descrita en el problema. Se elimina el sustantivo de connotación negativa y se reemplaza con un adverbio o un adjetivo de connotación positiva.</p>	<p><b>Problema</b> El informe de inventario tiene demoras</p> <p><b>(regla 1)</b> Lograr que se registre el inventario rápidamente</p> 

## 5.METRICAS DEL MODELO

Las métricas de trazabilidad y consistencia que se definen para el modelo propuesto buscan unas prácticas de control que permitan obtener objetivos del sistema (requisitos iniciales) más pertinentes y contextualizados con la organización y con las necesidades que expresa el interesado en el dominio del problema.

Las métricas que se proponen a continuación se trabajan de la siguiente forma:

- Métricas base: Son métricas que determinan el número de elementos más significativos presentes en el modelo propuesto y que son requisitos para la medición que se propone y para la construcción de las métricas derivadas (Rolón, Ruiz, Rubio & Piattini, 2005).
- Métricas derivadas: Son aquellas que se obtiene como resultado de aplicar funciones de medición en otras métricas base y/o derivadas (Rolón, Ruiz, Rubio & Piattini, 2005).

Con las métricas base y derivadas propuestas, es posible medir la consistencia y trazabilidad del modelo propuesto.

Para la aplicación de las métricas se deben tener en cuenta las siguientes definiciones:

**Relación semántica:** Conceptos representados en el dominio que se unen mediante una relación estructural o conceptos que tengan el concepto tipo con un valor asociado.

Ejemplo: en las Figuras 15 y 16 se visualiza la relación semántica entre dos conceptos.



Figura 15. Ejemplo de relación semántica (1)

El concepto respuesta tiene una relación semántica con el concepto información y viceversa.



Figura 16. Ejemplo de relación semántica (2)

El concepto informe tiene una relación semántica con el valor notas

**Relación Sintáctica:** Se da entre conceptos, adjetivos, adverbios y negaciones iguales y también entre relaciones dinámicas (o de logro) iguales, es decir, sólo se presenta cuando los elementos son iguales.

El objetivo de la métrica propuesta es medir la consistencia y trazabilidad de los problemas del dominio y de los objetivos del sistema respecto de los objetivos.

- Indicador: Que el 100% de los problemas del dominio tengan una relación sintáctica o semántica con al menos un objetivo.
- Indicador: Que el 100% de los objetivos del sistema tengan una relación sintáctica o semántica con al menos un problema del dominio.

En la Tabla 15 se definen las métricas base propuestas



Tabla 15. Métricas base


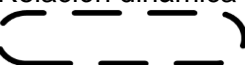
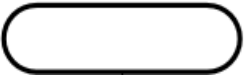

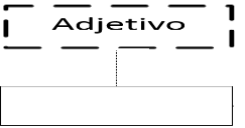


Notación	Nombre	Métrica	Definición
No aplica	NPEF	Número de problemas del dominio especificados y formalizados	Indica el número de problemas que se especifican y formalizan dentro del modelo
No aplica	NOSEF	Número de objetivos del sistema especificados y formalizados	Indica el número de objetivos del sistema que se especifican y formalizan dentro del modelo
Concepto  Relación dinámica  Relación de logro  Nota (adverbio)  Nota (adjetivo)  Nota (valor)  Negación 	NEO	Número de elementos del objetivo (Conceptos, relaciones dinámicas, relaciones de logro, notas (adverbios), notas (adjetivos), notas (valores), negaciones) especificados en un objetivo	Indica el número de elementos presentes en un objetivo formalizado

Tabla 15. Métricas base (Continuación)

Notación	Nombre	Métrica	Definición
No aplica	NERSS	Número de elementos con relación sintáctica o relación semántica entre un objetivo y un problema	Indica el número de elementos que tienen una relación sintáctica o relación semántica entre un objetivo y un problema teniendo en cuenta la representación del dominio

En la Tabla 16 se definen las métricas derivadas

Tabla 16. Métricas derivadas

Nombre	Métrica	Formula	Definición
NRSS	Nivel de relación sintáctica y semántica entre un objetivo y un problema	$NRSS = (NERSS * 100\%) / NEO$	Indica el nivel de relación entre un objetivo y un problema.  Se considera que un problema tiene relación con un objetivo cuando el $NRSS \geq 50\%$
CPRO	Cantidad de problemas que tiene relación con los objetivos	No Aplica	Indica el número de problemas que se relacionan con los objetivos.
CORP	Cantidad de objetivos del sistema que tienen relación con los problemas del dominio	No Aplica	Indica el número de objetivos del sistema que se relacionan con los problemas del dominio.

Tabla 16. Métricas derivadas (Continuación)

<b>Nombre</b>	<b>Métrica</b>	<b>Formula</b>	<b>Definición</b>
NCTM	Nivel de consistencia y trazabilidad del modelo	$\text{NCTM} = \frac{((\text{CPRO} + \text{CORP}) * 100\%)}{(\text{NOSEF} + \text{NPEF})}$	<p>Indica el nivel de consistencia y trazabilidad del modelo.</p> <p>Se considera que el modelo es consistente y trazable cuando el <math>\text{NCTM} \geq 70\%</math></p>

## **6.CASO DE ESTUDIO**

### **6.1 CASO DE ESTUDIO NRO. 1**

Especificación y formalización de objetivos del sistema (requisitos iniciales) a partir de los objetivos del área de registro académico que se definen por una institución de educación básica y de los problemas que se detectan dentro del proceso de educación de requisitos de software. Esto se realiza con el objetivo de implementar un software académico que apoye dicha área. El caso de estudio se realizó en la Empresa de Software IDE Sistemas (Informática para el desarrollo empresarial) Ltda. ([www.idesistemas.co](http://www.idesistemas.co)), dentro de su proceso de documentación requerida en la construcción del software SINCO (Software de calificaciones y administración de gestión académica). Para este caso de estudio se aplican todas las etapas del modelo propuesto.

**Etapas nro. 1.** Especificación de los objetivos organizacionales (objetivos del área de registro académico). En la Tabla 17 se visualiza la especificación de los objetivos a partir de los objetivos que se definen en el área.

Tabla 17. Especificación de objetivos (caso de estudio 1)

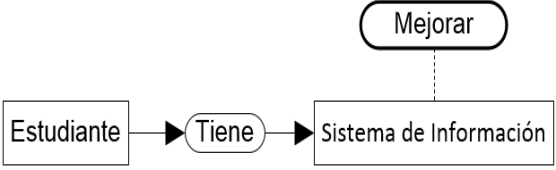
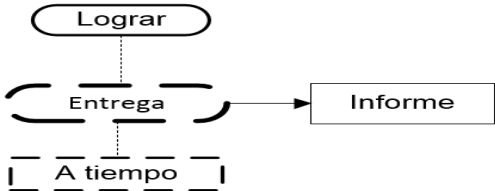
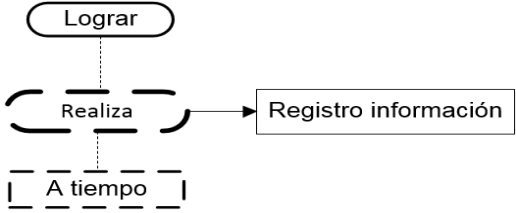
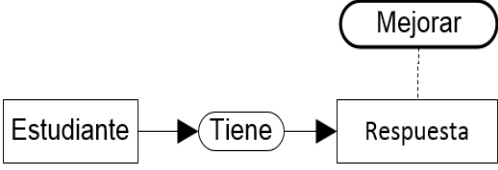
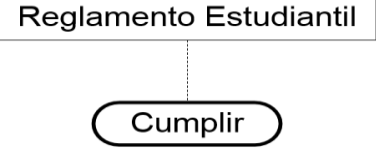
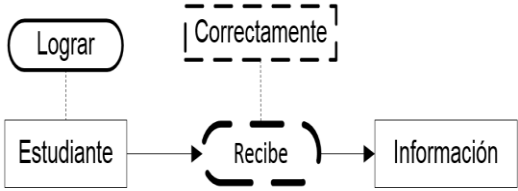
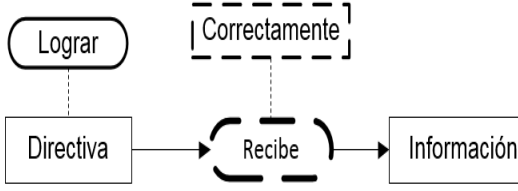
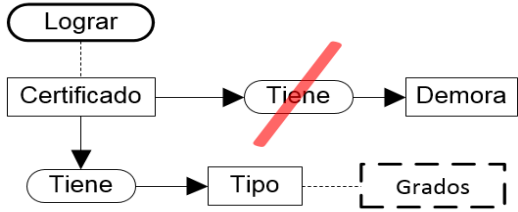
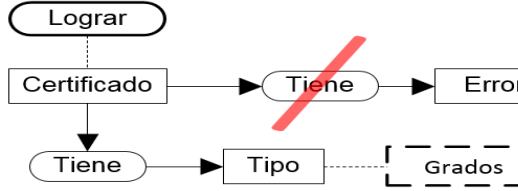
Objetivos propuestos en el área	Objetivos utilizando las estructuras
<p>Registrar, mantener y procesar toda la información de los estudiantes de la institución.</p>	<p>Mejorar el sistema de información para los estudiantes</p>  <p>Lograr que se entreguen los informes a tiempo</p>  <p>Lograr que se realice el registro de información a tiempo</p>  <p>Mejorar las respuestas a los estudiantes</p> 
<p>Efectuar el cumplimiento del Reglamento estudiantil vigente</p>	<p>Cumplir el reglamento estudiantil</p> 

Tabla 17. Especificación de objetivos (caso de estudio 1) (Continuación)

Objetivos propuestos en el área	Objetivos utilizando las estructuras
<p>Generar información estadística de las asignaturas, cursos, alumnos y cualquiera otra información que sea relevante para la toma de decisiones.</p>	<p>Lograr que el estudiante reciba la información correctamente</p>  <pre> graph LR     Lograr1(Lograr) --- Estudiante[Estudiante]     Estudiante --&gt; Recibe1(Recibe)     Recibe1 --- Informacion1[Información]     Recibe1 -.-&gt; Correctamente1[Correctamente]     style Recibe1 stroke-dasharray: 5 5     style Correctamente1 stroke-dasharray: 5 5     </pre> <p>Lograr que la directivas reciban la información correctamente</p>  <pre> graph LR     Lograr2(Lograr) --- Directiva[Directiva]     Directiva --&gt; Recibe2(Recibe)     Recibe2 --- Informacion2[Información]     Recibe2 -.-&gt; Correctamente2[Correctamente]     style Recibe2 stroke-dasharray: 5 5     style Correctamente2 stroke-dasharray: 5 5     </pre>
<p>Certificar los grados académicos y títulos que el alumno haya cursado efectivamente.</p>	<p>Lograr que los certificados de grados no tengan demoras</p>  <pre> graph LR     Lograr3(Lograr) --- Certificado3[Certificado]     Certificado3 --&gt; Tiene3(Tiene)     Tiene3 --- Demora3[Demora]     Certificado3 --&gt; Tiene4(Tiene)     Tiene4 --&gt; Tipo3[Tipo]     Tipo3 -.-&gt; Grados3[Grados]     style Tiene3 stroke: red, stroke-width: 2px     </pre> <p>Lograr que los certificados de grados no tengan errores</p>  <pre> graph LR     Lograr4(Lograr) --- Certificado4[Certificado]     Certificado4 --&gt; Tiene5(Tiene)     Tiene5 --- Error4[Error]     Certificado4 --&gt; Tiene6(Tiene)     Tiene6 --&gt; Tipo4[Tipo]     Tipo4 -.-&gt; Grados4[Grados]     style Tiene5 stroke: red, stroke-width: 2px     </pre>

En la Figura 17 se visualiza el diagrama de objetivos que se define y acuerda con el interesado.

