

**MÉTODO DE CONVERSIÓN DE UN DIÁLOGO CONTROLADO A UN
DISCURSO EN UN-LENCEP**

WILIAM ALFONSO ARÉVALO CAMACHO

Ingeniero de Sistemas



Universidad Nacional de Colombia
Facultad de Minas – Escuela de Sistemas
Maestría en Ingeniería – Ingeniería de Sistemas
Área de Ingeniería de Software
Medellín, Colombia.
2010



Método de conversión de un diálogo controlado a un discurso en UN-Lencep
Tesis de Maestría en Ingeniería – Ingeniería de Sistemas
Universidad Nacional de Colombia.
William A. Arévalo Camacho

MÉTODO DE CONVERSIÓN DE UN DIÁLOGO CONTROLADO A UN DISCURSO EN UN-LENCEP

WILIAM ALFONSO ARÉVALO CAMACHO

Trabajo de investigación presentado como requisito parcial para optar al
Título Magister en Ingeniería – Ingeniería de Sistemas.

Este documento tiene únicamente propósitos de evaluación y no debería ser consultado o
referido por cualquier persona diferente a los evaluadores.

Director: CARLOS MARIO ZAPATA JARAMILLO, PhD.

Universidad Nacional de Colombia
Facultad de Minas – Escuela de Sistemas
Maestría en Ingeniería – Ingeniería de Sistemas
Área de Ingeniería de Software
Medellín, Colombia.
2010



Método de conversión de un diálogo controlado a un discurso en UN-Lencep
Tesis de Maestría en Ingeniería – Ingeniería de Sistemas
Universidad Nacional de Colombia.
William A. Arévalo Camacho

DEDICATORIA

A mi esposa, por su amor incondicional y apoyo en el deseo de cumplir mis metas.

A mi familia por el apoyo y el respaldo que siempre me brindan.

William.



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

AGRADECIMIENTOS

Estas son algunas de las personas y entidades que hicieron realidad este logro, para todos ellos muchas gracias. Mi mayor agradecimiento es para el profesor Carlos Mario Zapata Jaramillo, quien puso a mi disposición todo su apoyo y conocimientos para llevar a cabo este trabajo, enseñándome la tarea de la Investigación. También, agradezco al profesor Jhoan Suarez, por la colaboración, tiempo e interés puesto con sus explicaciones y puntos de vista, que desde su área fueron de gran apoyo en la estructuración de las reglas. Además, agradezco las enseñanzas, apoyo y colaboración de los profesores Gloria Lucía Giraldo Gómez, Jovani Alberto Jiménez Builes, Fernando Arango Isaza, John William Branch, quienes me brindaron su experiencia y conocimientos para llevar a feliz término esta meta.

Adicionalmente, quiero extender este agradecimiento a los alumnos Natalia Meneses, Bryan Zapata, Camilo Trujillo y David Moreno, por sus portes hechos en el proceso de investigación. Además, agradezco el apoyo recibido de los proyectos “Un Modelo de diálogo para la generación automática de especificaciones en UN-Lencep”, financiado por la DIME y “Entorno colaborativo de apoyo a la mejora de procesos para la industria de software colombiana”, financiado por Colciencias. Finalmente, destacar el apoyo recibido de la Corporación Centro de Investigación y Desarrollo de Nuevas Tecnologías “CIDENET” y la Corporación Parquesoft Popayán.



TABLA DE CONTENIDO

DEDICATORIA	I
AGRADECIMIENTOS	II
TABLA DE CONTENIDO	III
INDICE DE FIGURAS	V
INDICE DE TABLAS	VII
RESUMEN	VIII
ABSTRACT	IX
I. INTRODUCCION	1
II. MARCO TEORICO	4
2.1 EDUCCIÓN DE REQUISITOS	4
2.1.1 <i>Técnicas de Educción de Requisitos</i>	5
2.2 LA COMUNICACIÓN Y LOS MODELOS DE DIÁLOGO	10
2.3 LENGUAJES CONTROLADOS	12
2.3.1 <i>UN-Lencep</i>	13
III. ANTECEDENTES	20
3.1 DIÁLOGOS DE OBTENCIÓN DE INFORMACIÓN	20
3.1.1 <i>Modelos de diálogo de educción de requisitos de software</i>	20
3.1.2 <i>Modelos de diálogo de educción de requisitos de productos</i>	29
3.2 DIÁLOGOS DE PRESENTACIÓN DE INFORMACIÓN	32
3.2.1 <i>Diálogos computacionales para presentar información gráfica o textual</i>	32
3.2.2 <i>Diálogos computacionales para presentar información hablada</i>	34
3.3 SIMULACIÓN DE LA RELACIÓN HUMANO-HUMANO	36
3.3.1 <i>Asistenciales y de gestión de información</i>	37
3.3.2 <i>Apoyo psicológico.</i>	39
3.3.3 <i>Realización de tareas o funciones</i>	42
3.4 ANÁLISIS CRÍTICO DE LA LITERATURA CONSULTADA	43
IV. DEFINICION DEL PROBLEMA DE INVESTIGACION	46
4.1 LIMITACIONES ENCONTRADAS	46
4.2 OBJETIVOS	49
4.2.1 <i>Objetivo General</i>	49
4.2.2 <i>Objetivos Específicos</i>	49
V. PROPUESTA DE SOLUCION	51



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

5.1	MÉTODO PROPUESTO	51
5.1.1	<i>Modelo de Entrevista</i>	51
5.1.2	<i>Reglas de Transformación del diálogo</i>	53
5.1.3	<i>Integración del Modelo</i>	59
5.1.4	<i>Prototipo Funcional</i>	61
VI.	VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN	71
6.1	CASO DE ESTUDIO: FACTURACIÓN EN PROPIEDAD HORIZONTAL	71
6.1.1	<i>Discurso en UN–Lencep del Proyecto Facturación PH</i>	73
6.1.2	<i>Esquema Preconceptual del Proyecto Facturación PH</i>	76
6.1.3	<i>Esquemas Conceptuales de UML del Proyecto Facturación PH</i>	77
6.2	CASO DE ESTUDIO: MODELO DE DIÁLOGO PARA LA EDUCCIÓN DE REQUISITOS BASADO EN UN-LENCEP	81
6.2.1	<i>Discurso en UN–Lencep del Proyecto Modelo de Diálogo</i>	83
6.2.2	<i>Esquema Preconceptual del Proyecto Modelo de Diálogo</i>	87
6.2.3	<i>Esquemas Conceptuales de UML del Proyecto Modelo de Diálogo</i>	88
6.3	CASO DE ESTUDIO: HERRAMIENTA DE APOYO A LA GESTIÓN DE PROYECTOS BASADA EN EL MODELO DE PROCESO COMPETISOFT	92
6.3.1	<i>Discurso en UN–Lencep del Proyecto Herramienta de Apoyo Basada en el Modelo Competisoft</i>	94
6.3.2	<i>Esquema Preconceptual del Proyecto Herramienta de Apoyo Basada en el Modelo Competisoft</i>	97
6.3.3	<i>Esquemas Conceptuales de UML del Proyecto Herramienta de Apoyo Basada en el Modelo Competisoft</i>	98
VII.	CONCLUSIONES	103
VIII.	TRABAJO FUTURO	105
	REFERENCIAS	106



INDICE DE FIGURAS

Figura 1 Representación gráfica de la entrevista.....	7
Figura 2. Definición de Comunicación (Ongallo, 2007).....	11
Figura 3. Elementos Básicos del EP (Zapata <i>et al.</i> , 2006).....	14
Figura 4. Plantilla del diagrama que genera FAES (Gilvaz & Leite, 1995).....	21
Figura 5. Diagrama de flujo de la propuesta de Lecouche <i>et al.</i> (1998).....	23
Figura 6. Arquitectura de EXPECT (Gil <i>et al.</i> , 2000).....	24
Figura 7. Configuración de la Herramienta (Kassel & Malloy, 2003).....	25
Figura 8. Propuesta de Kozima <i>et al.</i> (2005).....	26
Figura 9. Actividades en FGDRE (Mohd & Salwah, 2008).....	28
Figura 10. Modelo de educación de requisitos (Wang & Zeng, 2009).....	31
Figura 11. Arquitectura de CORVIS (Hwang <i>et al.</i> , 2006).....	33
Figura 12. Estructura del la propuesta de (Umeda <i>et al.</i> , 2003).....	35
Figura 13. Arquitectura del Modelo de Diálogo (Varge <i>et al.</i> , 2008).....	36
Figura 14. Arquitectura del sistema DIHANA (Hurtado <i>et al.</i> , 2005).....	39
Figura 15. Esquema del modelo propuesto (Prendinger <i>et al.</i> , 2005).....	40
Figura 16. Estructura del modelo afectivo (Liu <i>et al.</i> , 2008).....	41
Figura 17. Diagrama Causa–Efecto del problema estudiado.....	48
Figura 18. Diagrama de Objetivos de la propuesta.....	50
Figura 19. Estructura del Modelo de Diálogo Propuesto.....	60
Figura 20. Menú Principal del Prototipo.....	61
Figura 21. Interfaz de las preguntas P1 y P2 de la entrevista.....	62
Figura 22. Interfaz de la pregunta P3 de la entrevista.....	63
Figura 23. Interfaz de la pregunta P4 de la entrevista.....	63
Figura 24. Interfaz de la pregunta P5 de la entrevista.....	64
Figura 25. Interfaz de la pregunta P6 de la entrevista.....	64
Figura 26. Interfaz de la pregunta P7 de la entrevista.....	65
Figura 27. Interfaz de la pregunta P8 de la entrevista.....	65
Figura 28. Interfaz de la pregunta P9 de la entrevista.....	65
Figura 29. Interfaz de la pregunta P10 de la entrevista.....	66
Figura 30. Interfaz de la pregunta P11 de la entrevista.....	66
Figura 31. Interfaz de la pregunta P12 de la entrevista.....	66
Figura 32. Interfaz de la pregunta P15 de la entrevista.....	67



Figura 33. Interfaz de la pregunta P16 de la entrevista	68
Figura 34. Interfaz de la pregunta P17 de la entrevista	68
Figura 35. Interfaz para visualizar el discurso del interesado	69
Figura 36. Interfaz para definir regla.....	70
Figura 37. Esquema Preconceptual generado con el discurso en UN-Lencep del proyecto Facturación PH	76
Figura 38. Diagrama de Clases del Proyecto Facturación PH.....	77
Figura 39. Diagrama de Comunicación del Proyecto Facturación PH.....	78
Figura 40. Diagrama de Máquina de Estados de los objeto Factura	79
Figura 41. Diagrama de Máquina de Estados de los objeto Recibo de Pago	79
Figura 42. Diagrama de Máquina de Estados de los objeto Soporte de Pago	79
Figura 43. Diagrama de Casos de Uso del Proyecto Facturación PH	80
Figura 44. Esquema preconceptual del Modelo de diálogo propuesto.....	87
Figura 45. Diagrama de Clases del proyecto Modelo de Diálogo.....	88
Figura 46. Diagrama de Comunicación del proyecto Modelo de Diálogo	89
Figura 47. Diagrama de Máquina de Estados de Pregunta y Regla	90
Figura 48. Diagrama de secuencias del Proyecto Modelo de Diálogo	90
Figura 49. Diagrama de Casos de Uso Realizar Entrevista.....	91
Figura 50. Diagrama de Casos de Visualizar UN-Lencep.....	91
Figura 51. Diagrama de Casos de Uso Aplicar Regla	92
Figura 52. Esquema Preconceptual del proyecto herramienta de apoyo basada en el Modelo Competisoft	98
Figura 53. Diagrama de Clases del proyecto herramienta de apoyo basada en el Modelo Competisoft	99
Figura 54. Diagrama de comunicación del proyecto herramienta de apoyo basada en el Modelo Competisoft.....	100
Figura 55. Diagrama de Máquina de Estados para el proyecto herramienta de apoyo basada en el Modelo Competisoft	100
Figura 56. Diagrama de secuencias del Proyecto herramienta de apoyo basada en el Modelo Competisoft	101
Figura 57. Diagrama de Caso de Uso Solicitar Software	102
Figura 58. Diagrama de Casos de Uso Establecer Patrón de Proceso	102