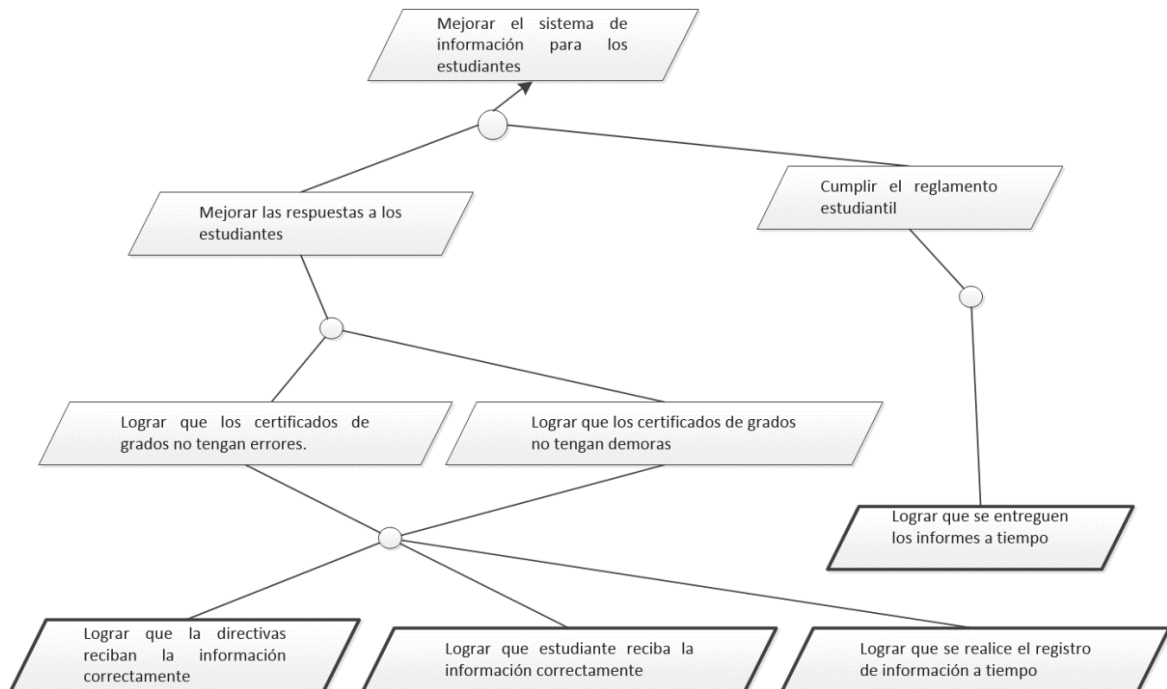


Figura 17. Diagrama de objetivos acordado con el interesado (caso de estudio 1)

**Etapa nro. 2.** Representación del dominio del área de registro académico incorporando los objetivos definidos. En la Figura 18 se visualiza la representación del dominio con objetivos.

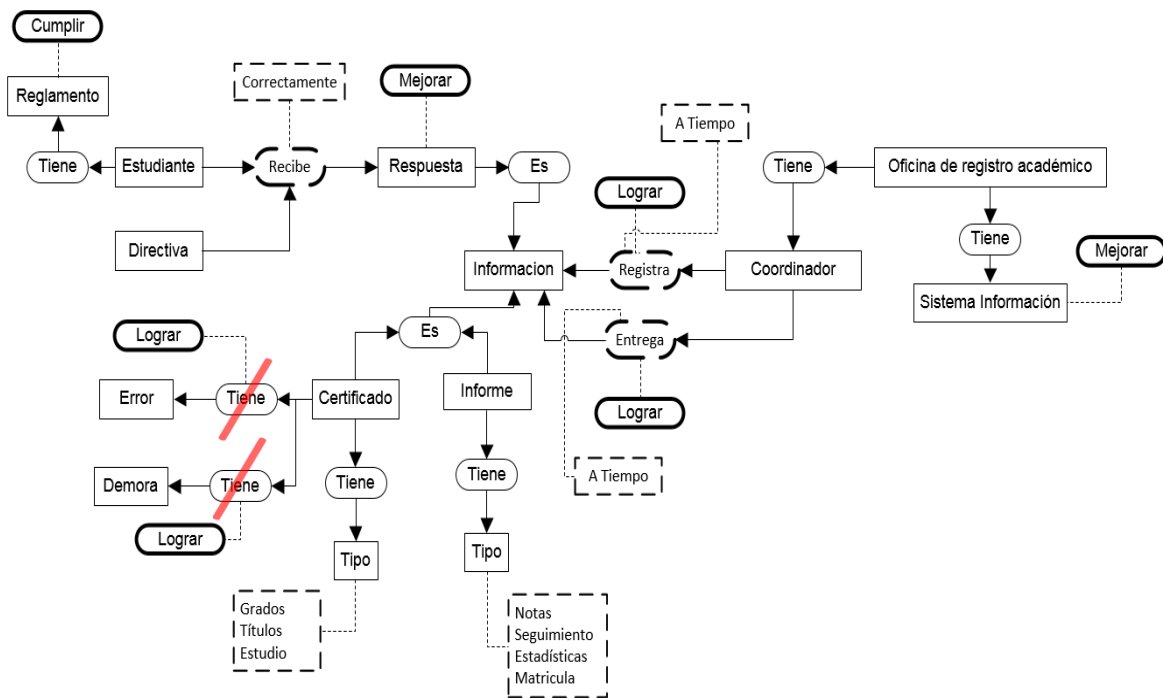


Figura 18. Representación del dominio con objetivos del área (Caso de estudio 1)

**Etapá nro. 3.** Especificación y formalización de problemas del dominio teniendo en cuenta los objetivos del área descritos anteriormente. En la Tabla 18 se muestran los problemas formalizados y especificados.



Tabla 18. Especificación y formalización de problemas (caso de estudio 1)

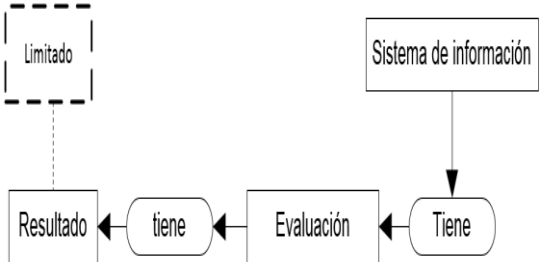
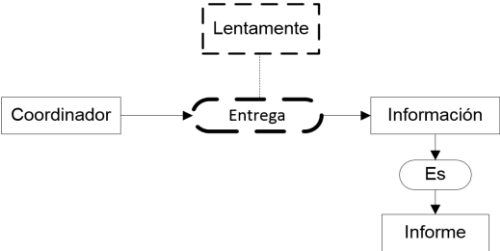
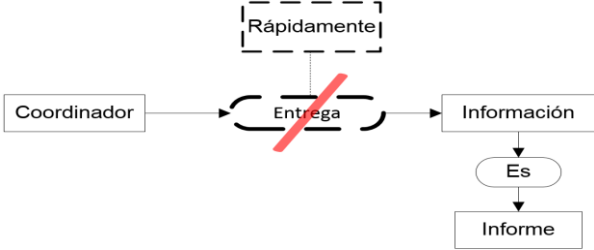

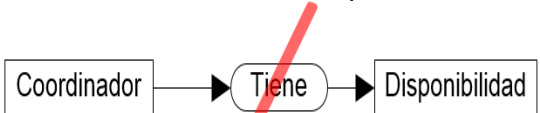
Objetivos	Problemas
<p>Mejorar el sistema de información para los estudiantes</p>	<p>El Sistema de información es limitado</p> 
<p>Lograr que se entreguen los informes a tiempo</p>	<p>El coordinador entrega lentamente los informes</p>  <p>El coordinador no entrega los informes rápidamente</p> 
<p>Lograr que se realice el registro de información a tiempo</p>	<p>El coordinador no registra la información</p>  <p>El coordinador no tiene disponibilidad</p> 

Tabla 18. Especificación y formalización de problemas (caso de estudio 1) (Continuación)

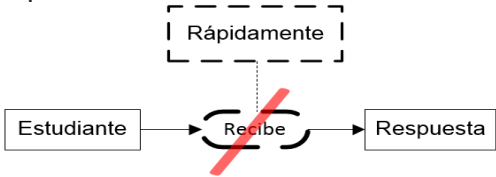
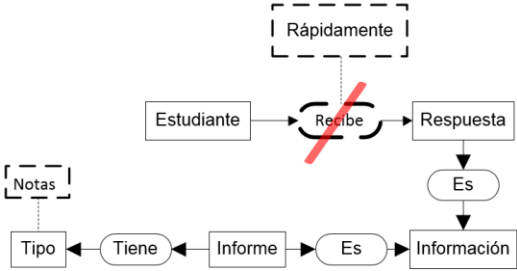
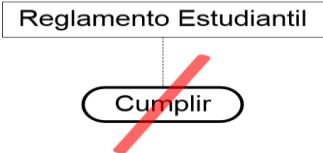
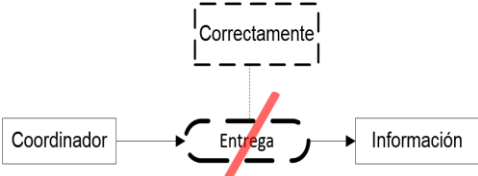
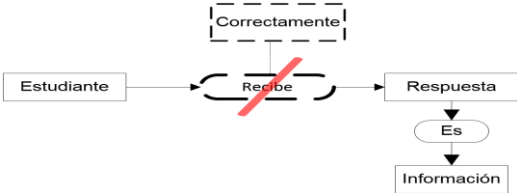
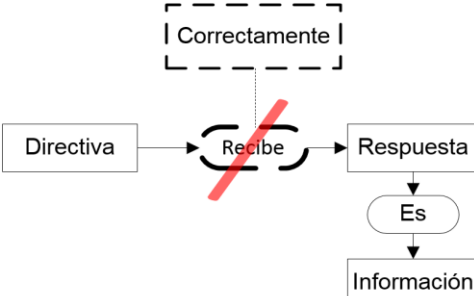
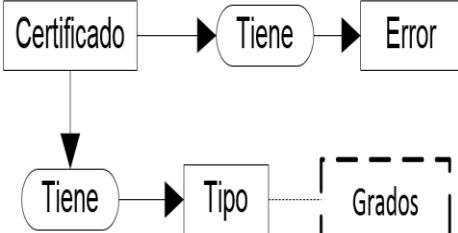
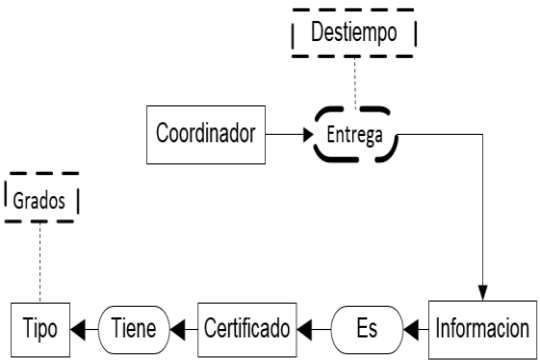
Objetivos	Problemas
<p>Mejorar las respuestas a los estudiantes</p>	<p>El estudiante no recibe las respuestas rápidamente</p>  <p>Los estudiantes no reciben sus notas rápidamente</p> 
<p>Lograr que se cumpla el reglamento estudiantil</p>	<p>El reglamento estudiantil no se cumple</p> 
<p>Lograr que el estudiante reciba la información correctamente</p>	<p>El coordinador no entrega la información correctamente</p>  <p>El estudiante no recibe la información correctamente</p> 

Tabla 18. Especificación y formalización de problemas (caso de estudio 1) (Continuación)

Objetivos	Problemas
<p>Lograr que la directivas reciban la información correctamente</p>	<p>Las directivas no reciben la información correctamente</p>  <pre> graph LR     Directiva[Directiva] --&gt; Recibe((Recibe))     Recibe --&gt; Respuesta[Respuesta]     Respuesta --&gt; Es((Es))     Es --&gt; Informacion[Información]     Correctamente[Correctamente] -.-&gt; Recibe     style Recibe stroke:#f00,stroke-width:2px     </pre>
<p>Lograr que los certificados de grados no tengan demoras</p>	<p>Los certificados de grado tienen errores</p>  <pre> graph LR     Certificado[Certificado] --&gt; Tiene1((Tiene))     Tiene1 --&gt; Error[Error]     Certificado --&gt; Tiene2((Tiene))     Tiene2 --&gt; Tipo[Tipo]     Tipo -.-&gt; Grados[Grados]     </pre> <p>El coordinador entrega los certificados de grado a destiempo</p>  <pre> graph TD     Grados[Grados] -.-&gt; Entrega((Entrega))     Coordinador[Coordinador] --&gt; Entrega     Entrega --&gt; Informacion[Información]     Informacion --&gt; Es((Es))     Es --&gt; Certificado[Certificado]     Certificado --&gt; Tiene((Tiene))     Tiene --&gt; Tipo[Tipo]     style Entrega stroke:#f00,stroke-width:2px     </pre>

**Etapa nro. 4.** Representación del dominio con vinculación de problemas. En las Figuras 19 y 20 se visualiza la representación del dominio con la vinculación de los problemas encontrados.

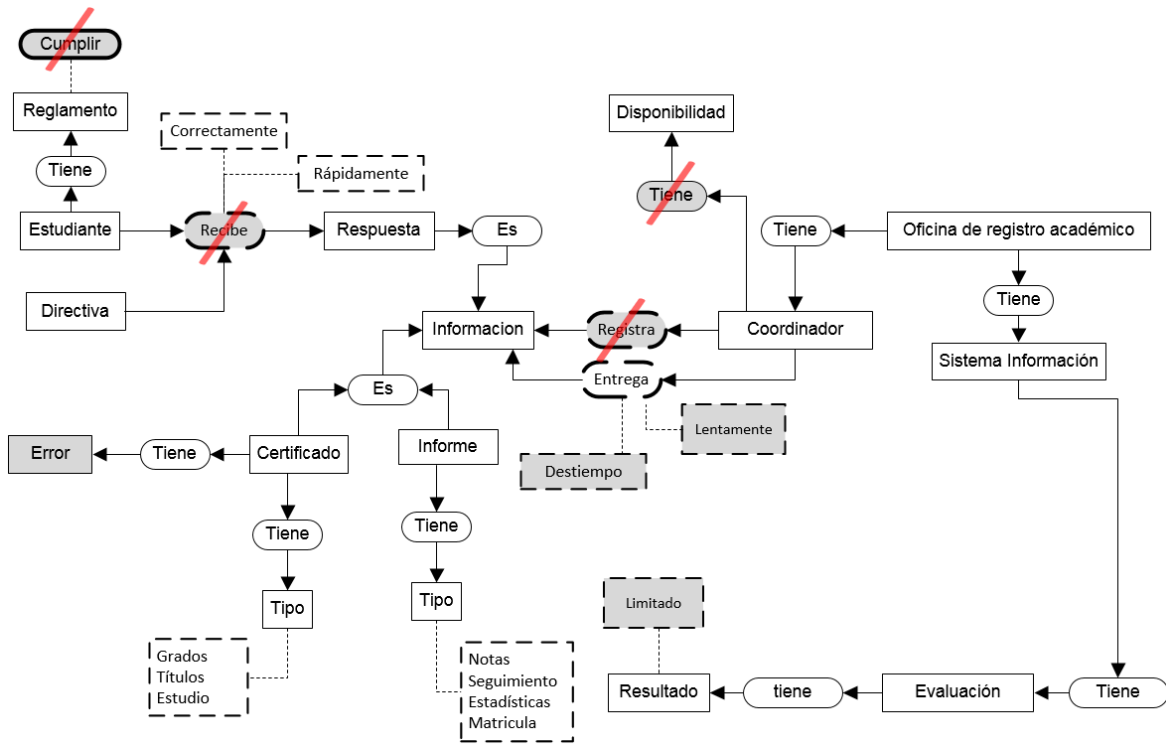


Figura 19. Representación del dominio con los problemas encontrados parte 1 (caso de estudio 1)

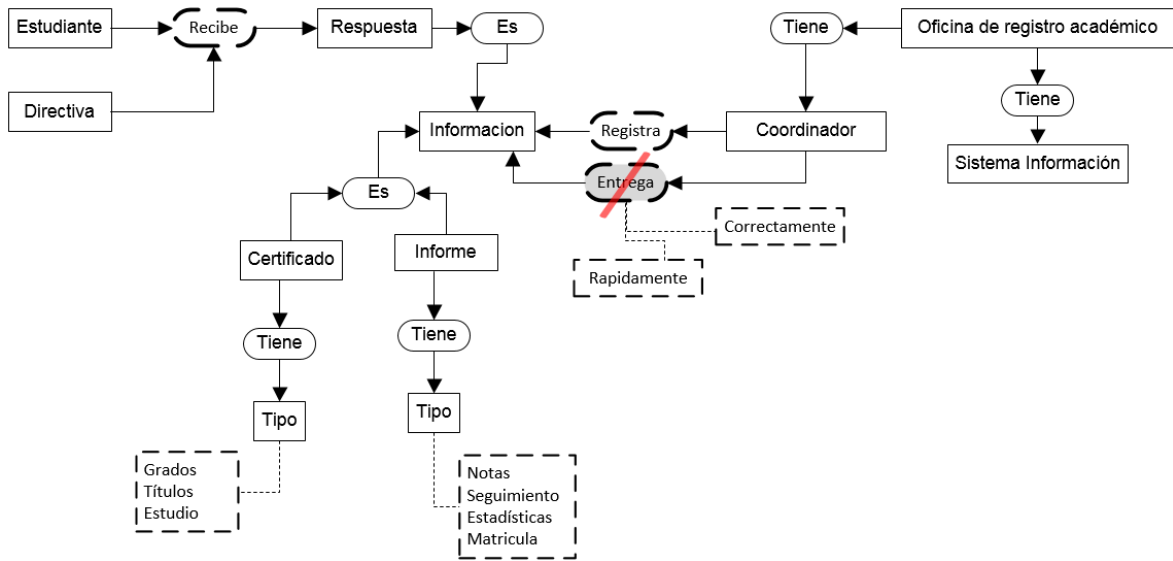
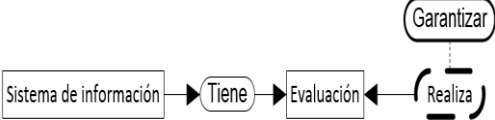
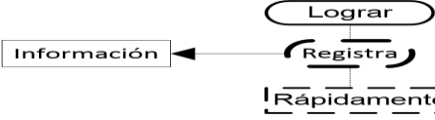
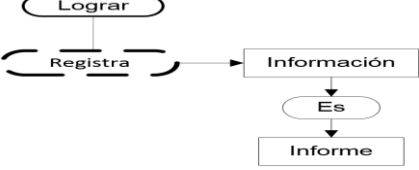
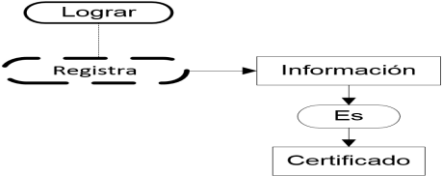
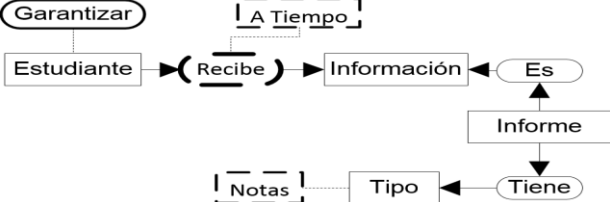


Figura 20. Representación del dominio con los problemas encontrados parte 2 (caso de estudio 1)

**Etapa nro. 5.** Especificación de los objetivos del sistema (requisitos iniciales). A partir de los problemas del dominio que se detectan. En la Tabla 19 se especifican algunos de los objetivos del sistema definidos.

Tabla 19. Especificación de objetivos del sistema (caso de estudio 1)

Problemas Detectados	Objetivos del sistema
El Sistema de información es limitado	<p>Garantizar que se realice la evaluación del sistema de información</p> 
<p>El coordinador entrega lentamente los informes</p> <p>El coordinador no entrega los informes rápidamente</p>	<p>Lograr que se registre la información rápidamente</p> 
El coordinador no registra la información	<p>Lograr que los informes se registren</p>  <p>Lograr que los certificados se registren</p> 
Los estudiantes no reciben sus notas rápidamente	<p>Garantizar que el estudiante reciba las notas a tiempo</p> 

En la Tabla 20 se muestran los resultados de las métricas definidas y que se aplican en el caso de estudio 1 para mostrar la trazabilidad y consistencia del modelo. Abreviaturas a tener en cuenta: Obj = objetivo; P = problema; Objs = objetivo del sistema.

Tabla 20. Aplicación de las métricas base (caso de estudio 1)

Nombre	Métrica	Valor
NPEF	Número de problemas del dominio especificados y formalizados	13
NOSEF	Número de objetivos del sistema especificados y formalizados	5
NEO	Número de elementos del objetivo (Conceptos, relaciones dinámicas, notas (adverbios), notas (adjetivos), negaciones, relaciones de logro) especificados en un objetivo	NEO (Obj1) = 3 NEO(Obj2) = 4 NEO(Obj3) = 4 NEO (Obj4) = 3 NEO(Obj5) = 2 NEO(Obj6) = 5 NEO(Obj7) = 5 NEO(Obj8) = 6 NEO(Obj9) = 6 NEO(Objs1) = 4 NEO(Objs2) = 4 NEO(Objs3) = 4 NEO(Objs4) = 4 NEO(Objs5) = 8
NERSS	Número de elementos con relación sintáctica o relación semántica entre un objetivo y un problema	NERSS (Obj1 y P1) = 1 NERSS (Obj1 y P2) = 0 NERSS (Obj1 y P3) = 0 NERSS (Obj1 y P4) = 0 NERSS (Obj1 y P5) = 0 NERSS (Obj1 y P6) = 1 NERSS (Obj1 y P7) = 1 NERSS (Obj1 y P8) = 0 NERSS (Obj1 y P9) = 0 NERSS (Obj1 y P10) = 0 NERSS (Obj1 y P11) = 1 NERSS (Obj1 y P12) = 0 NERSS (Obj1 y P13) = 0

Tabla 20. Aplicación de las métricas base (caso de estudio 1) (Continuación)

Nombre	Métrica	Valor
NERSS	Número de elementos con relación sintáctica o relación semántica entre un objetivo y un problema	<p>NERSS (Obj2 y P1) = 0  NERSS (Obj2 y P2) = 2  NERSS (Obj2 y P3) = 2  NERSS (Obj2 y P4) = 1  NERSS (Obj2 y P5) = 0  NERSS (Obj2 y P6) = 1  NERSS (Obj2 y P7) = 1  NERSS (Obj2 y P8) = 0  NERSS (Obj2 y P9) = 1  NERSS (Obj2 y P10) = 1  NERSS (Obj2 y P11) = 1  NERSS (Obj2 y P12) = 0  NERSS (Obj2 y P13) = 0</p> <p>NERSS (Obj3 y P1) = 0  NERSS (Obj3 y P2) = 0  NERSS (Obj3 y P3) = 0  NERSS (Obj3 y P4) = 0  NERSS (Obj3 y P5) = 0  NERSS (Obj3 y P6) = 0  NERSS (Obj3 y P7) = 0  NERSS (Obj3 y P8) = 0  NERSS (Obj3 y P9) = 0  NERSS (Obj3 y P10) = 0  NERSS (Obj3 y P11) = 0  NERSS (Obj3 y P12) = 0  NERSS (Obj3 y P13) = 0</p> <p>NERSS (Obj4 y P1) = 0  NERSS (Obj4 y P2) = 1  NERSS (Obj4 y P3) = 1  NERSS (Obj4 y P4) = 1  NERSS (Obj4 y P5) = 0  NERSS (Obj4 y P6) = 2  NERSS (Obj4 y P7) = 2  NERSS (Obj4 y P8) = 0  NERSS (Obj4 y P9) = 1  NERSS (Obj4 y P10) = 2  NERSS (Obj4 y P11) = 1  NERSS (Obj4 y P12) = 0  NERSS (Obj4 y P13) = 1</p>