INDICE DE TABLAS

Tabla 1. Descripción de los elementos del UN-Lencep. Elaboración propia del autor con base en Zapata <i>et al.</i> (2006).....	14
Tabla 2. Verbos de logro (Zapata & Lezcano, 2009)	16
Tabla 3. Análisis de los modelos de diálogo en Educación de Requisitos.....	45
Tabla 4. Preguntas del Modelo y su secuencia.....	52
Tabla 5. Regla para la obtención de Roles como Conceptos.....	54
Tabla 6. Regla para la obtención de la Tríada Estructural ‘es’	54
Tabla 7. Regla para la obtención de la Tríada Estructural ‘tiene’	55
Tabla 8. Regla para la obtención del Posible Valor	56
Tabla 9. Regla para la obtención de la Tríada Dinámica.....	57
Tabla 10. Regla para la obtención de la Implicación	57
Tabla 11. Regla para la obtención del Condicional.....	58
Tabla 12. Regla para la obtención del Conjunto de Logro	59
Tabla 13. Respuestas de la entrevista con el Administrador de San Jacinto	71
Tabla 14. Respuestas de la entrevista para prototipo del Modelo de Diálogo.....	81
Tabla 15. Respuestas de la entrevista para el proyecto herramienta de apoyo basada en el Modelo Competisoft.....	93



RESUMEN

El análisis de requisitos es una etapa del ciclo de vida del software que comprende las tareas de educación, modelado, validación y especificación de los requisitos. Adicionalmente, la especificación de requisitos debe ser correcta, no ambigua, completa, consistente, jerarquizada, verificable, modificable y trazable. Para realizar este proceso se requiere un contacto permanente con el interesado, de forma que toda la información se constate con él. Existen diversas técnicas para obtener la información necesaria para una especificación de requisitos, pero la que más se utiliza es la entrevista, la cual es un diálogo entre el analista y el interesado. Sin embargo, en un diálogo es común encontrar factores que afectan el entendimiento entre los actores. Esto hace que la tarea de obtener la información necesaria del interesado sea una de las más complejas en el proceso de especificación de requisitos.

En procura de disminuir la ambigüedad existente en el diálogo, se desarrollaron los lenguajes controlados, que son subconjuntos del lenguaje natural. Los lenguajes controlados poseen una estructura similar al lenguaje natural, con reglas léxicas, reglas gramaticales, signos y palabras. Los lenguajes controlados tienen diversos usos. En la especificación de requisitos de software se encuentra el lenguaje controlado UN-Lencep, que permite presentar del discurso del interesado de una forma que se pueda validar. Adicionalmente, presenta la información de manera concreta, inambigua y completa. Sin embargo, la información se debe obtener mediante el diálogo con el interesado, dejando en manos del analista la identificación de los elementos necesarios para la estructuración del discurso, lo que posibilita la aparición de errores. Para reducir esta problemática, se propone, en esta Tesis, la estructuración de una secuencia ordenada de preguntas y la definición de las reglas necesarias para convertir las respuestas en el discurso del interesado, expresado en UN-Lencep. Adicionalmente, en un prototipo funcional se incluyen estos elementos y se valida con la especificación de algunos proyectos que requieren el desarrollo de una aplicación de software.



ABSTRACT

The requirements analysis is a phase of software development life-cycle. It comprises: requirements elicitation, modeling, validation and specification. Also, the requirements specification should be: correct, unambiguous, complete, consistent, hierarchical, verifiable, modifiable, and traceable. This process requires a permanent contact with the stakeholder, in such way that he can verify the information. There are several techniques for obtain information for the requirements specification, but the most used is the interview, which is a dialog between the analyst and the stakeholder. In such dialog the stakeholder has the information about the software domain and the analyst knows how make the software specification. However, some problems arise in dialogs that affect the understanding between their actors. They make most complex the task of requirements specification process, in order to obtain the needed information from the stakeholder.

The controlled languages (subsets of natural language) help to decrease the ambiguity in the dialog. The controlled languages have a similar structure to the natural language, with lexical and grammatical rules, signs, and words. The controlled languages have several uses. For the software requirements specification, a controlled language called UN-Lencep is used, for allowing elaboration and validation of the stakeholder discourse. UN-Lencep presents the information in a concrete, unambiguous, and complete way. However, the information must be obtained by using a dialog with the stakeholder, but the analyst must identify the elements of the discourse structure, and this is a possible cause of mistakes in the process. As a way to reduce this problem, we propose, in this Thesis, the organization of an ordered sequence of questions that allow guidance to analyst on the identification of the needed elements for creating the discourse. Also, we establish a set of rules for translating the answers into a UN-Lencep stakeholder discourse. We include these features into a functional prototype, which we use in a lot of projects that needs to develop a software application, so we validate the functionality of all elements.



I. INTRODUCCION

Según lo expresa Leite, (1986) en la Ingeniería de software la educación de requisitos es una de las tareas iniciales en el desarrollo de una aplicación. Esta tarea consiste en identificar las características del dominio de aplicación que debe tener el producto software por desarrollar para satisfacer las necesidades del interesado (término tomando del vocablo inglés *stakeholder*). Esta es la tarea más compleja e importante del desarrollo, pues es aquí donde se establecen todos los elementos que harán parte del producto a desarrollar y es la base de todos los procesos subsiguientes (Sommerville, 2007).

Según Leffingwell y Widrig (1999), en la educación de requisitos existen diversas técnicas, tales como entrevistas y cuestionarios, tormentas de ideas, puntos de vista y escenarios, entre otras. No obstante, la técnica que más se utiliza en la educación de requisitos es la entrevista, dado que permite obtener información puntual y concreta por medio del diálogo con el interesado. Además, esta técnica ayuda a encontrar información oculta, difícil de obtener con otras técnicas. Pese a esto, no existe un esquema definido de las preguntas a realizar ni el orden en que el analista las hará, pues es difícil determinar el área en la que se desarrollará una determinada aplicación de software.

La mayoría de técnicas utilizadas en la educación de requisitos se basan en un diálogo entre el analista y el interesado, pues es éste último quien posee la información del dominio de aplicación del software. Sin embargo, el problema más frecuente es la incomprensión de las necesidades del interesado (Leffingwell & Widrig, 1999). Adicionalmente, el interesado no tiene una idea concreta de lo que necesita y no posee una cultura que permita simplificar la información para evitar la información inútil. Esto hace que el interesado sea poco específico y tienda a confundir o mezclar las características que debe tener el proceso a sistematizar (Sommerville, 2007).



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

Usualmente, los analistas presentan la información obtenida del interesado en artefactos denominados esquemas conceptuales. Los más utilizados son los esquemas de UML (*Unified Modeling Language*) lenguaje gráfico para visualizar, especificar, construir y documentar sistemas (Fowler & Scott, 1999). Estos artefactos requieren, para su elaboración y lectura, ciertos conocimientos técnicos que, normalmente, no posee el interesado, lo cual le dificulta la validación de la información (Zapata *et al.*, 2006).

Para ayudar en las actividades de análisis, modelado, depuración y pruebas, se desarrollaron algunas herramientas conocidas como CASE (*Computer-Aided Software Engineering*), que pueden incluir generadores de código fuente a partir del modelo del sistema (Sommerville, 2007). Sin embargo, pese a permitir la automatización de gran parte del proceso de desarrollo de software, estas herramientas se diseñan para que el ingeniero de software las utilice, por lo tanto, es muy poco el aporte que se puede obtener del interesado en su uso.

Considerando esta problemática, Zapata *et al.* (2006) desarrollaron UN-Lencep (Lenguaje Controlado para la Especificación de Esquemas Preconceptuales). Este lenguaje permite plasmar el discurso del interesado de forma cercana al lenguaje natural, facilitando el entendimiento y la validación de la información allí contenida. UN-Lencep se concibe como un paso intermedio entre el discurso del interesado y los esquemas conceptuales del UML. Adicionalmente, posee tratamiento computacional para la obtención de algunos esquemas conceptuales de UML.

Aunque UN-Lencep facilita la comprensión de la información del interesado, es el analista quien debe extraer, del diálogo con él, los elementos necesarios para construir el discurso (Zapata *et al.*, 2006). Por ello, el analista debe buscar estrategias que le permitan obtener dicha información, desde el conocimiento del interesado. En este orden de ideas, Zapata y Giraldo (2009) proponen un juego que simula una entrevista, como una técnica para ayudar al analista en la obtención de la información necesaria para construir el discurso.



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

La entrevista se usa porque, de las diversas formas de comunicación, el diálogo es el medio más utilizado de comunicación entre las personas. No obstante, el diálogo es una de las actividades más complejas de la interacción entre los seres humanos, que se afecta con diferentes elementos que producen errores e incompreensión entre sus actores (Ongallo, 2007). Adicionalmente, en un diálogo analista-interesado es común encontrar que estos defectos en la comunicación se incrementan, debido a las diferencias del conocimiento de cada actor (Zapata *et al.*, 2006). Por ello, establecer limitantes con un diálogo controlado hace que sus actores sean concretos y específicos. Esto ayuda a disminuir los errores de comunicación presentes en el lenguaje natural (Ongallo, 2007). Además, con una entrevista donde se definan las preguntas y su orden, se logra que el interesado responda de forma concreta, se evita la desatención y se obtienen mejores resultados (Clark-Carter, 2002).

La presente Tesis de Maestría delimita las características del diálogo para el proceso de educación de requisitos. Además, establece un modelo de entrevista con un grupo de preguntas y el orden para realizarlas. Asimismo, define un conjunto de reglas que permiten identificar los elementos de UN-Lencep desde el diálogo controlado. Para la validación de la propuesta, se incorpora el método definido en una herramienta computacional que automatiza todo el proceso.

Esta Tesis se estructura de la siguiente forma: en el Capítulo II se define el Marco Conceptual del tema de investigación que motiva esta Tesis; el Capítulo III se realiza un análisis descriptivo de los aportes más relevantes, complementado con un análisis crítico de la literatura consultada, incluyendo una comparación de los diferentes métodos; en el Capítulo IV se realiza la Definición del Problema de Investigación que motiva esta Tesis; en el Capítulo V se presenta y debate la propuesta de solución encontrada al problema planteado; en el Capítulo VI se presentan los casos de estudio y las pruebas realizadas para validar la propuesta. Finalmente, en el Capítulo VII se resumen los hallazgos y se presenta el trabajo futuro que se puede derivar de esta propuesta.



II. MARCO TEORICO

La presente Tesis se enmarca en la ingeniería de requisitos, la cual es la disciplina que abarca todas las actividades de educación, documentación y mantenimiento de los requisitos de un sistema basado en computadores. Esta disciplina es una rama de la ingeniería de software, la cual, comprende todos los aspectos de la producción de software desde las etapas iniciales del sistema, hasta el mantenimiento de éste después de su uso. Del mismo modo, la ingeniería de software está inmersa en la ingeniería de sistemas, que, comprende el desarrollo de hardware y software, políticas y procesos de diseño y distribución de los sistemas (Sommerville, 2007).

2.1 Educación de Requisitos

El análisis de requisitos lo componen la educación, modelado, validación y especificación de requisitos. En la educación de requisitos se definen las características que tendrá el producto software. En el modelado se organizan los requisitos obtenidos de forma coherente. En la validación se organizan los requisitos educidos de acuerdo con la prioridad y se solucionan los conflictos encontrados. La última tarea es plasmar los requisitos en el documento de especificación (Sommerville, 2007).

La IEEE (*Institute of Electrical and Electronics Engineers*) define el estándar 830, el cual establece las características que debe tener una especificación de software. De acuerdo con esto, la especificación debe ser: correcta (cada requisito declarado se debe encontrar en el software), no ambigua (un requisito solo tiene una única significación), completa (todos los elementos relevantes se deben especificar), consistente (los requisitos no pueden ser contradictorios), jerarquizada (se pueden clasificar los requisitos por su importancia), verificable (se puede demostrar que el sistema cumple con los requisitos), modificable (los



cambios en los requisitos se pueden estructurar fácilmente) y trazable (se puede identificar el origen y los elementos del sistema que lo satisfacen) (IEEE, 1998).

La educación de requisitos se considera una de las tareas más difíciles del desarrollo de software, dado que los interesados suelen expresar los requisitos de manera muy general o en sus propios términos y con el conocimiento implícito de su trabajo. También, es común que el interesado no tenga claridad de lo que desea y cómo lo desea y esto ocasiona especificaciones inadecuadas (Leffingwell & Widrig, 1999). Comúnmente, esta tarea se realiza mediante un diálogo entre el analista y el interesado, a fin de determinar las características que tendrá el producto software a realizar (Leite, 1987).

2.1.1 Técnicas de Educación de Requisitos

Para lograr la comunicación entre el analista y el interesado y mejorar el entendimiento entre ellos, se vienen desarrollando diversas técnicas que ayudan en la educación de requisitos. Estas técnicas surgen, precisamente, de la interacción entre los actores del proceso de educación de requisitos. De las técnicas desarrolladas, unas pocas poseen un alto índice de usabilidad, pero de todas ellas la más utilizada es la entrevista (Leffingwell & Widrig, 1999). A continuación, se presenta, de forma detallada algunas de estas técnicas.

2.1.1.1 Entrevista.

La entrevista es una técnica ampliamente utilizada, debido a la facilidad de aplicación y a la poca preparación que se requiere para su práctica, pues se basa en la estructura pregunta-respuesta. Además, permite obtener gran cantidad de información, de una forma directa y sencilla. Sin embargo, es indispensable que el



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

analista comprenda el dominio del conocimiento para el cual se desarrollará la aplicación de software. Esto le dará mayor capacidad de comprender las necesidades del interesado. (Leffingwell & Widrig, 1999).

Así, las preguntas de una entrevista toman gran relevancia, pues permiten ubicar al analista con las necesidades del interesado. Por otro lado, son la base para la obtención de la información que permitirá diseñar y desarrollar una posible solución. Por ello, teniendo en cuenta la pericia del analista, la entrevista facilita identificar elementos ocultos, que con otras técnicas no se obtendrían (Leffingwell & Widrig, 1999).

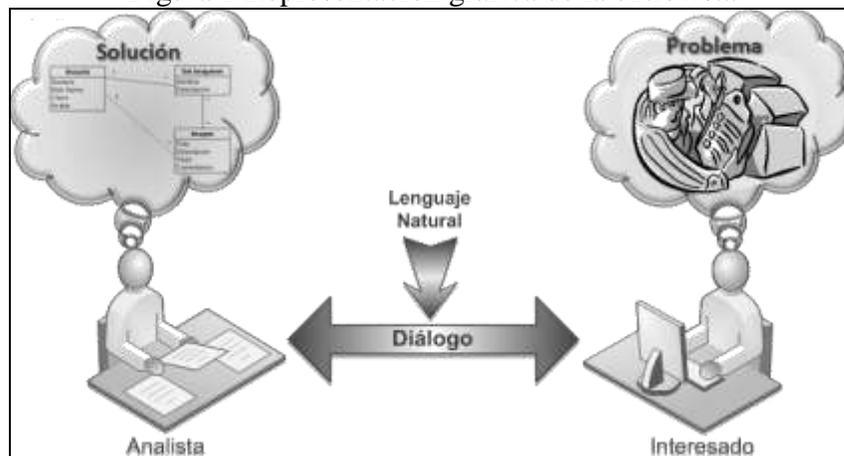
La entrevista con los interesados es una mezcla, pues las respuestas de alguna pregunta pueden conducir a otros cuestionamientos, los cuales se discuten de forma menos estructurada. La entrevista abierta pocas veces es productiva, por lo que es necesario definir algunas preguntas para empezar y mantener la entrevista centrada en el sistema a desarrollar. Adicionalmente, ésta técnica funciona mejor cuando se realiza individualmente a cada uno de los interesados (Sommerville, 2007). Sin embargo, el mayor inconveniente que presenta esta técnica es que, mientras el interesado posee el conocimiento del dominio del problema, el analista, con esta información, piensa en cómo crear los esquemas conceptuales de la especificación y el diseño (Zapata *et al.*, 2006). La figura 1 muestra el desarrollo de la entrevista de educación de requisitos.

Desde el punto de vista psicológico, Clark-Carter (2002) define tres modelos de entrevista. El primero es la entrevista abierta, la cual es una charla con el entrevistado que, generalmente, se usa para la exploración de un tema que el entrevistador desconoce. Sin embargo, con este modelo de entrevista se puede obtener información innecesaria o perder el enfoque del tema principal. El segundo es la entrevista semiestructurada, la cual usa un conjunto de preguntas definida para



obtener información del dominio y el entrevistador posee conocimientos del tema principal. No obstante, el entrevistador puede incluir preguntas de acuerdo con el flujo de la entrevista. Por último, la entrevista estructurada se usa cuando el entrevistador tiene claridad de las posibles respuestas del entrevistado. Así, cada pregunta presenta un conjunto de posibles respuestas. De esta forma, se disminuyen los defectos del diálogo y se obtienen respuestas concretas y específicas.

Figura 1 Representación gráfica de la entrevista



2.1.1.2 Grupos de Discusión.

Un grupo de discusión es una técnica que reúne a los interesados y genera un espacio para la deliberación sobre las necesidades del proyecto. Esta técnica permite que cada uno de los presentes exprese sus necesidades y expectativas respecto del producto a desarrollar. De esta forma, el(los) analista(s), obtiene(n) los elementos relevantes para el diseño de la solución en consenso con los interesados (Leffingwell & Widrig, 1999).



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

Como esta actividad se realiza en grupo, el líder del grupo de desarrollo debe realizar la labor de moderador o, en su defecto, establecer quien realizará esta labor. El moderador será quien se encargue del avance de la reunión. Por la importancia de este rol, se requiere experiencia o un nivel mínimo de capacitación. También, debe ser una persona que infunda respeto tanto en el grupo de desarrollo como en el grupo de los interesados (Leffingwell & Widrig, 1999).

Otro aspecto importante, es una buena selección del grupo de interesados, quienes deben tener amplio dominio del problema, de forma que sus aportes sean efectivos para el proceso. Esta es una técnica intensiva y requiere máximo dos días para su desarrollo, con el fin de no entorpecer las labores de los interesados dentro de la organización. De esta forma, se obtiene el punto de vista de cada uno de los participantes, lo cual le añade completitud en la información (Leffingwell & Widrig, 1999).

2.1.1.3 Tormenta de Ideas.

La tormenta de ideas, también conocida como lluvia de ideas, es una técnica de educación de requisitos simple, fácil de usar y didáctica. El objetivo principal de esta técnica es generar ideas de todas las partes involucradas y muchas veces se usa en los grupos de discusión. La tormenta de ideas tiene dos fases: la generación y la reducción de ideas. En la fase de generación se detallan las ideas expresadas y en la fase de reducción se analiza, organiza, agrupa y sintetiza la información obtenida en la fase anterior (Leffingwell & Widrig, 1999).

En la primera fase, el facilitador orienta al grupo, involucra a todos los asistentes, toma nota de las ideas expresadas sin analizarlas o juzgarlas. Posteriormente, en la segunda fase se realiza la depuración de la información obtenida, se agrupan las



ideas similares, se hace una pequeña descripción y se priorizan las ideas seleccionadas en la depuración (Leffingwell & Widrig, 1999).

Para realizar una sesión exitosa de tormenta de ideas, se deben tener en cuenta las siguientes consideraciones: el facilitador explica el objetivo del proceso, no se permiten las críticas a las ideas expresadas, como tampoco se debatirá sobre ellas y se incentiva la imaginación para generar una mayor cantidad de ideas. El facilitador puede combinar las ideas para generar un nuevo enfoque, el cual se analiza y evalúa con los participantes (Leffingwell & Widrig, 1999).

2.1.1.4 Escenarios.

Con la técnica de escenarios, el interesado describe, con ejemplos de la vida real, las interacciones del sistema. Esta técnica es de gran utilidad en situaciones donde el interesado no tiene claro lo que desea. De esta forma, con la descripción que realiza el interesado se obtienen los requisitos. Cada escenario abarca una o más interacciones que realiza el interesado con el sistema. Esta técnica se utiliza, especialmente, en las metodologías ágiles de desarrollo de software (Sommerville, 2007).

Para realizar la educación de requisitos utilizando esta técnica, se inicia con un bosquejo de la interacción y se le agregan detalles hasta describirla completamente. Un escenario contiene, entre otros elementos, una descripción general de lo que el interesado espera del sistema, una descripción del flujo normal de los eventos, una descripción de los posibles errores y cómo manejarlos, una relación de otras actividades que se podrían llevar a cabo al mismo tiempo y una descripción del sistema cuando termina la interacción (Sommerville, 2007).



2.1.1.5 Prototipos.

El desarrollo de prototipos facilita la comprensión de los requisitos del cliente. El prototipo se centra en experimentar con los requisitos que no se comprenden completamente (Sommerville, 2007). Esta técnica le permite al interesado ver una parte del nuevo sistema o de una funcionalidad, para determinar si se ajusta a sus necesidades, en tanto que, para el analista, le permite capturar aspectos relevantes en la construcción de la interfaz de usuario para incluirlas en el sistema (Leffingwell & Widrig, 1999).

Las técnicas aquí presentadas no son las únicas que se utilizan en la educación de requisitos de software; también se encuentran técnicas como los casos de uso, los juegos de roles y los puntos de vista, entre otras. Sin embargo, las técnicas descritas son algunas de las que más se utilizan, como también algunas de ellas son básicas para el desarrollo de otras técnicas.

2.2 La Comunicación y los Modelos de Diálogo

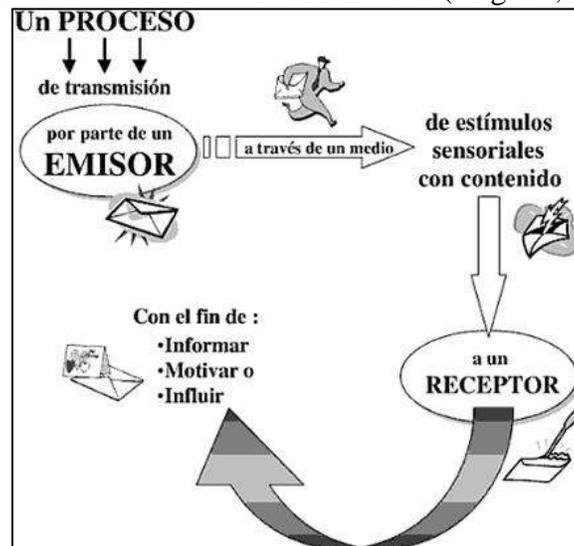
La comunicación es un intercambio de información, ideas, experiencias o estímulos, utilizando un medio, y se puede dar entre dos o más individuos, con el uso de elementos comunes del conocimiento. También, involucra los procesos físicos y psicológicos mediante los cuales los seres humanos se relacionan y comprenden para alcanzar un objetivo común (Ongallo, 2007). La figura 2 presenta la definición gráfica del proceso de comunicación.