Tabla 20. Aplicación de las métricas base (caso de estudio 1) (Continuación)

Nombre	Métrica	Valor
NERSS	Número de elementos con relación sintáctica o relación semántica entre un objetivo y un problema	<p>NERSS (Obj5 y P1) = 0  NERSS (Obj5 y P2) = 0  NERSS (Obj5 y P3) = 0  NERSS (Obj5 y P4) = 0  NERSS (Obj5 y P5) = 0  NERSS (Obj5 y P6) = 0  NERSS (Obj5 y P7) = 0  NERSS (Obj5 y P8) = 2  NERSS (Obj5 y P9) = 0  NERSS (Obj5 y P10) = 0  NERSS (Obj5 y P11) = 0  NERSS (Obj5 y P12) = 0  NERSS (Obj5 y P13) = 0</p> <p>NERSS (Obj6 y P1) = 0  NERSS (Obj6 y P2) = 1  NERSS (Obj6 y P3) = 1  NERSS (Obj6 y P4) = 1  NERSS (Obj6 y P5) = 0  NERSS (Obj6 y P6) = 3  NERSS (Obj6 y P7) = 3  NERSS (Obj6 y P8) = 0  NERSS (Obj6 y P9) = 2  NERSS (Obj6 y P10) = 4  NERSS (Obj6 y P11) = 3  NERSS (Obj6 y P12) = 1  NERSS (Obj6 y P13) = 1</p> <p>NERSS (Obj7 y P1) = 0  NERSS (Obj7 y P2) = 1  NERSS (Obj7 y P3) = 1  NERSS (Obj7 y P4) = 1  NERSS (Obj7 y P5) = 0  NERSS (Obj7 y P6) = 2  NERSS (Obj7 y P7) = 2  NERSS (Obj7 y P8) = 0  NERSS (Obj7 y P9) = 2  NERSS (Obj7 y P10) = 2  NERSS (Obj7 y P11) = 4  NERSS (Obj7 y P12) = 1  NERSS (Obj7 y P13) = 1</p>



Tabla 20. Aplicación de las métricas base (caso de estudio 1) (Continuación)

Nombre	Métrica	Valor
NERSS	Número de elementos con relación sintáctica o relación semántica entre un objetivo y un problema	<p>NERSS (Obj8 y P1) = 0  NERSS (Obj8 y P2) = 1  NERSS (Obj8 y P3) = 1  NERSS (Obj8 y P4) = 1  NERSS (Obj8 y P5) = 0  NERSS (Obj8 y P6) = 0  NERSS (Obj8 y P7) = 1  NERSS (Obj8 y P8) = 0  NERSS (Obj8 y P9) = 1  NERSS (Obj8 y P10) = 1  NERSS (Obj8 y P11) = 1  NERSS (Obj8 y P12) = 3  NERSS (Obj8 y P13) = 3</p> <p>NERSS (Obj9 y P1) = 0  NERSS (Obj9 y P2) = 1  NERSS (Obj9 y P3) = 2  NERSS (Obj9 y P4) = 2  NERSS (Obj9 y P5) = 1  NERSS (Obj9 y P6) = 1  NERSS (Obj9 y P7) = 3  NERSS (Obj9 y P8) = 0  NERSS (Obj9 y P9) = 2  NERSS (Obj9 y P10) = 2  NERSS (Obj9 y P11) = 2  NERSS (Obj9 y P12) = 4  NERSS (Obj9 y P13) = 3</p> <p>NERSS (Objs1 y P1) = 2  NERSS (Objs1 y P2) = 0  NERSS (Objs1 y P3) = 0  NERSS (Objs1 y P4) = 0  NERSS (Objs1 y P5) = 0  NERSS (Objs1 y P6) = 0  NERSS (Objs1 y P7) = 0  NERSS (Objs1 y P8) = 0  NERSS (Objs1 y P9) = 0  NERSS (Objs1 y P10) = 0  NERSS (Objs1 y P11) = 0  NERSS (Objs1 y P12) = 0  NERSS (Objs1 y P13) = 0</p>

Tabla 20. Aplicación de las métricas base (caso de estudio 1) (Continuación)

Nombre	Métrica	Valor
NERSS	Número de elementos con relación sintáctica o relación semántica entre un objetivo y un problema	<p>NERSS (Objs2 y P1) = 0  NERSS (Objs2 y P2) = 1  NERSS (Objs2 y P3) = 2  NERSS (Objs2 y P4) = 1  NERSS (Objs2 y P5) = 0  NERSS (Objs2 y P6) = 2  NERSS (Objs2 y P7) = 2  NERSS (Objs2 y P8) = 0  NERSS (Objs2 y P9) = 1  NERSS (Objs2 y P10) = 1  NERSS (Objs2 y P11) = 1  NERSS (Objs2 y P12) = 1  NERSS (Objs2 y P13) = 1</p> <p>NERSS (Objs3 y P1) = 0  NERSS (Objs3 y P2) = 2  NERSS (Objs3 y P3) = 2  NERSS (Objs3 y P4) = 2  NERSS (Objs3 y P5) = 0  NERSS (Objs3 y P6) = 1  NERSS (Objs3 y P7) = 2  NERSS (Objs3 y P8) = 0  NERSS (Objs3 y P9) = 1  NERSS (Objs3 y P10) = 1  NERSS (Objs3 y P11) = 1  NERSS (Objs3 y P12) = 1  NERSS (Objs3 y P13) = 1</p> <p>NERSS (Objs4 y P1) = 0  NERSS (Objs4 y P2) = 2  NERSS (Objs4 y P3) = 2  NERSS (Objs4 y P4) = 2  NERSS (Objs4 y P5) = 0  NERSS (Objs4 y P6) = 1  NERSS (Objs4 y P7) = 2  NERSS (Objs4 y P8) = 0  NERSS (Objs4 y P9) = 1  NERSS (Objs4 y P10) = 1  NERSS (Objs4 y P11) = 1  NERSS (Objs4 y P12) = 1  NERSS (Objs4 y P13) = 2</p>

Tabla 20. Aplicación de las métricas base (caso de estudio 1) (Continuación)

Nombre	Métrica	Valor
NERSS	Número de elementos con relación sintáctica o relación semántica entre un objetivo y un problema	NERSS (Objs5 y P1) = 0 NERSS (Objs5 y P2) = 2 NERSS (Objs5 y P3) = 2 NERSS (Objs5 y P4) = 1 NERSS (Objs5 y P5) = 0 NERSS (Objs5 y P6) = 3 NERSS (Objs5 y P7) = 6 NERSS (Objs5 y P8) = 0 NERSS (Objs5 y P9) = 1 NERSS (Objs5 y P10) = 3 NERSS (Objs5 y P11) = 2 NERSS (Objs5 y P12) = 2 NERSS (Objs5 y P13) = 2

Tabla 21. Aplicación de las métricas derivadas (caso de estudio 1)

Nombre	Métrica	Formula
NRSS	Nivel de relación sintáctica y semántica entre un objetivo y un problema	$NRSS = (NERSS * 100\%) / NEO$ NRSS (Obj1 y P1) = 33% NRSS (Obj1 y P2) = 0% NRSS (Obj1 y P3) = 0% NRSS (Obj1 y P4) = 0% NRSS (Obj1 y P5) = 0% NRSS (Obj1 y P6) = 33% NRSS (Obj1 y P7) = 33% NRSS (Obj1 y P8) = 33% NRSS (Obj1 y P9) = 33% NRSS (Obj1 y P10) = 33% NRSS (Obj1 y P11) = 33% NRSS (Obj1 y P12) = 0% NRSS (Obj1 y P13) = 0%

Tabla 21. Aplicación de las métricas derivadas (caso de estudio 1) (Continuación)

Nombre	Métrica	Formula
NRSS	Nivel de relación sintáctica y semántica entre un objetivo y un problema	NRSS (Obj2 y P1) = 0% NRSS (Obj2 y P2) = 50% NRSS (Obj2 y P3) = 50% NRSS (Obj2 y P4) = 25% NRSS (Obj2 y P5) = 0% NRSS (Obj2 y P6) = 25% NRSS (Obj2 y P7) = 25% NRSS (Obj2 y P8) = 0% NRSS (Obj2 y P9) = 25% NRSS (Obj2 y P10) = 25% NRSS (Obj2 y P11) = 25% NRSS (Obj2 y P12) = 0% NRSS (Obj2 y P13) = 0%  NRSS (Obj3 y P1) = 0% NRSS (Obj3 y P2) = 0% NRSS (Obj3 y P3) = 0% NRSS (Obj3 y P4) = 0% NRSS (Obj3 y P5) = 0% NRSS (Obj3 y P6) = 0% NRSS (Obj3 y P7) = 0% NRSS (Obj3 y P8) = 0% NRSS (Obj3 y P9) = 0% NRSS (Obj3 y P10) = 0% NRSS (Obj3 y P11) = 0% NRSS (Obj3 y P12) = 0% NRSS (Obj3 y P13) = 0%  NRSS (Obj4 y P1) = 0% NRSS (Obj4 y P2) = 33% NRSS (Obj4 y P3) = 33% NRSS (Obj4 y P4) = 33% NRSS (Obj4 y P5) = 0% NRSS (Obj4 y P6) = 66% NRSS (Obj4 y P7) = 66% NRSS (Obj4 y P8) = 0% NRSS (Obj4 y P9) = 33% NRSS (Obj4 y P10) = 66% NRSS (Obj4 y P11) = 33% NRSS (Obj4 y P12) = 0% NRSS (Obj4 y P13) = 33%

Tabla 21. Aplicación de las métricas derivadas (caso de estudio 1) (Continuación)

Nombre	Métrica	Formula
NRSS	Nivel de relación sintáctica y semántica entre un objetivo y un problema	NRSS (Obj5 y P1) = 0% NRSS (Obj5 y P2) = 0% NRSS (Obj5 y P3) = 0% NRSS (Obj5 y P4) = 0% NRSS (Obj5 y P5) = 0% NRSS (Obj5 y P6) = 0% NRSS (Obj5 y P7) = 0% NRSS (Obj5 y P8) = 100% NRSS (Obj5 y P9) = 0% NRSS (Obj5 y P10) = 0% NRSS (Obj5 y P11) = 0% NRSS (Obj5 y P12) = 0% NRSS (Obj5 y P13) = 0%  NRSS (Obj6 y P1) = 0% NRSS (Obj6 y P2) = 20% NRSS (Obj6 y P3) = 20% NRSS (Obj6 y P4) = 20% NRSS (Obj6 y P5) = 0% NRSS (Obj6 y P6) = 60% NRSS (Obj6 y P7) = 60% NRSS (Obj6 y P8) = 0% NRSS (Obj6 y P9) = 40% NRSS (Obj6 y P10) = 80% NRSS (Obj6 y P11) = 60% NRSS (Obj6 y P12) = 20% NRSS (Obj6 y P13) = 20%  NRSS (Obj7 y P1) = 0% NRSS (Obj7 y P2) = 20% NRSS (Obj7 y P3) = 20% NRSS (Obj7 y P4) = 20% NRSS (Obj7 y P5) = 0% NRSS (Obj7 y P6) = 40% NRSS (Obj7 y P7) = 40% NRSS (Obj7 y P8) = 0% NRSS (Obj7 y P9) = 40% NRSS (Obj7 y P10) = 40% NRSS (Obj7 y P11) = 80% NRSS (Obj7 y P12) = 20% NRSS (Obj7 y P13) = 20%

Tabla 21. Aplicación de las métricas derivadas (caso de estudio 1) (Continuación)