Para Ongallo (2007) la comunicación existe desde el momento en que se establece un diálogo, definido como una relación humano-humano, donde cada interlocutor acepta las



diferencias con el otro. Para que haya una comunicación efectiva, los actores del diálogo deben mantener un nivel mínimo de entendimiento. Según lo plantean Aguilera y Mañas (2001) para la ingeniería de sistemas, los conceptos de comunicación y diálogo se llevan a la relación humano-máquina y máquina-máquina. Así, en el diálogo humano-máquina la comunicación se establece a través de la Interfaz de Usuario. Bertrán y Xampeny (1978) establecen el diálogo máquina-máquina como una relación coordinada de mensajes entre ambas máquinas, la cual se debe definir de manera adecuada.

Figura 2. Definición de Comunicación (Ongallo, 2007).



Al ser el diálogo el medio que más se utiliza en la comunicación entre humanos, con el transcurrir del tiempo se definieron las reglas culturales que afectan esta interacción. Pese a lo anterior, es común, en el diálogo, encontrar ciertos factores que dificultan el entendimiento entre sus actores. Así, la falta de atención de los interlocutores en los planteamientos del otro, las diferencias culturales, de contexto y de sistemas métricos, el desconocimiento del tema, la concentración en pequeños detalles y los problemas personales, entre muchos otros elementos, son la causa de malentendidos y errores en la comunicación (Ongallo, 2007).



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

La interacción de las personas con los artefactos tecnológicos dio lugar al desarrollo del diálogo humano-máquina. En la ingeniería de software, esta interacción se da a través de la interfaz de usuario (Aguilera & Mañas, 2001). Para lograr una comunicación efectiva con el usuario, el grupo de desarrollo debe procurar reducir los defectos del diálogo. No obstante, la interacción humano-máquina no se establece culturalmente. Para el usuario es complejo y algunas veces intimidante el diálogo con la máquina (Allen *et al.*, 2001).

El ser humano posee una habilidad innata para comunicarse y, para ello, comúnmente, utiliza el lenguaje natural. Durante un diálogo, también, se evidencian muchas otras capacidades humanas, las cuales se asocian con la noción de inteligencia. Para lograr que un computador use el lenguaje natural, necesita poseer una inteligencia similar a la de un humano. Esta habilidad se intenta alcanzar con el uso del PLN (Procesamiento de Lenguaje Natural). Sin embargo, el mayor inconveniente que presenta el PLN es la ambigüedad existente en el lenguaje natural (Allen *et al.*, 2001).

Con el propósito de disminuir la ambigüedad del lenguaje natural y facilitar el PLN Allen *et al.* (2001) proponen restringir las posibilidades del usuario de un sistema a las necesarias para obtener la información requerida. Así, la máquina realiza la pregunta, en lenguaje natural, y el usuario ingresa la respuesta, también en lenguaje natural. De esta forma, se controla la interacción y se alcanza un discurso más predecible, que lleva a un mejor reconocimiento y procesamiento del lenguaje. No obstante, el usuario se puede ver obligado a suministrar información innecesaria para el objetivo que se persigue. Esto puede hacer la interacción menos eficiente.

2.3 Lenguajes Controlados

Debido a la ambigüedad del lenguaje natural, se presentan ciertos defectos en la traducción de documentación técnica, lo que genera imprecisiones que se convierten en errores, bien



sea realizada por un humano o un sistema de información. Esta preocupación dio lugar al surgimiento de los lenguajes controlados, los cuales son subconjuntos del lenguaje natural con sintaxis, semántica y terminología restringidas que facilitan el entendimiento. Así, se provee mayor claridad y coherencia de la información, se reduce la ambigüedad y se facilita la reutilización de los términos, entre otras tantas características que mejoran la transferencia y traducción de documentos (Zapata & Rosero, 2008).

Uno de los principales usos de los lenguajes controlados es la elaboración de documentación técnica, pero gracias a su funcionalidad se pueden usar en otras áreas como la ingeniería de software, las telecomunicaciones y la web entre otras. Los lenguajes controlados mejoran el entendimiento, pues producen una estandarización general de todos los mensajes que se escriben con ellos, conservando el sentido original de quien lo realiza (Zapata & Rosero, 2008). Además, los lenguajes controlados permiten precisión y sistematización, lo cual facilita la verificación y validación formal de la información allí plasmada. Esto se logra, pese a que algunos de estos lenguajes poseen una expresión limitada que dificulta su entendimiento (Wang & Zeng, 2009).

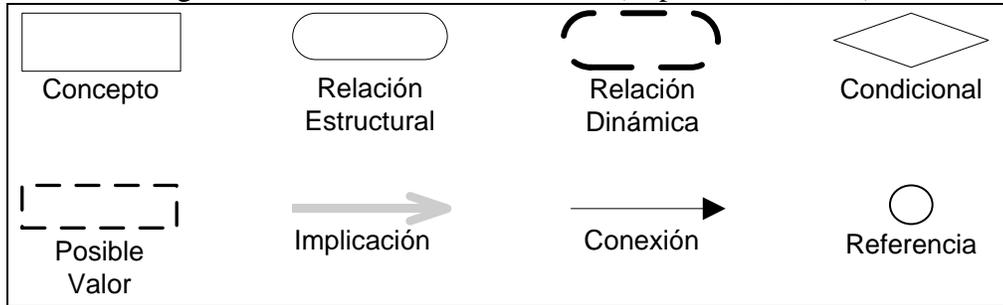
2.3.1 UN-Lencep

El UN-Lencep se concibe como un lenguaje controlado de especificación requisitos, del cual se obtienen los EPs (Esquema Preconceptual). Un EP se considera un paso intermedio entre el discurso del interesado y los esquemas conceptuales de UML, la figura 3 muestra los elementos básicos de un EP. El discurso en UN-Lencep le permite al interesado entender la información plasmada en él, en tanto que, al analista, le ayuda a comprender los elementos que hacen parte del dominio del problema, el cual el interesado expresa en lenguaje natural. Por otro lado, este lenguaje controlado posee tratamiento computacional, lo cual facilita la traducción del discurso a la especificación de requisitos en forma automática. Asimismo, el discurso posee limitaciones en la



estructura de las frases que se pueden expresar, pero sin límites en el vocabulario que se puede utilizar (Zapata *et al.*, 2006).

Figura 3. Elementos Básicos del EP (Zapata *et al.*, 2006)



Las palabras de las frases de un discurso en UN–Lencep se pueden representar en los elementos básicos que componen un EP. Así, se puede identificar el concepto, la relación estructural, la relación dinámica, la relación de logro y la conexión. La tabla 1 sintetiza los elementos básicos y las estructuras del EP. Asimismo, muestra las características de composición de cada uno de ellos y los elementos básicos que los pueden conformar. A continuación, se hace una descripción detallada de cada uno de estos elementos.

Tabla 1. Descripción de los elementos del UN-Lencep. Elaboración propia del autor con base en Zapata *et al.* (2006).

Elemento de UN-Lencep	Composición
Concepto (C)	Actor, categoría, característica o subcaracterística.
Posible Valor (PV)	Concepto y ejemplo.
Relación Estructural (RE)	Verbo estructural “es” o “tiene”.
Triada Estructural (TE)	Concepto y verbo estructural.
Relación Dinámica (RD)	Verbo de acción.
Triada Dinámica (TD)	Concepto y verbo de acción.



Continuación Tabla 1. Descripción de los elementos del UN-Lencep. Elaboración propia del autor con base en Zapata et al. (2006).

Elemento de UN-Lencep	Composición
Relación de Logro (RL)	Verbo de logro.
Conjunto de Logro (CL)	Concepto, triada dinámica o triada estructural y verbo de logro.
Condicional (CN)	Instrucción lógica y triada dinámica.
Implicación (I)	Triada dinámica o condicional y triada dinámica.

2.3.1.1 *Concepto.*

Son sustantivos o frases nominales del discurso del interesado, generalmente, representan un actor, categoría, característica o subcaracterística del dominio del problema. En un EP, un concepto sólo debe aparecer una sola vez. Toda interacción con éste se hará con una conexión. Así, se mantiene unificado el EP (Zapata & Arango, 2007).

2.3.1.2 *Relación Estructural.*

Este elemento representa una dependencia permanente entre conceptos. Así, en el discurso se utilizan únicamente los verbos “Tiene” o “Es” para la relación estructural, por ejemplo: ‘*Empleado es Persona*’ o ‘*Persona tiene Nombre*’ (Zapata & Arango, 2007).

2.3.1.3 *Relación Dinámica.*



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

Son verbos conjugados que demuestran las acciones, operaciones o funciones que se realizan en el dominio del problema. Para este elemento se suelen utilizar verbos de acción o ejecución por ejemplo ‘Registra’ o ‘Paga’ (Zapata & Arango, 2007).

2.3.1.4 Relación de Logro.

“Son verbos dinámicos y de escasa duración, que describen un evento que tiene lugar en un instante temporal único y definido, sin fases, como alcanzar la cima de una montaña”. Se asocian con los objetivos de la organización y se usan para el diagrama de objetivos. En UN–Lencep, se establece un conjunto de verbos para la relación de logro, la tabla 2 muestra la lista de estos (Zapata & Lezcano, 2009).

Tabla 2. Verbos de logro (Zapata & Lezcano, 2009)

Tipo	Verbo	Sentido	Verbo	Sentido
Mantenimiento	Administrar	Administer	Alcanzar	Attain
	Dar		Lograr	
	Gestionar		Obtener	
	Conservar	Keep	Conceder	Concede
	Guardar		Otorgar	
	Mantener		Reconocer	
	Preservar		Avalar	
	Ofrecer	Offer	Garantizar	Guarantee
Mejoramiento	Desarrollar	Develop	Acrecer	Increase
	Fomentar		Acrecentar	
	Mejorar		Agrandar	
	Urbanizar		Aumentar	
	Decrecer	Decrease	Engrandar	
	Disminuir		Incrementar	
	Valorear	Increase the value of	Menguar	
			Reducir	

Continuación Tabla 2. Verbos de logro (Zapata & Lezcano, 2009)

Tipo	Verbo	Sentido	Verbo	Sentido
-------------	--------------	----------------	--------------	----------------



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

Wiliam A. Arévalo Camacho

Realización	Causar	Make	Formar	Make
	Componer		Formular	
	Confeccionar		Hacer	
	Crear		Meter	
	Disponer		Obrar	
	Efectuar		Preparar	
	Ejecutar		Producir	
	Fabricar		Avanzar	
			Promover	

2.3.1.5 Conexiones.

Son flechas que unen los conceptos con relaciones estructurales o relaciones dinámicas, creando asociación entre ellos y denotan la conformación de la frase (Zapata & Arango, 2007).

Los elementos básicos permiten la construcción de las estructuras de un EP, las cuales representan las frases del discurso en UN-Lencep. Estas estructuras son: tríada dinámica, tríada estructural, posible valor, conjunto de logro, implicación y condicional. La descripción detallada de ellos se presenta a continuación.

2.3.1.6 Tríada Dinámica.

Estructura conformada por dos conceptos: uno origen, que representa el actor, y otro destino, que representa la característica sobre la que recae la acción. Los conceptos se unen con una relación dinámica, ejemplo ‘Veterinario Revisa Mascota’ (Zapata & Arango, 2007).



2.3.1.7 *Tríada Estructural.*

Estructura conformada por dos conceptos (uno origen y otro destino) unidos por una relación estructural, ejemplo ‘Mascota tiene Nombre’ (Zapata & Arango, 2007).

2.3.1.8 *Posible Valor.*

Estructura conformada por un concepto y una lista de valores fijos que puede tener el concepto, dentro del dominio del problema. Representa las instancias o características fijas de un elemento, ejemplo ‘Válido es un Posible Valor de Estado’ (Zapata & Arango, 2007).