Nombre	Métrica	Formula
NRSS	Nivel de relación sintáctica y semántica entre un objetivo y un problema	<p>NRSS (Obj8 y P1) = 0%</p> <p>NRSS (Obj8 y P2) = 16%</p> <p>NRSS (Obj8 y P3) = 16%</p> <p>NRSS (Obj8 y P4) = 16%</p> <p>NRSS (Obj8 y P5) = 0%</p> <p>NRSS (Obj8 y P6) = 0%</p> <p>NRSS (Obj8 y P7) = 16%</p> <p>NRSS (Obj8 y P8) = 0%</p> <p>NRSS (Obj8 y P9) = 16%</p> <p>NRSS (Obj8 y P10) = 16%</p> <p>NRSS (Obj8 y P11) = 16%</p> <p>NRSS (Obj8 y P12) = 50%</p> <p>NRSS (Obj8 y P13) = 50%</p> <p>NRSS (Obj9 y P1) = 0%</p> <p>NRSS (Obj9 y P2) = 16%</p> <p>NRSS (Obj9 y P3) = 33%</p> <p>NRSS (Obj9 y P4) = 33%</p> <p>NRSS (Obj9 y P5) = 16%</p> <p>NRSS (Obj9 y P6) = 16%</p> <p>NRSS (Obj9 y P7) = 50%</p> <p>NRSS (Obj9 y P8) = 0%</p> <p>NRSS (Obj9 y P9) = 33%</p> <p>NRSS (Obj9 y P10) = 33%</p> <p>NRSS (Obj9 y P11) = 33%</p> <p>NRSS (Obj9 y P12) = 66%</p> <p>NRSS (Obj9 y P13) = 50%</p> <p>NRSS (Objs1 y P1) = 50%</p> <p>NRSS (Objs1 y P2) = 0%</p> <p>NRSS (Objs1 y P3) = 0%</p> <p>NRSS (Objs1 y P4) = 0%</p> <p>NRSS (Objs1 y P5) = 0%</p> <p>NRSS (Objs1 y P6) = 0%</p> <p>NRSS (Objs1 y P7) = 0%</p> <p>NRSS (Objs1 y P8) = 0%</p> <p>NRSS (Objs1 y P9) = 0%</p> <p>NRSS (Objs1 y P10) = 0%</p> <p>NRSS (Objs1 y P11) = 0%</p> <p>NRSS (Objs1 y P12) = 0%</p> <p>NRSS (Objs1 y P13) = 0%</p>

Tabla 21. Aplicación de las métricas derivadas (caso de estudio 1) (Continuación)

Nombre	Métrica	Formula
NRSS	Nivel de relación sintáctica y semántica entre un objetivo y un problema	<p>NRSS (Objs2 y P1) = 0%</p> <p>NRSS (Objs2 y P2) = 25%</p> <p>NRSS (Objs2 y P3) = 50%</p> <p>NRSS (Objs2 y P4) = 25%</p> <p>NRSS (Objs2 y P5) = 0%</p> <p>NRSS (Objs2 y P6) = 50%</p> <p>NRSS (Objs2 y P7) = 50%</p> <p>NRSS (Objs2 y P8) = 0%</p> <p>NRSS (Objs2 y P9) = 25%</p> <p>NRSS (Objs2 y P10) = 25%</p> <p>NRSS (Objs2 y P11) = 25%</p> <p>NRSS (Objs2 y P12) = 25%</p> <p>NRSS (Objs2 y P13) = 25%</p> <p>NRSS (Objs3 y P1) = 0%</p> <p>NRSS (Objs3 y P2) = 50%</p> <p>NRSS (Objs3 y P3) = 50%</p> <p>NRSS (Objs3 y P4) = 50%</p> <p>NRSS (Objs3 y P5) = 50%</p> <p>NRSS (Objs3 y P6) = 25%</p> <p>NRSS (Objs3 y P7) = 50%</p> <p>NRSS (Objs3 y P8) = 0%</p> <p>NRSS (Objs3 y P9) = 25%</p> <p>NRSS (Objs3 y P10) = 25%</p> <p>NRSS (Objs3 y P11) = 25%</p> <p>NRSS (Objs3 y P12) = 25%</p> <p>NRSS (Objs3 y P13) = 25%</p> <p>NRSS (Objs4 y P1) = 0%</p> <p>NRSS (Objs4 y P2) = 50%</p> <p>NRSS (Objs4 y P3) = 50%</p> <p>NRSS (Objs4 y P4) = 50%</p> <p>NRSS (Objs4 y P5) = 0%</p> <p>NRSS (Objs4 y P6) = 25%</p> <p>NRSS (Objs4 y P7) = 50%</p> <p>NRSS (Objs4 y P8) = 0%</p> <p>NRSS (Objs4 y P9) = 25%</p> <p>NRSS (Objs4 y P10) = 25%</p> <p>NRSS (Objs4 y P11) = 25%</p> <p>NRSS (Objs4 y P12) = 25%</p> <p>NRSS (Objs4 y P13) = 50%</p>

Tabla 21. Aplicación de las métricas derivadas (caso de estudio 1) (Continuación)

Nombre	Métrica	Formula
NRSS	Nivel de relación sintáctica y semántica entre un objetivo y un problema	NRSS (Objs5 y P1) = 0% NRSS (Objs5 y P2) = 25% NRSS (Objs5 y P3) = 25% NRSS (Objs5 y P4) = 12.5% NRSS (Objs5 y P5) = 0% NRSS (Objs5 y P6) = 37.5% NRSS (Objs5 y P7) = 75% NRSS (Objs5 y P8) = 0% NRSS (Objs5 y P9) = 12.5% NRSS (Objs5 y P10) = 37.5% NRSS (Objs5 y P11) = 25% NRSS (Objs5 y P12) = 25% NRSS (Objs5 y P13) = 25%
CPRO	Cantidad de problemas que tiene relación con los objetivos	9
CORP	Cantidad de objetivos del sistema que tienen relación con los problemas del dominio	5
NCTM	Nivel de consistencia y trazabilidad del modelo	$NCTM = ((9 + 5) * 100\%) / (5 + 13)$  <b>77.7% de consistencia y trazabilidad del modelo</b>

Se anexa carta de la Empresa IDE Sistemas Ltda. Donde se llevó a cabo la validación del modelo en un caso real.

## 6.2 CASO DE ESTUDIO NRO. 2

Especificación y formalización de los problemas a partir de los objetivos de la investigación planteados en la propuesta doctoral "A Method based on organizational patterns and key performance indicators for translating organizational objectives into source code". Para este caso de estudio se aplican las etapas 1, 2 y 3 del Modelo.



**Etapas nro. 1** Especificación de objetivos organizacionales (del proyecto), utilizando las estructuras sintácticas y semánticas definidas. En la Tabla 22 se visualizan los objetivos propuestos y los especificados utilizando las reglas.

Tabla 22. Especificación de Objetivos (caso de estudio 2)

Objetivos propuestos	Objetivos utilizando las estructuras
Definition of a method based on organizational patterns for translating organizational objectives into source code by using key performance indicators	Achieving a defined method based on organizational patterns  Achieving the organizational objectives are correctly translated  Achieving key performance indicators are correctly used
Identifying a syntactic and semantic structure for representing and formalizing key performance indicators	Achieving the analyst identifies a syntactic structure  Achieving the analyst identifies a semantic structure  Achieving the analyst represents key performance indicators  Achieving the analyst formalizes key performance indicators
Describing a structure for representing source code generated from key performance indicators	Achieving the analyst describes a structure for representing source code efficiently  Ensuring the analyst generates source code from key performance indicators
Providing consistent and traceable relationships among organizational objectives, key performance indicators, and source code generated from key performance indicators.	Achieving the consistent and traceable relationships are provided

Tabla 22. Especificación de Objetivos (caso de estudio 2) (Continuación)

Objetivos propuestos	Objetivos utilizando las estructuras
Designing a pattern-based framework that facilitates knowledge reusability related to translating organizational objectives into source code by using key performance indicators.	Achieving the design of a pattern-based framework
Providing a KPI model traceable with the organizational objectives and suitable to automated generation of source code.	Achieving a key performance indicator model
Validating the proposed method by using a case study.	Ensuring validation of the method

En las Figuras 21 y 22 se visualiza el diagrama de objetivos. Para su mejor visualización se separa en dos figuras.

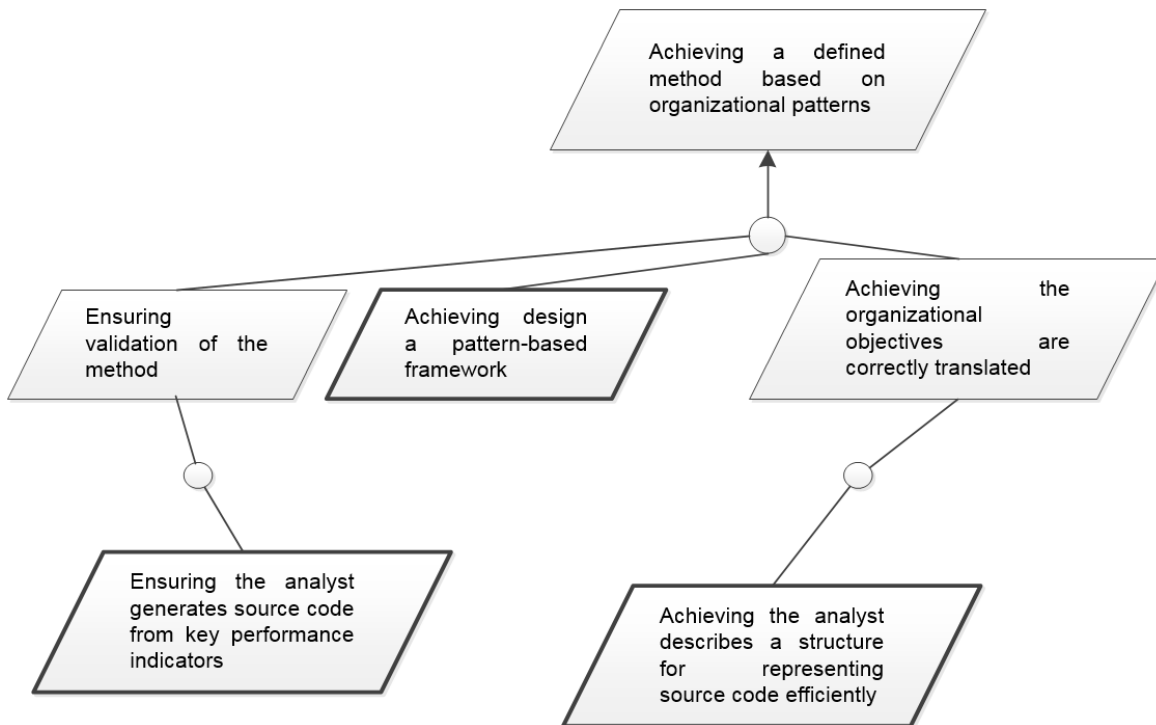


Figura 21. Diagrama de objetivos (caso de estudio 2), parte 1



**Etapá nro. 3:** Especificación de problemas que se detectan a partir de los objetivos definidos. En la Tabla 23 se visualizan los problemas.

Tabla 23. Especificación de problemas (caso de estudio 2)

<b>Objetivos</b>	<b>Problemas especificados</b>
Achieving a defined method based on organizational patterns	The analyst does not define the method correctly.
Achieving the organizational objectives are correctly translated	The analyst does not translate the organizational objectives.
Achieving key performance indicators are correctly used	The use key performance indicator is poor
Achieving the analyst identifies a syntactic structure	The key performance indicators does not have syntactic structure
Achieving the analyst identifies a semantic structure	The key performance indicators does not have semantic structure
Achieving the analyst represents key performance indicators	The analyst does not adequately represent key performance indicators
Achieving the analyst formalizes key performance indicators	The formalization of key performance indicator is poor
Achieving the analyst describes a structure for representing source code efficiently	The analyst incorrectly describes structures
Ensuring the analyst generates source code from key performance indicators	The key performance indicator is not available
Achieving the consistent and traceable relationships are provided	The consistent and traceable relationships are poor
Achieving design a pattern-based framework	The pattern-based framework design is no met
Achieving a key performance indicator model	The model validation has mistakes
Ensuring validation of the method	

### 6.3 CASO DE ESTUDIO NRO. 3

En Vargas et al (2015) se describe un caso de estudio como un ejemplo de aplicación del modelo. El caso de estudio es el siguiente: “El centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) desea automatizar la oficina de almacén e inventario

del Departamento de recursos materiales y servicios. Este Departamento es el responsable por la planeación, coordinación y evaluación de todas las actividades relacionadas con la gestión de recursos materiales, la prestación de servicios generales y de mantenimiento del CENIDET. Las dependencias que integran el Departamento son: La oficina de compras, la oficina de almacén e inventario y la oficina de servicios generales.

El caso de estudio se centra en la oficina de almacén e inventario, que es la responsable del monitoreo y el registro de las entradas y las salidas del inventario a partir del stock disponible, así como de llevar periódicamente el control del inventario.

La Oficina de almacén e inventario lleva a cabo las siguientes actividades: registro del inventario, inventario de las compras realizadas, almacenamiento de los productos consumibles y asignación y cancelación del inventario al personal del centro de investigación. El caso de estudio se centra en el proceso de registro y cancelación de inventarios.

La oficina de almacén e inventario pretende la construcción o adquisición de un sistema de registro del inventario, ya sea utilizando tecnología de código de barras, identificación por radio frecuencia (RFID) o un software de registro manual de productos”.

La aplicación total del modelo se puede visualizar en Vargas et al. (2015).

## **6.4 PRODUCTOS DE NUEVO CONOCIMIENTO GENERADOS A PARTIR DE LA TESIS DOCTORAL**

A partir de la Tesis Doctoral que se presenta, se generaron varios productos de nuevo conocimiento que se alinean con los objetivos planteados en la Tesis. Los productos incluyen artículos en revistas indexadas nacionales e internacionales y capítulos de libros de editoriales nacionales. En la Tabla 24 se muestra una relación de dichos productos.

Tabla 24. Productos de nuevo conocimiento generados

Nr o.	Título de la Publicación	Tipo de Publicación	Revista / Libro	Autores	Clasificación
1	Specification of problems from the business goals in the context of early software requirements elicitation	Artículo Científico	Journal of the Facultad de Minas, Universidad Nacional de Colombia (DYNA) ISSN 0012-7353 y ISSN 2346-2183 (2014)	Carlos Mario Zapata y Fabio Alberto Vargas A	A1 (Publindex), SCOPUS, JCR, Latindex, RedALyc
2	Determinación de objetivos de mejoramiento a partir de problemas del diagrama causa-efecto en el contexto del UNC-METHOD	Capítulo de Libro	Libro Ingeniería del software e ingeniería del conocimiento: Dos disciplinas interrelacionadas (2014)	Carlos Mario Zapata y Fabio Alberto Vargas A	Editorial Universidad Medellín, Libro de Investigación
3	Método de consistencia en la relación entre problemas y objetivos para el proceso de elicitation de requisitos	Artículo Científico	Revista Cuaderno Activa, Tecnológico de Antioquia - Institución Universitaria ISSN 2027-8101 (2012)	Fabio Alberto Vargas A	Índice E-Revistas
4	Reglas Sintáctico-semánticas para Relacionar los Objetivos Organizacionales y los Problemas en el Contexto de la Educación Temprana de Requisitos de Software	Artículo Científico	Revista Latinoamericana de Ingeniería de Software, ISSN 2314-2642 (2013)	Carlos Mario Zapata y Fabio Alberto Vargas A	E-Revistas, DOAJ, Latindex, CiteFactor
5	Innovación en el diseño y evaluación de proyectos: establecimiento de las relaciones lingüísticas entre Objetivos y Problemas	Artículo Científico	Revista Lámpsakos ISSN: 2145-4086 ed: Fondo Editorial Fundación Universitaria Luis Amigo (2012)	Carlos Mario Zapata y Fabio Alberto Vargas A	Categoría C (Publindex)

Tabla 24. Productos de nuevo conocimiento generados (Continuación)

Nr o.	Título de la Publicación	Tipo de Publicación	Revista / Libro	Autores	Clasificación
6	GBRAM from a SEMAT Perspective	Capítulo de Libro	Libro: SOFTWARE ENGINEERING: METHODS, MODELING, AND TEACHING. En: Colombia ISBN: 78-958-775-080-5 (2014)	Carlos Mario Zapata y Fabio Alberto Vargas A	Centro De Publicaciones Universidad Nacional De Colombia Sede Medellín
7	Pasantía Investigativa Centro Nacional de Desarrollo para Investigación de México e INFOTEC	Pasantía Investigativa	CENIDET e INFOTEC	Fabio Alberto Vargas A	No aplica
8	Método para especificar y formalizar requisitos a partir de problemas del dominio en el contexto de educación de requisitos de software	Artículo Científico	Revista Iberoamericana de Automática e Informática industrial. ISSN 1697-7920 (Enviado y en revisión). 2015	Fabio Alberto Vargas, Carlos Mario Zapata, Hugo Estrada Esquivel, Alicia Martínez	A2 (Publindex), JCR, SCOPUS e ISI
9	Formalization of domain problems described in the cause-and-effect diagram in the context of the software development process	Artículo Científico	Revista Facultad de Ingeniería Universidad de Antioquia, ISSN 0120-6230 (Enviado y en revisión). 2015	Carlos Mario Zapata y Fabio Alberto Vargas A	A1 (Publindex), SCOPUS, e ISI

A continuación se realiza un resumen de dichos productos relacionados en la tabla anterior.

### Artículos

<b>Título:</b>	Specification of problems from the business goals in the context of early software requirements elicitation
<b>Autores:</b>	Carlos Mario Zapata y Fabio Alberto Vargas A
<b>Revista:</b>	DYNA 81 (184), pp. 72-80. April, 2014 Medellín. ISSN 0012-7353 Printed, ISSN 2346-2183 Online
<b>Indexación:</b>	A1 de Publindex, SCOPUS, JCR, Latindex, RedALyc
<b>Estado:</b>	Publicado
<b>Año:</b>	2014

<b>Resumen:</b>	One of the main activities of the early elicitation of software requirements is the recognition and specification of organizational problems. Such activity is intended to allow for an initial requirements definition and the fulfillment of the stakeholder needs. Such problems can be directly traced to the organizational goals for achieving contextualized software applications and alignment with the organizational <i>raison d'être</i> . In current elicitation methods based on goals and problems, the relationships are detected by the analyst and the stakeholder by using their experience and knowledge. However, traceability among goals and problems is still not achieved. In this paper we propose a method for specifying problems based on business goals. This method is composed by a set of semantic and syntactic rules used by the analyst for expressing the problem from the goal statements. Also, we present a laboratory example based on a KAOS goal diagram.
<b>Palabras clave:</b>	Business goals; problems; semantic rules; syntactic rules

<b>Título:</b>	Método de consistencia en la relación entre problemas y objetivos para el proceso de elicitación de requisitos
<b>Autores:</b>	Fabio Alberto Vargas A
<b>Revista:</b>	Revista Cuaderno Activa, Tecnológico de Antioquia, Institución Universitaria ISSN 2027-8101 (2012)
<b>Indexación:</b>	E-Revistas, en proceso de indexación en EBSCO y en Publindex
<b>Estado:</b>	Publicado
<b>Año:</b>	2014
<b>Resumen:</b>	La captura de requisitos es un proceso manual que lleva a cabo el analista con base en su experiencia e interpretación. En este proceso, la definición de los problemas por solucionar y su relación con los objetivos de la organización se realizan sin seguir unas pautas que garanticen la consistencia. En muchos casos, esto trae consigo problemas posteriores en el ciclo de vida del software. Métodos de ingeniería de software utilizan el diagrama de objetivos de KAOS y el diagrama causa-efecto para representar objetivos y problemas, pero siguen siendo una tarea de interpretación del analista, sin tener en cuenta métodos de consistencia para su representación. El artículo aborda la generación de un método que establezca de forma automática y consistente la relación entre objetivos de la organización y los problemas que se detectan en el proceso de educación de requisitos, además de presentar un conjunto de estructuras lingüísticas para la representación de problemas y objetivos.
<b>Palabras claves:</b>	Estructura de problemas, estructura de objetivos, relaciones de consistencia, estructuras gramaticales, educación de requisitos.



<b>Título:</b>	Reglas Sintáctico-semánticas para Relacionar los Objetivos Organizacionales y los Problemas en el Contexto de la Educación Temprana de Requisitos de Software
<b>Autores:</b>	Carlos Mario Zapata y Fabio Alberto Vargas A
<b>Revista:</b>	Revista Latinoamericana de Ingeniería de Software, ISSN 2314-2642
<b>Indexación:</b>	E-Revistas, DOAJ, Latindex, CiteFactor
<b>Estado:</b>	Publicado
<b>Año:</b>	2013
<b>Resumen:</b>	Los procesos de ingeniería de software de alta calidad requieren la educación temprana de requisitos funcionales y no funcionales. Este proceso lo llevan a cabo los analistas y los interesados, tratando de solucionar los problemas de información de la organización y contrastándolos con el contexto del negocio, representado en sus objetivos organizacionales. Este proceso se suele realizar de forma manual y, si bien existen algunos trabajos que establecen una relación entre los problemas y los objetivos, se basan en la negación total de unos para obtener los otros. Esto conduce al desarrollo de aplicaciones de software no pertinentes para la organización, que no resuelven sus problemas prioritarios o que no se alinean con sus objetivos organizacionales. Por estas razones, en este artículo se propone una relación entre los problemas a solucionar y los objetivos organizacionales, empleando un conjunto de reglas sintáctico-semánticas que validen dicha relación y que se reflejen en requisitos consistentes y contextualizados con la organización. Estas reglas se validan con un caso de estudio
<b>Palabras claves:</b>	Objetivos Organizacionales, problemas, educación temprana de requisitos, reglas sintácticas, reglas semánticas
<b>Título:</b>	Innovación en el diseño y evaluación de proyectos: establecimiento de las relaciones lingüísticas entre objetivos y problemas.
<b>Autores:</b>	Carlos Mario Zapata Jaramillo, Fabio Alberto Vargas A
<b>Revista:</b>	Revista Lámpsakos ISSN: 2145-4086 ed: Fondo Editorial Fundación Universitaria Luis Amigo.
<b>Indexación:</b>	Categoría C (Publindex)
<b>Estado:</b>	Publicado
<b>Año:</b>	2012
<b>Resumen:</b>	El lenguaje desempeña un papel importante en el diseño y evaluación de proyectos, especialmente cuando se trata de identificar y enunciar problemas y objetivos. Aunque no se propone formalmente, en algunos ámbitos se reconoce que existen estructuras para expresar objetivos y problemas ligados con una organización. Sin embargo, no se suele plantear una relación directa entre ellas o, si se requiere, se plantean de forma

	intuitiva. Esta relación es importante porque permite a los miembros de la organización plantear soluciones a los problemas que se ligen con sus objetivos estratégicos. Por ello, en este artículo se presenta una propuesta innovadora para establecer las relaciones lingüísticas entre los problemas de una organización y sus objetivos. Además se muestran algunos ejemplos.
<b>Palabras claves:</b>	Objetivos organizacionales, problemas, Relaciones lingüísticas, Análisis organizacional