2.3.1.9 *Implicación.*

Representa una relación de consecuencia o causa-efecto entre tríadas dinámicas o entre condicionales y tríadas dinámicas. La secuencia muestra el orden en que se deben realizar dichas acciones (Zapata & Arango, 2007).

2.3.1.10 *Condicional.*

Estructura conformada por una expresión que puede contener conceptos y operadores entre ellos, como condición para que se lleve a cabo una tríada dinámica (Zapata & Arango, 2007).



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

2.3.1.11 Conjunto de Logro.

Esta estructura utiliza una relación de logro y se complementa con un concepto, tríada dinámica o tríada estructural. Se utiliza para representar los objetivos de la organización o del proceso y las acciones intangibles que se realizan dentro de éste (Zapata & Lezcano, 2009).



III. ANTECEDENTES

La literatura especializada presenta diversos modelos de diálogo, que ayudan a realizar diversas tareas. Dichos modelos de diálogo se pueden categorizar en diálogos para obtener información, diálogos para presentar información y diálogos para simular la relación humano-humano. A continuación, se presenta una breve descripción de algunos de los trabajos consultados.

3.1 Diálogos de obtención de información

Este tipo de modelos de diálogo se utiliza para obtener información del usuario, la cual varía de acuerdo al objetivo que persigue cada trabajo. En este rango, se pueden agrupar los modelos de diálogo para la educación de requisitos de software y los de educación de requisitos de productos. Estos usos son los más representativos que se encuentran en los modelos de diálogo para obtener información. A continuación, se presentan algunas de las propuestas consultadas.

3.1.1 Modelos de diálogo de educación de requisitos de software

Con el fin de asistir al ingeniero de sistemas en el proceso de la entrevista con interesado, para obtener los requisitos de software, Gilvaz y Leite (1995) desarrollan una herramienta que guía al analista en este proceso. Para ello, utilizan un cuestionario pre-establecido y la documentación la presentan basados en el modelo SADT (*Structured analysis and design technique*), desarrollado por Ross (1977). Este modelo permite el tratamiento de sistemas complejos mediante la construcción de diversos elementos conocidos, tales como diagramas de actividades o “*actigramas*” y diagramas de datos o “*datagramas*”, entre otros. De este modo, Gilvaz y Leite (1995) se centran en la



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

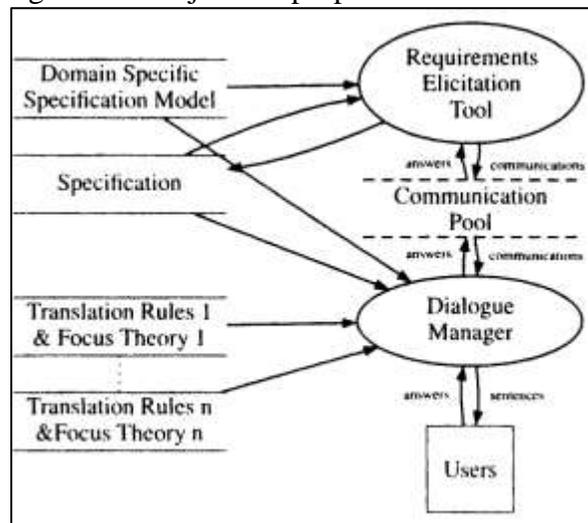
prolongando las sesiones y reduciendo las instrucciones con información oral y escrita. Para validarlo, se aplicó en un grupo de telemercaderistas, consumidores y administradores, a fin de obtener las características de un sistema para telemercadeo. Para ello, se diseñan algunos escenarios, se realizan lluvias de ideas en las que cada participante expresa su conocimiento. Cada grupo crea en un escenario y tiene la libertad de presentar sus ideas y proponer nuevos escenarios utilizando fotografías y otras herramientas audiovisuales. Los analistas transforman todas las ideas recopiladas en requisitos e interfaces personalizadas que los diseñadores pueden convertir en ideas de una aplicación o en innovaciones de las actuales. Sin embargo, esta estrategia no presenta tratamiento computacional para la transformación de la información. Adicionalmente, su uso se limita al conocimiento del analista sobre el dominio del problema y, según lo planteado por los mismos autores, su aplicación es muy limitada.

Lecoeuche *et al.* (1998, 1999) presentan un modelo de diálogo para la obtención de los requisitos de software desde los conocimientos del interesado. En esta propuesta, se encuentra una herramienta de educación de requisitos, que verifica que los datos que ingresa el usuario son correctos y lo guía en la obtención de los requisitos de forma adecuada. Para ello, utiliza un modelo de especificación de requisitos del dominio del problema, que contiene los conocimientos del área y la situación que se especificó para realizar el proceso de educación. El *pool* de comunicaciones controla las interacciones entre el sistema y el usuario. Así, recibe las solicitudes de la herramienta de educación de requisitos y las transmite al gestor de diálogo. Igualmente, recibe las respuestas del usuario y las transfiere en la herramienta de educación. Del otro lado, está el gestor de diálogo, el cual presenta al usuario las comunicaciones de la herramienta de educación de requisitos de forma que emula la comunicación en lenguaje natural. Luego, recibe la información del usuario y la transforma en la notación que entiende la herramienta de educación de requisitos. Para realizar esta transformación, el gestor de diálogo utiliza unas reglas, que se basan en cuatro formas de interacción que se describen como relación directa, relación de especialización, relación de generalización y relación simple.



Este modelo se planteó para que lo utilizara el interesado directamente, pero se limita al dominio de conocimiento de la aplicación. Así, si la información del dominio y la situación a educir no se encuentran en una base de conocimientos, no se puede llevar a cabo el proceso mediante el uso de este modelo. La figura 5 presenta el diagrama de flujo de datos de la solución propuesta por Lecouche *et al.* (1998, 1999).

Figura 5. Diagrama de flujo de la propuesta de Lecouche *et al.* (1998)

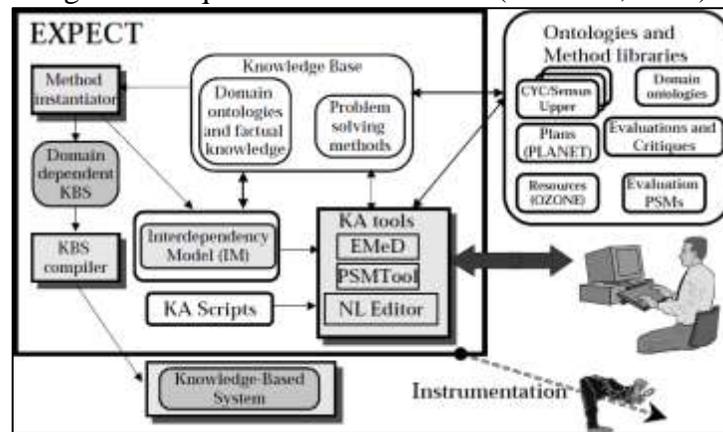


EXPECT es una iniciativa presentada por Gil *et al.* (2000) y Blythe *et al.* (2001), que busca crear una base de conocimientos basada en un sistema de información, para proveer soporte a los ingenieros y los usuarios finales unificando los conocimientos y términos. Esto se realiza con el propósito de hacer más accesible al usuario final los conocimientos técnicos para realizar una especificación de software. EXPECT es una herramienta para obtener el conocimiento de un dominio por medio de preguntas estructuradas. Esta herramienta se basa en las características de la tarea y esta información se almacena en una ontología que conforma una base de conocimientos, la cual puede utilizar posteriormente el analista para comprender el dominio de aplicación del sistema a desarrollar. Otra característica de EXPECT es que permite importar ontologías de otros sistemas para complementar su base de conocimientos. La principal



ventaja de este sistema, es que guía al usuario para ingresar los datos. Aunque, como tal, no es una propuesta para educir requisitos de software, se plantea como una solución para mejorar la información del dominio del problema y proveer un conjunto de conocimientos que faciliten la labor para desarrollar este proceso. La figura 6 muestra la arquitectura de este sistema.

Figura 6. Arquitectura de EXPECT (Gil *et al.*, 2000)

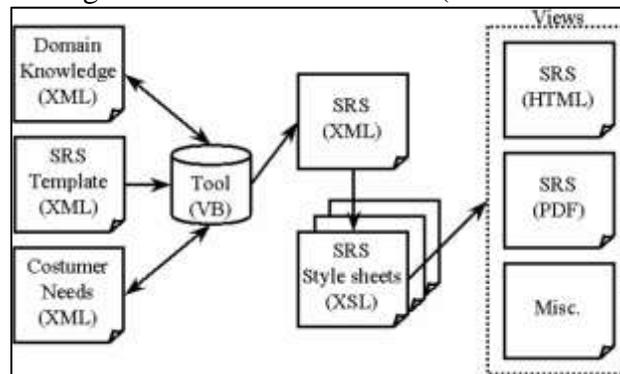


Kato *et al.* (2001) analizan el comportamiento de un experto en el proceso de educación de requisitos y lo emulan mediante una herramienta para realizar este proceso, el cual sirve de guía a un analista novato. La entrevista se realiza de acuerdo con lo que se establece el estándar 830 de la IEEE (1998). Esta propuesta le permite al analista ver el orden en que debe realizar las preguntas. Esta secuencia se estableció de acuerdo con los resultados del seguimiento a los expertos y se encuentra enfocada en el área comercial. Para realizar el proceso, se propone el uso de un tesoro con el conocimiento del dominio del problema y una plantilla con las características que debe tener la especificación de acuerdo con el estándar 830. Las preguntas se presentan al analista, quien introduce las respuestas del interesado y se van generando las diversas preguntas de acuerdo con la evolución de la entrevista. Este trabajo mejora algunas características de los planteamientos de Gilvaz y Leite (1995), Leite y Gilvaz (1996) y Lecouche *et al.* (1998, 1999), pero se mantiene su aplicación en un dominio específico del conocimiento.



Mediante una técnica parcialmente automatizada, Kassel y Malloy (2003) proponen una estrategia que sirve de puente entre el analista y el interesado para la educación de requisitos. Ellos no consideran viable la completa automatización del proceso de educación y especificación de requisitos, pues consideran que abolir la intervención del analista humano es impráctico. Así, proponen una técnica donde el interesado, el experto del dominio y el analista interactúan directamente con el sistema. Para ello, el experto del dominio de la aplicación alimenta la información en una base de datos, con su experiencia en el dominio. Posteriormente, los interesados responden preguntas cerradas, con respuestas del tipo si/no, que se basan en la información del dominio almacenada en la base de datos. Finalmente, el analista utiliza esta información para generar automáticamente los diagramas que harán parte de la especificación de requisitos. Además, se pueden generar documentos en PDF y en XML, que contienen la información que suministra el interesado en un formato cercano al lenguaje natural para su validación. Pese a facilitar la lectura y la validación del discurso, para el uso de esta propuesta se requiere un experto en el dominio de la aplicación a desarrollar y sólo se puede utilizar después de obtener dicha información. La figura 7 muestra la configuración de la herramienta desarrollada para el desarrollo de esta estrategia.