<b>Título:</b>	Formalization of domain problems described in the cause-and-effect diagram in the context of the software development process
<b>Autores:</b>	Carlos Mario Zapata Jaramillo y Fabio Alberto Vargas A
<b>Revista:</b>	Revista Facultad de Ingeniería Universidad de Antioquia, ISSN 0120-6230
<b>Indexación:</b>	A1 (Publindex), SCOPUS, e ISI
<b>Estado:</b>	Evaluación
<b>Año:</b>	2015
<b>Resumen:</b>	Identifying and specifying problems in the early requirements elicitation process is the direct responsibility of the system analyst. During the domain analysis, the analyst collects and uses problems as a main source for the requirements specification. Problems are described in domain discourses and are represented in diagrams such as cause-and-effect diagrams and problem trees, among others. After that, the analyst should relate them to the system and the stakeholder goals. The problem representation should be difficult to understand for stakeholders, since problems are often poorly described or inadequately expressed. In addition, the stakeholder misunderstanding leads to another set of gaps in the software requirements elicitation, such as low traceability and consistency among requirements, organizational goals, and domain problems. In this paper, we propose a formalization of the problems described in the cause-and-effect diagram based on the so-called pre-conceptual schemas. By using such schemas, both the analyst and the stakeholder can recognize how problems are related to the domain for which a software application is being developed. We also apply such formalization by using a case study article style.
<b>Palabras claves:</b>	Domain, problems, cause-and-effect diagram, formalization, preconceptual schemas

<b>Título:</b>	Método para especificar y formalizar requisitos a partir de problemas del dominio en el contexto de educación de requisitos de software
<b>Autores:</b>	Fabio Alberto Vargas Agudelo, Carlos Mario Zapata Jaramillo, Alicia Martínez Rebollar y Hugo Estrada Esquivel.
<b>Revista:</b>	Revista Iberoamericana de Automática e Informática industrial. ISSN 1697-7920 2015

<b>Indexación:</b>	A2 (Publindex), JCR, SCOPUS e ISI
<b>Estado:</b>	En evaluación
<b>Año:</b>	2015
<b>Resumen:</b>	El Análisis Organizacional, al interior de la Ingeniería de requisitos, requiere un alto grado de consistencia en la información recopilada y una validación permanente de la misma entre el analista y el interesado. Uno de los insumos más importantes dentro de esta etapa es la definición de los requisitos, es decir los objetivos que el sistema de información debe cumplir, teniendo en cuenta los problemas detectados en el dominio de aplicación. Esta tarea la lleva a cabo el analista con base en su experiencia, pero sin que existan formas de verificar la consistencia de la información. Por ello, en este artículo se define un método para especificar y formalizar los requisitos a partir de los problemas del dominio detectados en la fase de educación de requisitos de software. El método propuesto se aplica en un ejemplo
<b>Palabras claves:</b>	Requisitos, problemas del dominio, consistencia, trazabilidad, objetivo del sistema

### Capítulos de Libro

<b>Título:</b>	Determinación de objetivos de mejoramiento a partir de problemas del diagrama causa - efecto en el contexto del UNC-METHOD
<b>Autores:</b>	Carlos Mario Zapata Jaramillo y Fabio Alberto Vargas A
<b>Libro:</b>	Ingeniería de software e ingeniería de conocimiento dos disciplinas interrelacionadas. En: Colombia ISBN: 978-958-8815-31-2
<b>Editorial:</b>	Universidad de Medellin
<b>Estado:</b>	Publicado
<b>Año:</b>	2014
<b>Resumen:</b>	<p>El desarrollo de software requiere, en sus etapas iniciales, un alto nivel de consistencia en la información que se genera tomando como base la comunicación entre el analista y el interesado y, además, el análisis organizacional que se realice.</p> <p>UNC-Method es un método de desarrollo de software que se basa en la definición de requisitos, apoyando al analista en la identificación y la especificación de problemas y objetivos del interesado. En metodologías de análisis organizacional, como Kepner-Tregoe y Marco Lógico, se intenta establecer una relación entre los objetivos y los problemas, pero con una alta participación del analista, quien realiza este proceso de forma manual. Por ello, en este artículo se define un método para la determinación de objetivos del diagrama de objetivos a partir de los problemas representados en el diagrama causa-efecto. Ambos diagramas se emplean en el contexto del UNC-Method para realizar el análisis del problema. El método de generación se limita a los objetivos de mejoramiento, dado que estos suelen constituir los objetivos de más alto nivel en las organizaciones, y emplea los esquemas preconceptuales como representaciones intermedias que permiten ligar conceptos del dominio. Todo esto se ejemplifica con un caso de estudio.</p>

<b>Palabras claves:</b>	Objetivos; UNC-Method; causa-efecto; consistencia
<b>Título:</b>	GBRAM from a SEMAT Perspective
<b>Autores:</b>	Carlos Mario Zapata, Fabio Alberto Vargas y Luis Fernando Castro
<b>Libro:</b>	SOFTWARE ENGINEERING: METHODS, MODELING, AND TEACHING. En: Colombia ISBN: 78-958-775-080-5
<b>Editorial:</b>	Centro de Publicaciones Universidad Nacional de Colombia
<b>Estado:</b>	Publicado
<b>Año:</b>	2014
<b>Resumen:</b>	<p>Software engineering has been evolving towards the standardization of processes and generating a common core of elements. Such a core could provide analysts and stakeholders with the tools—<i>e.g.</i>, methods, practices, etc.—for improving several aspects of the software development process. Standardized processes are useful for recognizing and establishing conditions that guarantee the relevance, quality, safety, efficiency, performance, and maintenance of a software application, regardless the software platform or the environment used (Johnson <i>et al.</i>, 2012).</p> <p>The SEMAT kernel supports the modern practice representation of methods like Scrum, XP, RUP, and CDM (Jacobson <i>et al.</i>, 2013). A set of elements is defined in order to collect information from the software engineering process and control the activities performed during the software development process—<i>e.g.</i> stakeholder management, requirements elicitation, and software system development, among others.</p>
<b>Palabras claves:</b>	Goals, SEMAT.

# 7.CONCLUSIONES y TRABAJO FUTURO

## 7.1 Conclusiones

Dominios como la ingeniería de software y el análisis organizacional utilizan, en muchos de sus procesos, la representación y especificación de problemas y objetivos y la relación entre ellos, a fin de determinar qué objetivos se derivan de un problema definido y viceversa. La adecuada determinación de objetivos del sistema a partir de los problemas encontrados, ayuda en la especificación temprana de requisitos de software consistentes y a una mayor trazabilidad.

La literatura especializada no plantea dentro de las metodologías de ingeniería de requisitos orientadas a objetivos y de análisis organizacional, métodos automáticos que permitan relacionar objetivos y problemas, sigue siendo un trabajo que depende del analista y de interesado, sin que medien procesos que garanticen una consistencia en la información.

Los métodos de ingeniería de requisitos orientados a objetivos no referencian de forma explícita los problemas de dominio, dificultado, en muchos casos, la especificación de requisitos contextualizados y pertinentes con la organización.

La especificación de problemas y objetivos por parte de analistas e interesados no se apoya en estructuras gramaticales que garanticen que lo que se enuncie sea realmente un problema o un objetivo y que además sea muy claro para los interesados.

Como una forma de solución a la problemática detectada, en esta Tesis se propuso, diseñó e implementó un modelo de especificación de requisitos que contiene los siguientes aportes:

- Estructuras lingüísticas (Sintáctico-Semánticas) para problemas y objetivos y la forma de relacionarlos que facilita la tarea del analista en los diferentes

procesos de ingeniería de requisitos y de análisis organizacional, estableciendo la consistencia y trazabilidad que existe entre objetivos y problemas.

- Vinculación de los problemas de dominio de forma explícita a los procesos de ingeniería de requisitos dentro de la etapa de análisis organizacional que garanticen aplicativos de software útiles y que realmente contribuyan a la solución de la problemática detectada.
- La definición de una forma consistente para identificar y estructurar problemas y objetivos que se puede utilizar en áreas diferentes a la ingeniería de software. Como por ejemplo en los procesos de análisis organizacional y los procesos de formulación y evaluación de proyectos.
- La generación de soluciones automatizadas que apoyen a los analistas de sistemas y a los interesados en su proceso de educación de requisitos de software, generando un mayor grado de confiabilidad en las soluciones de software planteadas, ya que éstas se alinearán con el contexto de la organización y serán pertinentes a las necesidades de la misma.
- La vinculación de los objetivos organizacionales garantizando una especificación más consistente de requisitos de software.
- La formalización de los problemas y objetivos mediante esquemas preconceptuales que garanticen un mayor entendimiento de los mismos para el analista y el interesado.
- La representación del dominio donde se visualizan gráficamente los objetivos y los problemas del dominio detectados por el analista y el interesado.
- Un conjunto de métricas base y derivadas que permiten validar la consistencia y trazabilidad del modelo y que se apliquen a un caso de estudio real.
- La aplicación del modelo en un caso real de la ingeniería de software, que permitió al analista de sistemas y a los interesados justificar la necesidad del software desarrollado y realizar una trazabilidad de los elementos más importantes y prioritarios a tener en cuenta en la solución de la problemática.
- Un conjunto de productos de nuevo conocimiento (artículos y capítulos de libro) como medios de socialización de la Tesis en la comunidad académica y científica.

Todos estos aportes y resultados pretenden automatizar el proceso de análisis organizacional dentro de la fase de educación de requisitos de software; aumentar la consistencia y trazabilidad de los requisitos de software; contribuir a la construcción de software pertinente y contextualizado con la organización; un mayor entendimiento y comprensión del negocio por parte del analista y el interesado.

Es importante destacar que el modelo propuesto se puede aplicar de acuerdo al contexto sólo con algunas etapas. Esto depende de la necesidad del interesado. Por ejemplo, si sólo se requiere la especificación de los problemas se aplican las etapas 1, 2 y 3, como se utilizó en el caso de estudio nro. 2.

Las métricas definidas son aplicables a la totalidad del modelo. Es decir en aquellos casos donde se llevan a cabo las cinco etapas del modelo propuesto.

Las principales dificultades encontradas durante la ejecución de la Tesis y la aplicación del modelo fueron: la falta de experiencia que tienen los analistas de sistemas en procesos de análisis organizacional y en el reconocimiento e identificación de problemas del dominio; la carencia de métodos formales que apoyen la representación de objetivos y problemas; la poca importancia que se le da al análisis organizacional en proyectos de ingeniería de software; la diversidad de metodologías e instrumentos utilizados en los procesos de educación de requisitos; la falta de documentación que se aplica en muchos proyectos de desarrollo de software que dificulta el mantenimiento y actualización.

## 7.2 Trabajo Futuro

Teniendo en cuenta el desarrollo de la Tesis y el proceso de revisión y análisis realizado en la misma, se proponen algunas líneas de trabajo futuro en el área, que pueden contribuir a mejorar el modelo que se presenta y además favorecer al área de la ingeniería de requisitos:

- La automatización del proceso mediante la implementación de una aplicación de software que relacione directamente los objetivos y los problemas utilizando las estructuras definidas y que genere las especificaciones y formalizaciones.

- La formulación de un conjunto de estructuras para la especificación de requisitos funcionales del software a partir de los objetivos del sistema propuestos en el modelo que garanticen trazabilidad y consistencia.
- El desarrollo de ontologías del dominio de un problema que faciliten la relación de términos del dominio en la especificación de objetivos y problemas y que faciliten una mejor relación sintáctica y semántica.
- La definición de otras relaciones entre objetivos organizacionales, problemas del dominio y objetivos del sistema que mejoren la trazabilidad.
- La formulación de otras relaciones semánticas y sintácticas entre objetivos y problemas del dominio.
- La formalización de objetivos y problemas y la representación del dominio mediante diferentes diagramas utilizados en métodos tradicionales de desarrollo de software como pueden ser: modelos del dominio, diagramas de clases, modelos entidad-relación, redes semánticas y ontologías, entre otros.
- El incremento del conjunto de reglas que facilita la trazabilidad de los problemas del dominio, objetivos organizacionales y del sistema con diferentes actores de los procesos de análisis organizacional y de análisis de requisitos.
- El análisis de elementos pragmáticos que permitan establecer la relación entre objetivos y problemas. Por ejemplo, para el caso de estudio podría ser intuitivamente claro que el problema “no todos los álbumes se pueden acceder” ataca directamente el objetivo “incrementar los usuarios”, pero sus frases no tienen elementos comunes ni estructuras que permitan indicar la relación que puede existir entre el objetivo y el problema. Así, se requieren mecanismos adicionales que contribuyan a definir ese nexo.