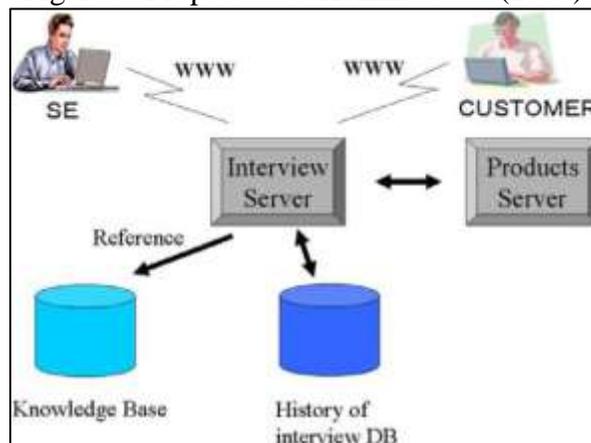
Figura 7. Configuración de la Herramienta (Kassel & Malloy, 2003)





Kozima *et al.* (2005) presentan un modelo de diálogo para guiar la entrevista, con el propósito de realizar una adecuada educación, análisis y descripción de los requisitos de software. Este modelo guía al analista para realizar la entrevista basándose en las reglas del negocio de un dominio específico. Este modelo se implementó en un prototipo bajo ambiente Web, el cual se compone de un servidor de la entrevista, un servidor del producto, una base de conocimientos del dominio del problema y una base de datos que almacena las entrevistas realizadas. El servidor de la entrevista guía el desarrollo de la misma, en tanto que el servidor del producto genera la especificación de requisitos de acuerdo con la información obtenida en la entrevista. De otro lado, la base de conocimientos posee la información de las experiencias de los analistas en el proceso de entrevista del dominio particular. Toda la información generada en la entrevista se lleva a una base de datos que almacena esta información. Para lograr esta interacción se establecieron unas reglas que determinan la secuencia de las preguntas y del diálogo. Esta propuesta aporta el desarrollo del proceso mediante el uso de herramientas Web, pero no se mejora el problema del dominio de aplicación, el cual aún se mantiene restringido. En la figura 8 se muestra la estructura de esta propuesta.

Figura 8. Propuesta de Kozima *et al.* (2005)



Al-Salem y Abu-Samaha (2007), realizan la educación de requisitos para aplicaciones Web utilizando un método que se basa en la técnica de Puntos de Vista para la educación



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

y formulación de los requisitos de software. Para ello, definen aplicaciones Web como sistemas complejos basados en una gran variedad de hardware, software, protocolos, lenguajes, interfaces y estándares. Así, los requisitos se representan con base en los servicios que prestará el sistema a desarrollar. Esto debido a que consideran una aplicación Web como una herramienta para proveer servicios a los usuarios. Para realizar la captura de los requisitos, se establece un cuestionario de preguntas a realizar. Posteriormente, con las respuestas obtenidas, escogen las características comunes y sobre ellas se centra el proceso. El método planteado cumple algunas características del estándar IEEE-830 para la especificación de requisitos. No obstante, la especificación que se obtiene en esta propuesta, ‘no es tan precisa y clara como la que se podría obtener con un método convencional de especificación de requisitos de software’, según los autores. Por otro lado, la especificación de requisitos que se obtiene en esta propuesta no la valida el interesado.

FGDRE (*Focus Group Discussion for Requirements Elicitation*) es una propuesta de Mohd y Salwah (2008), con la cual se realiza el proceso de educación de requisitos, mediante la discusión de las características que debe tener el nuevo sistema. Para realizar este proceso se reúne un grupo de interesados, como una forma de involucrarlos en el proceso. Esta propuesta se enfoca en realizar la especificación de software educativo y su validación se realiza con profesores de colegio, quienes aportan su conocimiento para la obtención de la especificación de la herramienta a desarrollar. Para ello, se utiliza una plataforma con pasos específicos a seguir, que le permite a los grupos participar directamente en todo el proceso. Cada participante tiene un rol asignado y el facilitador guía el proceso, asegurando la participación y comunicación de todos los integrantes del grupo que utilizan. Cuando un punto de vista se identifica, se debate para desarrollarlo y, posteriormente, se incluye en la propuesta. Todo esto, le permite los interesados entender la importancia de la actividad de educación de requisitos. Pese a la activa participación del interesado en el desarrollo de los requisitos, lo que se podría



Mediante un juego, Zapata y Giraldo (2009) proponen destacar la relevancia del proceso de educación de requisitos. Así, presentan una pequeña entrevista entre el analista y el interesado, con el propósito que el jugador entienda las funciones de cada rol. Para ello, el participante obtiene unas palabras, las cuales son las respuestas del interesado a la entrevista, con las cuales debe armar el discurso en UN-Lencep y, posterior a ello, el respectivo EP. Con el EP completo, el moderador del juego revisa que exista consistencia entre éste, el discurso y la entrevista planteada. Gana quien complete, en la forma más consistente, el diálogo y el EP. En este artículo se ofrece un aporte a la enseñanza de los estudiantes de ingeniería. Sin embargo, en el modelo de diálogo presentado, la interpretación de la entrevista y la generación del discurso las realiza el jugador, pues no se plantea una propuesta para realizar estos procedimientos.

3.1.2 Modelos de diálogo de educación de requisitos de productos

Villaseñor *et al.* (2004) proponen una técnica en la que el usuario puede expresarse de forma multimodal con indicaciones realizadas por medio de un cursor gráfico asociado al ratón para realizar el diseño de una cocina. Para ello, utilizan una ontología propia del dominio, para la interpretación del discurso, con un corpus multimodal recolectado por medio de la técnica del mago de Oz. El usuario puede expresarse de forma multimodal, con un diálogo en lenguaje natural, mediante un agente conversacional simulado, conjuntamente con indicaciones que realiza un cursor gráfico asociado con el ratón. De esta forma, se realiza el diseño. Las acciones permitidas son sencillas, como colocar objetos, moverlos o eliminarlos. El modelo de diálogo propuesto se compone de varios subsistemas: reconocimiento de habla, comprensión del lenguaje y gestión del diálogo, entre otros. Esta propuesta se enfoca en el diseño de una cocina y permite definir las ubicaciones de cada uno de los elementos que la conformarán, aunque es de resaltar que su aplicación sólo se da en este campo.



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

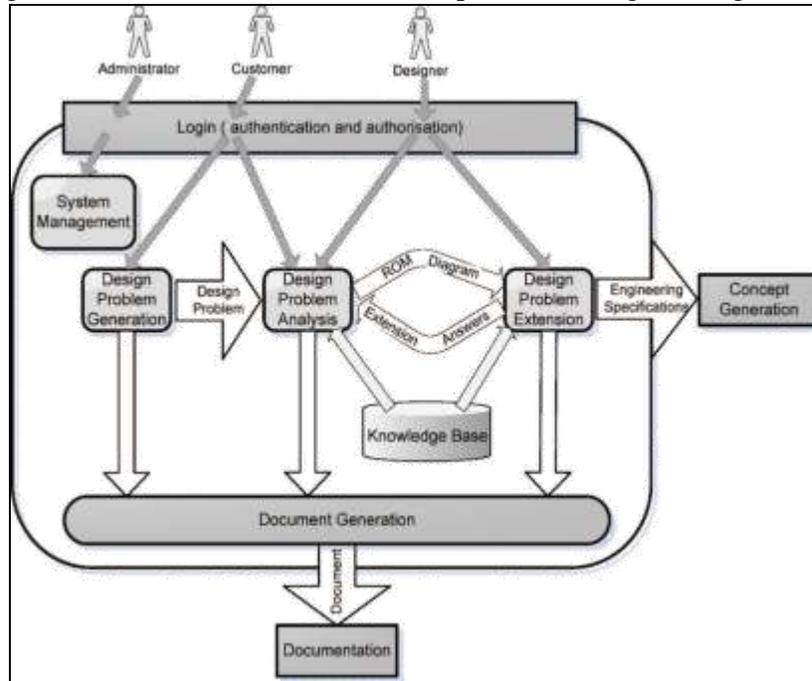
William A. Arévalo Camacho

Producir equipos médicos con interfaces amigables que faciliten su uso, ayuden a reducir los accidentes durante su manejo y mejoren la interacción con los usuarios, es uno de los objetivos de Garmer *et al.* (2004). Para ello, presentan una propuesta que involucra a los usuarios finales en la especificación de las interfaces de equipos médicos, mediante el uso de entrevistas, individuales y en grupos. Así, se logró identificar aspectos importantes de usabilidad, para mejorar el diálogo entre los usuarios finales y los equipos desarrollados. Esta propuesta se probó en dos hospitales, uno grande y otro pequeño, en el desarrollo de respiradores artificiales para la unidad de cuidados intensivos. Así, se tomó un grupo de enfermeras con una experiencia superior a año y medio e inferior a nueve años, a las cuales se les realizó una entrevista de acuerdo con su experiencia. Sin embargo, la estrategia presentada se enmarca en un dominio particular. Adicionalmente, no se establece el tratamiento computacional de la información obtenida.

Wang y Zeng (2009) proponen un método para la educación de los requisitos y diseño de un producto, basados en el análisis lingüístico de las respuestas, el cual transforma las respuestas a un lenguaje gráfico llamado ROM (*Recursive Object Model*). Para ello, se hacen dos tipos de preguntas, una se enfoca en obtener los elementos de la estructura del ROM y la otra en obtener información específica del dominio del producto y, mediante una plantilla, se determina el orden de las preguntas. Este proceso inicia con una descripción en lenguaje natural del diseño deseado. En consecuencia, se define el proceso de diseño como orientado al interesado y el proceso de educación de requisitos como orientado al analista con la generación de descripciones estructuradas y formales. La figura 10 presenta la estructura de esta propuesta.



Figura 10. Modelo de educación de requisitos (Wang & Zeng, 2009)



El proceso de la propuesta de Wang y Zeng (2009) se divide en ocho pasos, primero: el diseñador transforma el discurso en lenguaje natural al diagrama ROM con la herramienta de análisis. Esto le facilita al diseñador entender mejor el problema. Segundo: el diseñador analiza los objetos del diagrama ROM e identifica los elementos que necesitan aclaración. Luego, aplicando unas reglas y la plantilla de preguntas, obtiene las preguntas a realizar para aclarar las dudas. Tercero: el diseñador recoge las respuestas de las preguntas generadas, para lo cual utiliza una base de conocimientos, internet o directamente del interesado. Cuarto: se repite desde el primero hasta el cuarto paso hasta concluir las preguntas y se genera un nuevo diagrama ROM que el diseñador comprende en su totalidad. Quinto: con el propósito de obtener una mejor descripción del dominio del problema y su relación con los objetos del diagrama ROM, se generan unas preguntas específicas del dominio y su orden lo define unas reglas predefinidas. Sexto: este paso es similar al tercero, pero con las preguntas generadas en el paso anterior. Séptimo: se repiten los pasos desde el primero hasta el séptimo, para garantizar



que los requisitos del producto que dependen del dominio se obtengan adecuadamente. Octavo: se actualiza el diseño con la descripción del problema. No obstante, esta propuesta requiere abundante intervención del analista. Asimismo, se necesita obtener información del dominio tanto del interesado como de otras fuentes. Sin embargo, no es claro el manejo de las preguntas relativas al dominio, pues se entiende que las preguntas se realizan de acuerdo con la interpretación del analista.

3.2 Diálogos de presentación de información

Los modelos de diálogo para la presentación de información utilizan una breve interacción con el usuario para determinar la información requerida. Así, se presenta la información de acuerdo con las características establecidas. En la literatura especializada se encuentran dos formas de presentar la información, en forma gráfica (o textual) y hablada. Los siguientes son algunos de los trabajos consultados.

3.2.1 Diálogos computacionales para presentar información gráfica o textual

Hwang *et al.* (2006) presentan CORVIS (*Comics Rendering system on Video Stream*), el cual permite la obtención de caricaturas desde una cinta de video, de forma semiautomática. Para ello, inicialmente se escogen, manualmente, las escenas que se desean caricaturizar. Éstas se transforman en ilustraciones y se les agregan efectos de caricatura como son líneas de velocidad, guías de trayectoria, imágenes de fondo y globos de texto para representar el diálogo entre los actores. Además, se pueden incluir palabras que imitan los sonidos originales de la película. De esta forma, se realiza un corte de comics de acuerdo a la secuencia de escenas seleccionadas. La figura 11 presenta la arquitectura de CORVIS. Un aporte similar lo presentan Caglioti *et al.* (2009), quienes proponen una técnica para obtener máscaras de movimiento de una



forma muy similar a la propuesta de Hwang *et al.* (2006). Estas técnicas producen una secuencia de imágenes con ciertas características, las cuales se presentan al usuario. Sin embargo, estas propuestas sólo se pueden aplicar en la generación de imágenes.

Figura 11. Arquitectura de CORVIS (Hwang *et al.*, 2006)



Con el propósito de ayudar a los pacientes a entender el lenguaje técnico del médico, Williams *et al.* (2007) proponen una técnica para presentar esta información en un lenguaje más simple para el paciente. Para ello, Utilizan plantillas que generan diálogos o monólogos, según la preferencia del usuario. Así, el usuario escoge el término que desea conocer y la forma de ver la información. Este término se explica mediante su inclusión en un discurso. Si el usuario escoge un monólogo, el sistema le presenta una enfermera jefe que le explicará el término. No obstante, si decide que sea un diálogo, el sistema le presenta una conversación entre la enfermera jefe y una enfermera novata, donde la enfermera jefe le explica el significado del término a la enfermera novata. El sistema genera automáticamente el discurso, consultando una base de datos para mostrar la información solicitada. Este sistema permite generar un discurso desde un término, de acuerdo con una plantilla preestablecida. Sin embargo, el discurso es estático y siempre que se solicite información sobre la misma palabra se mostrará la misma información. Además, sigue vinculado a un dominio específico, para este caso la medicina.



Prendinger *et al.* (2007) presentan un método para la generación automática de contenidos desde texto usando RST (*Rhetorical Structure Theory*) para descomponer el texto e identificar sus relaciones. De esta forma, crean diálogos del tipo pregunta-respuesta que ayudan a explicar o entender mejor un determinado tema. Esta propuesta se presenta como una estrategia para usar en *E-Healthcare* e *E-Learning*, pues los textos se pueden transformar en diálogos entre el paciente y el farmacéuta o entre el estudiante y el profesor. Así, se facilita al usuario la comprensión de un determinado texto. No obstante, su uso se limita en el dominio de aplicación, pese a la generación del discurso.

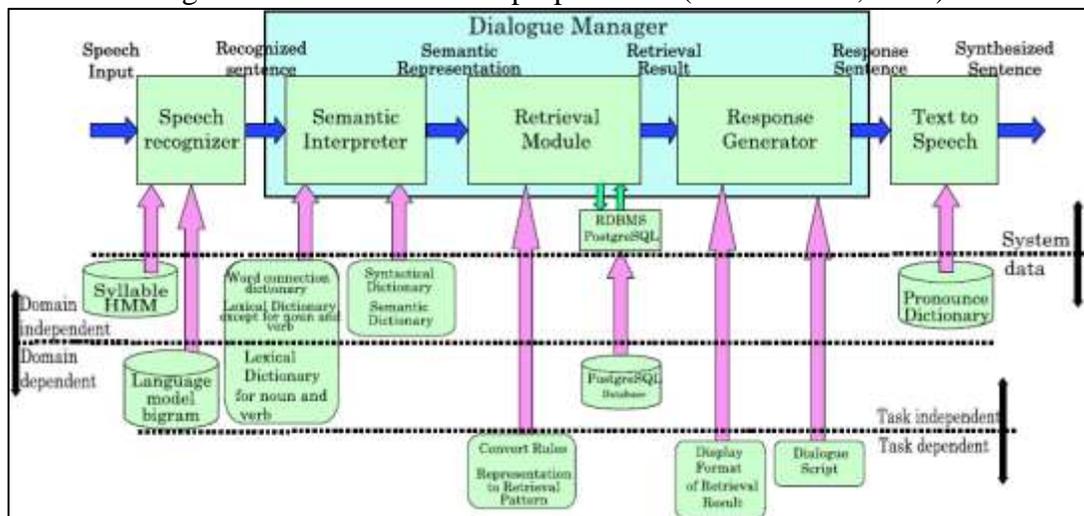
3.2.2 Diálogos computacionales para presentar información hablada

Un intérprete semántico con un generador de respuesta cooperativo para un sistema de diálogo multimodal, que posee entradas y salidas de voz y gráficas, lo presentan Nakagawa *et al.* (2000). El sistema de diálogo entiende el lenguaje natural, con algunos fenómenos ambiguos, como interjecciones, palabras omitidas, alteración del orden de las palabras, palabras desconocidas y otras características. Algunas expresiones no se analizan, debido al no reconocimiento causado por la incompletitud de expresiones o la ausencia de información en la base de datos y del intérprete semántico. Por ello, si la consulta de un usuario no se puede responder, el gestor de diálogo realiza preguntas al usuario a fin de obtener las condiciones necesarias para seleccionar una respuesta efectiva. Además, si el sistema no puede recuperar cualquier información relacionada con la pregunta del usuario, el generador de respuesta cooperativa en el sistema de diálogo deberá proponer un plan alternativo. Con esta propuesta se aclara el discurso de un usuario, a fin de que el sistema pueda determinar efectivamente lo que el interesado desea y se presenta la información tanto hablada como por pantalla en mandarín. Aunque se utiliza para presentar información, también se puede catalogar como un sistema para mejorar la interacción humano-máquina.



El uso de los sistemas de reconocimiento de voz se viene incrementando, pero su aplicación industrial es costosa. Por ello, Umeda *et al.* (2003) proponen un sistema portátil para el reconocimiento de voz, el cual utiliza la misma estructura de la propuesta de Nakagawa *et al.* (2000), pero se realiza una mejora a tres módulos: el intérprete semántico, el módulo de recuperación y el generador de respuestas. La figura 12 muestra la estructura básica de estas propuestas, el cual utilizan Suzuki *et al.* (2004) para generar un sistema de reservas en un hotel.

Figura 12. Estructura del la propuesta de (Umeda *et al.*, 2003)

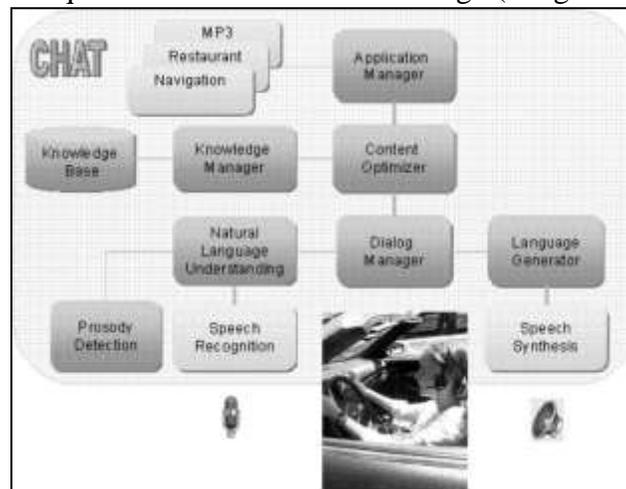


Varges *et al.* (2008) desarrollan estrategias para la selección y presentación de información precisa. Se enfocan en un computador de abordaje en un carro, que permite la selección de una acción simple desde una lista concreta de tareas disponibles, mediante un diálogo de reconocimiento de voz con el usuario. De este modo, se logra que el conductor centre su atención en la conducción del vehículo y se evita la distracción en la ejecución de las tareas definidas. En esta propuesta, se explora la relación entre las técnicas de respuesta a pregunta y sus restricciones en sistemas de diálogo hablado. Además, se desarrollan estrategias para seleccionar y presentar información de forma



precisa. Así, se facilita la selección de restaurantes y otras tareas de navegación. Además, ayuda en la selección de música en el reproductor de MP3, para lo cual se basa en un sistema de diálogo tipo *chat*. Pese a las características del diálogo, éste se centra en una tarea específica para la selección de un único elemento de una serie más amplia presentada. La figura 13 muestra la arquitectura del modelo propuesto.

Figura 13. Arquitectura del Modelo de Diálogo (Varge *et al.*, 2008)



3.3 Simulación de la relación humano-humano

Los sistemas de simulación del diálogo humano-humano, buscan disminuir el impacto de la interacción con la máquina, haciéndole creer al usuario que está interactuando con otro humano. En este ítem se encuentran, principalmente, tres usos relevantes, asistenciales y de gestión de información, apoyo psicológico y realización de tareas o funciones. A continuación, se presentan algunas de las propuestas consultadas, según la clasificación dada.



3.3.1 Asistenciales y de gestión de información

Con el fin de atender las peticiones y consultas de los clientes de un restaurante, se desarrolla un sistema de diálogo de reconocimiento del habla en lenguaje natural, que simula la interacción con un empleado. SAPLEN (Sistema Automático de Pedidos en Lenguaje Natural) se considera un sistema experto basado en reglas y guiado por objetivos. Para su elaboración, se tomaron como base algunos corpus recolectados de situaciones reales en un restaurante. Con el sistema, se pretende emular un comportamiento similar al del empleado al elaborar un pedido. Los módulos principales del sistema son: la interfaz de entrada, el módulo de control y la interfaz de salida. La interfaz de entrada recibe las consultas del usuario por teclado y se realiza la interpretación semántica. El módulo de control recibe la interpretación semántica de la interfaz de entrada y envía a la interfaz de salida el nuevo estado de la consulta. La interfaz de salida envía a la pantalla, en lenguaje natural, la información recibida del módulo de control. (López-Cózar & Rubio, 1997) (López-Cózar *et al.*, 2000). Esta propuesta permite una mejor interacción con el usuario y facilita la obtención de nueva información, pues cuando no se encuentra la información se genera un diálogo para adquirir este conocimiento. Sin embargo, se encuentra limitada en el dominio de aplicación.

Un *sofbot* (Robot de Software) de charla llamado SAMI (Sistema Asistente Modelo Inteligente) que utiliza la técnica de RBC (Razonamiento Basado en Casos) es la propuesta de Camacho (2004). El contexto de aplicación de las conversaciones de SAMI, son los temas relacionados con el aula virtual. De esta forma, se constituye como un asistente virtual que brinda ayuda a los usuarios. La interacción se realiza en lenguaje natural, mediante un diálogo escrito entre el usuario y el sistema, de tal forma que el usuario piense que está hablando con otra persona. El *sofbot* analiza el diálogo y busca en su base de conocimientos la respuesta más adecuada posible utilizando la técnica RBC. La estructura de SAMI presenta un módulo de ingreso de la información, un



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

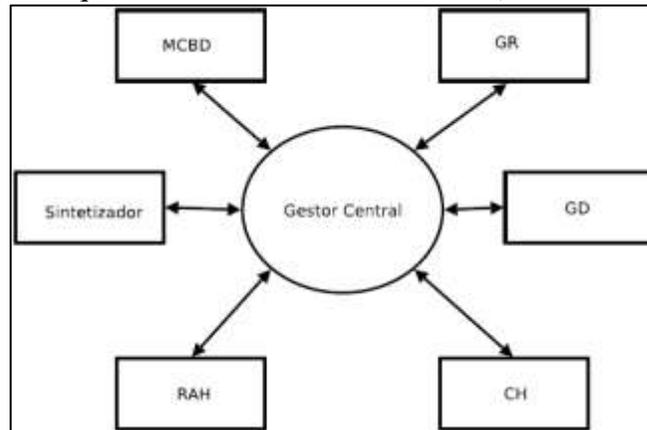
motor de búsqueda, una capa de acceso a los datos, una base de conocimientos y una interfaz de obtención de conocimiento. La base de conocimientos centraliza toda la información necesaria para que SAMI pueda contestar a las preguntas de los usuarios. La capa de acceso a datos sirve de interfaz de accesos a la base de conocimientos. El motor de búsqueda encapsula la funcionalidad de tomar las preguntas de los usuarios, recupera la información de la base de conocimientos y devuelve la respuesta. La interfaz de usuario pasa las preguntas del usuario al sistema y luego presenta la respuesta. El módulo de ingreso de conocimiento es el maestro de SAMI y permite administrar la base de conocimientos. Además, cuenta con reportes estadísticos que le permiten ver el estado de la base de conocimientos. Esta propuesta se enfoca en un dominio particular del conocimiento y requiere una base de datos del dominio de aplicación. Sin embargo, el autor propone que se utilice en otros campos como *E-Learning*.