- La construcción de una herramienta de análisis que esté en capacidad de determinar la relación entre los objetivos y problemas y luego trazar los diagramas de manera consistente.
- Definir métricas aplicables a resultados parciales del modelo. Es decir a aplicaciones del modelo donde no se ejecuten todas las etapas.
- Ampliar y Automatizar las métricas de consistencia y trazabilidad del modelo propuesto para hacerlas más prácticas y ágiles para analistas interesados.

# A. Anexo: Certificado de Caso de Estudio

Visitenos en: [www.idesistemas.co](http://www.idesistemas.co)



Medellín, Mayo 19 de 2015

## IDE SISTEMAS CERTIFICA:

Que en el proceso de documentación del Software SINCO (Software de calificaciones y administración de gestión académica) durante la etapa de levantamiento de requisitos, se aplicó el modelo "Modelo para la Especificación de requisitos iniciales de software a partir de la relación sintáctica y semántica entre objetivos y problemas", para el reconocimiento organizacional, la detección de problemas del dominio y la especificación inicial de los requisitos del sistema.

El modelo aplicado nos permitió tener un alto reconocimiento del dominio de la aplicación, así como poder contar con requisitos de software contextualizados y pertinentes para la organización. También es destacable que el modelo aplicado permitió una trazabilidad hacia atrás del software implementado durante el proceso de actualización y modificación de requisitos.

Esperamos incorporar el modelo en futuros proyectos que adelante la empresa, como herramienta de análisis organizacional y de educación de requisitos de software. Este modelo podrá ser utilizado para nuevos proyectos y también para la documentación de proyectos ya existentes.

Atentamente,

Anderson Rojas Sánchez  
Gerente  
[anderson@idesistemas.com.co](mailto:anderson@idesistemas.com.co)  
[www.idesistemas.co](http://www.idesistemas.co)

## Referencias Bibliográficas

Adriano, N. (2006). Comparación del Proceso de Elicitación de Requerimientos en el desarrollo de Software a Medida y Empaquetado. Propuesta de métricas para la Elicitación (Doctoral dissertation, Tesis de M. Sc. Universidad Nacional de la Plata).

Aguilar, J. A., Garrigós, I., & Mazón, J. N. (2011). A goal-oriented approach for optimizing non-functional requirements in web applications. In *Advances in Conceptual Modeling. Recent Developments and New Directions* (pp. 14-23). Springer Berlin Heidelberg.

Ali, R., Dalpiaz, F., & Giorgini, P. (2010). A goal-based framework for contextual requirements modeling and analysis. *Requirements Engineering*, 15(4), 439-458.

Alrajeh, D., Russo, A., Lockerbie, J., Maiden, N., Mavin, A., & Novak, M. (2013). Computational alignment of goals and scenarios for complex systems. In *Proceedings of the 2013 International Conference on Software Engineering* (pp. 1249-1252).

Antón, A. I., & Potts, C. (1998). The use of goals to surface requirements for evolving systems. In *Proceedings of the 1998 International Conference on Software Engineering* (pp. 157-166).

Burnay, C., Jureta, I. J., Linden, I., & Faulkner, S. (2014). A framework for the operationalization of monitoring in business intelligence requirements engineering. *Software & Systems Modeling*, 13, 1-22.

Camacho, H; Cámara, Luis; Cascante, R y Sainz, H (2001): El Enfoque del Marco Lógico: 10 casos prácticos. ADC-CIDEAL, Madrid.

Castro, J., Kolp, M., & Mylopoulos, J. (2002). Towards requirements-driven information systems engineering: the Tropos project. *Information systems*, 27(6), 365-389.

Dardenne, A., Van Lamsweerde, A., & Fickas, S. (1993). Goal-directed requirements acquisition. *Science of computer programming*, 20(1), 3-50.

Chaves, M. A. (2011). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes*, 6(10), 1-13.

Choi, S., Park, S., & Sugumaran, V. (2012). A rule-based approach for estimating software development cost using function point and goal and scenario based requirements. *Expert Systems with Applications*, 39(1), 406-418.

Chung, L., Nixon, B.A., Yu, E., Mylopoulos, J., (1995). *Non-Functional Requirements in Software Engineering*. Springer Science & Business Media.

Engelsman, W., & Wieringa, R. (2014). Understandability of goal-oriented requirements engineering concepts for enterprise architects. In *Advanced Information Systems Engineering* (pp. 105-119). Springer International Publishing.

Eriksson, H. E., & Penker, M. (2000). *Business modeling with UML*. Chichester: Wiley.

Estrada, H., Martínez, A., Pastor, O., & Sánchez, J. (2002). Generación de Especificaciones de Requisitos de Software a partir de Modelos de Negocios: un enfoque basado en metas. In *V Workshop de Engenharia de Requisitos WER*. Vol (2).

Estrada, H., Martínez, A., Santillán, L. C., & Pérez, J. (2014). A New Service-Based Approach for Enterprise Modeling. *Computación y Sistemas*, 17(4), 625-639.

Fenton, N., & Bieman, J. (2014). *Software metrics: a rigorous and practical approach*. CRC Press.

González-Baixauli, B., Laguna, M. A., & do Prado Leite, J. C. S. (2004). Análisis de Variabilidad con Modelos de Objetivos. In *VII Workshop de Engenharia de Requisitos WER*.

Grillo O., & La Rosa M. (2009). Sistema Administrador de Requerimientos y Planificador de Tareas. (Tesis pregrado, Pontificia Universidad Católica del Perú).

Herrmann, A., & Paech, B. (2008). MOQARE: misuse-oriented quality requirements engineering. *Requirements Engineering*, 13(1), 73-86.

Horkoff, J., Barone, D., Jiang, L., Yu, E., Amyot, D., Borgida, A., & Mylopoulos, J. (2014). Strategic business modeling: representation and reasoning. *Software & Systems Modeling*, 13(3), 1015-1041.

IEEE (1990): Standard 610, Computer Dictionary. Nueva York.

Kepner, C. H., & Tregoe, B. B. (1976). *The rational manager; a systematic approach to problem solving and decision making*. Kepner-Tregoe.

Ishikawa, K., (1986). *Guide to quality control*. Asian Productivity Organization.

Martínez, R, A., Estrada, E, H., & Gama, M, L. A. (2009). Una guía rápida de la metodología tropos. *REVISTA GTI*, 7(19), 67-77.

Lindvall, M., & Sandahl, K. (1996). Practical implications of traceability. *Softw, Pract. Exper.* 26(10), 1161-1180.

Liu, L., & Jin, Z. (2007). Requirements Analyses Integrating Goals and Problem Analysis Techniques. *Tsinghua Science & Technology*, 12(6), 729-740.

Martínez, A., Pastor, O., & Estrada, H. (2005). A pattern language to join early and late requirements. *Journal of Computer Science & Technology*, 5(2), 64-70.

Martínez, A., Pastor, O., Mylopoulos, J., & Giorgini, P. (2006). From Early Requirements to Late Requirements: A goal-based approach. In *Proceedings of Eight International Bi-Conference Workshop on Agent-Oriented Information System* (pp. 5-12).

Martínez, M. R., Muñoz, V., Monserrat, M. A., Muñoz, V., & Serafín, J. G. (2014). Cultura Organizacional Y Efectividad En Las Pequeñas Empresas Constructoras De Puebla, México (Organizational Culture and Effectiveness in Small Construction Businesses in Puebla, México). *Revista Internacional Administración & Finanzas*, 7(4), 79-92.

Morandini, M., Dalpiaz, F., Nguyen, C. D., & Siena, A. (2014). The Tropos Software Engineering Methodology. In *Handbook on Agent-Oriented Design Processes* (pp. 463-490). Springer Berlin Heidelberg.

Navarro, J. J., Valero-García, M., Sanchez, F., & Tubella, J. (2000). Formulación de los objetivos de una asignatura en tres niveles jerárquicos. *Memorias de las VI Jornadas sobre la Enseñanza Universitaria de la Informática JENUI* (pp. 457-462).

Nguyen, T. H., Vo, B. Q., Lumpe, M., & Grundy, J. (2014). KBRE: a framework for knowledge-based requirements engineering. *Software Quality Journal*, 22(1), 87-119.

Perales, F. J. (1993). La resolución de problemas: Una revisión estructurada. *Enseñanza de las Ciencias*, 11 (2), 170-178.

Rolón, E., Ruiz, F., Rubio, F. G., & Piattini, M. (2005). Aplicación de métricas software en la evaluación de modelos de procesos de negocio. *Revista Electrónica de la Sociedad Chilena de Ciencia de la Computación*, 6(1), 1-10.

Rupérez, F. L. (1989). Dependencia-independencia de campo y educación científica. *Revista de educación*, (289), 235-258.

Rittel, H. W., & Webber, M. M. (1973). Dilemmas in a general theory of planning. *Policy sciences*, 4(2), 155-169.

Saravia, J. A. (2004). *Guía para la elaboración del marco lógico*. Universidad Nacional de Colombia.

Tabares, M. S., Moreira, A., Anaya, R., Arango, F., & Araujo, J. (2007). A traceability method for crosscutting concerns with transformation rules. In *Proceedings of the Early*

Aspects at ICSE: Workshops in Aspect-Oriented Requirements Engineering and Architecture Design (p. 7).

Van Lamsweerde, A., (2004). Goal-oriented requirements engineering: a roundtrip from research to practice [engineering read engineering]. In Proceedings Requirements Engineering Conference (pp. 4-7).

Van Dijk, T. A. (2005). Estructuras y funciones del discurso: una introducción interdisciplinaria a la lingüística del texto ya los estudios del discurso. Siglo XXI.

Vargas, F. (2010). Método para establecer la consistencia de los problemas en el diagrama causa-efecto con el diagrama de objetivos de KAOS. (Tesis de maestría Universidad Nacional de Colombia).

Vargas, F., Zapata, C. M., Martinez, A & Estrada, H. (2015). Método para especificar y formalizar requisitos a partir de problemas del dominio en el contexto de educación de requisitos de software. Revista Iberoamericana de Automática e Informática industrial. Por aparecer.

Yu, E. (1995). Modelling Strategic Relationships for Process Reengineering. (Ph.D. Thesis University of Toronto).

Waseem, M. (2014). Software Requirement Engineering (Doctoral dissertation, International Islamic University Islamabad).

Wieringa, R., Maiden, N., Mead, N., & Rolland, C. (2006). Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. Requirements Engineering, 11(1), 102-107.

Zapata, C. M. & Arango, F. (2004). Alineación entre metas organizacionales y elicitación de requisitos del software. Dyna, 71(143), 101-110.

Zapata, C., & Arango, F. (2009). The UNC-Method: a problem based software development method. Ingeniería e Investigación, 29(1), 69-75.

Zapata, C. M., Acevedo, J. F., & Moreno, D. A. (2011). Representación de relaciones semánticas entre problemas y objetivos mediante lógica de predicados. *Revista EIA*, (15), 61-72.

Zapata, C., (2007). Definición de un esquema preconceptual para la obtención automática de esquemas conceptuales de UML (Doctoral dissertation, Ph. D. Thesis, Universidad Nacional de Colombia).

Zowghi, D., & Gervasi, V. (2002). The Three Cs of requirements: consistency, completeness, and correctness. In *International Workshop on Requirements Engineering: Foundations for Software Quality*, Essen, Germany: Essener Informatik Beitiage (pp. 155-164)

Zapata, C., & Vargas, F. (2009). Una revisión de la literatura en consistencia entre problemas y objetivos en Ingeniería de Software y Gerencia Organizacional. *Revista Escuela de Ingeniería de Antioquia*, (11), 117-129.

Zapata, C. M., & Lezcano, L. A. (2009). Caracterización de los verbos usados en el diagrama de objetivos. *Dyna*, 76(158), 219-228.