DIHANA realiza el estudio y desarrollo de un sistema robusto de diálogo modular y distribuido para el acceso a sistemas de información. De esta forma, se proporciona información en lenguaje natural sobre servicios, horarios y precios de trenes de largo recorrido y en español. DIHANA presenta los siguientes módulos: el módulo de reconocimiento automático del habla, que identifica la secuencia de palabras que se ajusta a la señal de voz de entrada; el módulo de comprensión, que extrae el significado de las palabras reconocidas y lo expresa en términos de un lenguaje semántico específico para la tarea; el módulo de diálogo, que decide el paso a seguir en el sistema tras cada intervención del usuario, basado en la interpretación semántica generada en el módulo anterior; el módulo de acceso a la base de datos de la aplicación, que recibe las peticiones de consulta a la base de datos del gestor de diálogo, las procesa y le devuelve el resultado; el módulo de generación de respuestas de usuario, que elabora la frase gramaticalmente correcta y en un lenguaje lo más cercano posible al lenguaje natural, mediante el uso de plantillas de respuestas adaptadas al tipo de consulta. La figura 14 muestra la arquitectura del sistema (Griol *et al.*, 2005) (Hurtado *et al.*, 2005) (Griol *et al.*, 2006). Este sistema realiza el reconocimiento de voz y le presenta al usuario la



información de acuerdo con las consultas realizadas en la base de datos, con una estructura similar a la propuesta de López-Cózar y Rubio (1997) y López-Cózar *et al.* (2000).

Figura 14. Arquitectura del sistema DIHANA (Hurtado *et al.*, 2005)



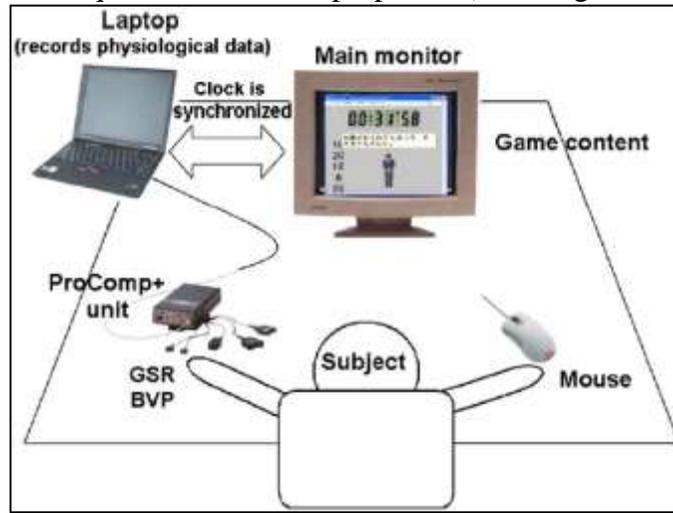
3.3.2 Apoyo psicológico.

La interacción con sistemas informáticos genera preocupación y estrés en los usuarios. Por ello, Prendinger *et al.* (2005) proponen un juego de matemáticas que, en su interfaz gráfica, presenta un personaje animado de nombre Shima, el cual posee características de comportamiento similares a las de una persona. Shima posee la capacidad de hacer gestos, sonreírle al usuario y hacer una venia que los japoneses reconocen como una disculpa. De esta forma, se ayuda a disminuir la tensión del usuario al momento de interactuar con el sistema para responder el cuestionario que evalúa el tema. Por otro lado, el sistema captura imágenes del usuario para determinar su estado anímico y genera gestos en Shima, como respuesta al usuario. Esta metodología se validó con cerca de 20 estudiantes con edades superiores a los 24 años, que hablan japonés nativo, idioma en el que se enfoca esta herramienta. La figura 15 muestra el esquema del modelo de



esta propuesta. Este trabajo es un aporte importante en la relación humano-máquina, pero es un modelo de diálogo restringido a un dominio y acciones particulares.

Figura 15. Esquema del modelo propuesto (Prendinger *et al.*, 2005)

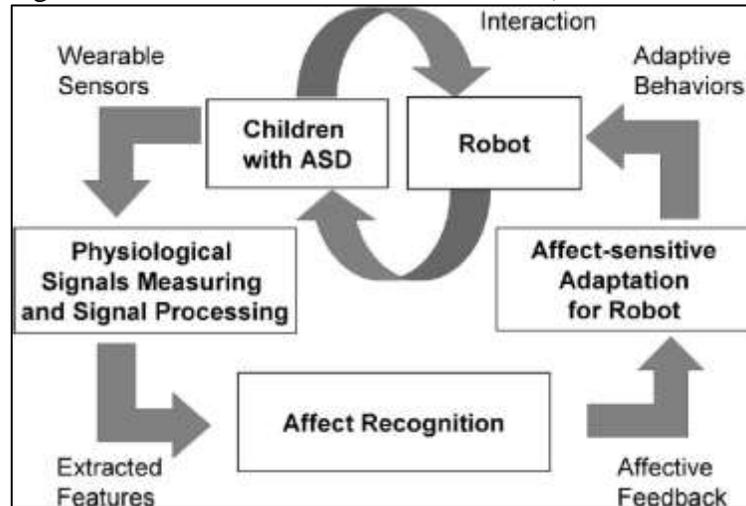


González (2006; 2008) presenta *Social Behavior Simulator*, una herramienta de diálogo para valorar la influencia de la interacción con herramientas automáticas de charla en los humanos. Para ello, se basan en datos de una persona real, de las estrategias y técnicas que utilizan los participantes cuando se encuentran en una situación social. Con esta información se pretende emular esta interacción y que el usuario no perciba que se encuentra dialogando con una máquina. Esto se hace con el fin de estudiar el modo en el que una persona se comporta cuando interactúa con otra. No obstante, el foco de atención no es el programa, sino la forma como la persona interactúa y se busca que el usuario tenga un comportamiento natural, es decir, que elija las opciones como lo haría en caso de encontrarse ante un ser humano real. Esta aplicación se proyecta para la simulación con pacientes reales para probar posibles intervenciones y para enseñar a los psicólogos un comportamiento social anómalo de ciertas patologías. Sin embargo, sólo se aplica a este ambiente.



Generar terapias de asistencia a niños con síndrome ASD (*Autism Spectrum Disorders*) con el uso de herramientas automáticas de intervención afectiva sensible, es una propuesta de Liu *et al.* (2008). Para ello, se diseñan modelos afectivos como terapeuta de los niños con ASD basados en sus respuestas fisiológicas. La interacción de niños con ASD y robots contribuye en la interacción social, pues los robots se utilizan para enseñar habilidades básicas de interacción social, mediante la imitación y la participación en juegos, lo cual fomenta la interacción con compañeros y adultos. La detección automática de señales afectivas es importante, pues la investigación sugiere que las personas tienden a interactuar con las computadoras como si lo hicieran con otras personas, siempre que la tecnología se comporte de una manera socialmente competente. La figura 16 muestra la estructura de la propuesta.

Figura 16. Estructura del modelo afectivo (Liu *et al.*, 2008)



La propuesta de Liu *et al.* (2008) es el desarrollo de modelos afectivos para niños con ASD por medio de una técnica de reconocimiento basada en la psicología de la interacción con sistemas. Esta propuesta permite identificar demostraciones afectivas de los usuarios y genera respuestas de la misma índole, como una forma de mejorar la



relación social del usuario. Sin embargo, su enfoque se limita sólo al estímulo de niños con ASD.

3.3.3 Realización de tareas o funciones

Un sistema de visión que le permite a un robot manipulador obtener información de la localización de un objeto para realizar tareas de ensamble en tiempo real, es la propuesta de Peña *et al.* (2002). Utilizar técnicas, como lógica difusa y redes neuronales, simplifica el manejo de modelos matemáticos complejos que requieren bastante tiempo en su implementación. En esta propuesta se utilizan redes neuronales artificiales en conjunto con una base de conocimientos dinámica. La parte medular en la celda es un brazo-robot de seis grados de libertad que realiza tareas de ensamble y obtiene información sensorial de reconocimiento de voz, visión y sensado de fuerza. La arquitectura del sistema se forma con una maestra industrial en donde residen la tarjeta de sensado de fuerza, el software de reconocimiento de voz y el controlador de la red neuronal y en otra computadora dedicada se implementa el sistema de visión. El software de reconocimiento de voz instruye el robot para los movimientos y velocidad. El sensado de fuerza provee al controlador de la red neuronal la información requerida para aprender la operación de ensamble en línea. El sistema de visión proporciona al brazo-robot, la información espacial de la localización y captura de las partes. Esta propuesta permite una interacción multimodal, pero su dominio de acción no entra en el alcance de esta propuesta.

Kruijff *et al.* (2007) y Zender y Kruijff (2007) utilizan un diálogo controlado para definir ubicaciones espaciales específicas, a partir de preguntas estructuradas que le permiten, a una plataforma robótica móvil autónoma, definir cuál es la tarea, dónde realizarla y quién la solicitó. Para ello, el robot utiliza un mapa, una ontología de objetos y su ubicación y un módulo de comunicación. Adicionalmente, el robot puede aumentar el



mapa de manera autónoma, mediante métricas adquiridas con información cualitativa sobre los lugares y objetos en el ambiente. Para el módulo de comunicación se tomaron varios corpus mediante la estrategia de mago de Oz. La comunicación tiene un enfoque basado en la ontología que proporciona un terreno común para el diálogo humano-robot, con el cual establecen referencias a áreas espaciales, de acuerdo con el diálogo. Adicionalmente, este modelo también permite presentar información tomando como base los conocimientos adquiridos. De esta forma, el robot puede responder a preguntas de ubicación y no sólo cumplir tareas. Sin embargo, su dominio de aplicación se limita al establecido en la propuesta y la obtención de la información se limita a las características de este dominio.

3.4 Análisis Crítico de la literatura consultada

En la literatura consultada se encuentran diversos modelos de diálogo, los cuales ofrecen diferentes alternativas. Sin embargo, en la educación de requisitos, la cual es el área de esta Tesis, se encuentra que la mayoría de los trabajos proponen guiar al analista en el proceso de la entrevista. De esta forma, se mejora el proceso y se disminuyen errores, especialmente con analistas novatos. Así, en un gran número de trabajos se plantea la definición de un modelo de entrevista, con preguntas establecidas de acuerdo con el dominio de aplicación.

Una de las limitantes que se encontró se refiere a la aplicación de estos modelos en dominios específicos. Esto se debe a la ambigüedad con la que se suele expresar el interesado y a los diferentes errores de comunicación que existen en el diálogo entre humanos, los cuales son más difíciles de entender para la máquina, pese a las diferentes técnicas de PLN. Por ello, para superar este inconveniente, la mayoría de los trabajos incluyen en su modelo una base de conocimientos del dominio del problema, la cual se usa en la desambiguación y la resolución de diferentes problemas de comunicación. Sin embargo, pese a que estas bases de conocimientos pueden contener información de diversos



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

dominios, el trabajo de concentrar esta información en ellas, es un trabajo dispendioso. No obstante, otra alternativa encontrada como solución a esta dificultad es el uso de lenguajes controlados, aunque algunos pueden ser difíciles de comprender o requieren ciertos conocimientos técnicos para su lectura.

De otro lado, la literatura consultada no presenta una estrategia clara de validación del interesado, en relación con la información resultante al concluir la tarea de educación de requisitos. Realizar esta validación ayuda a identificar y disminuir los errores en una etapa temprana, lo que se traduce en una disminución de costos, tiempo y esfuerzo para las demás etapas del desarrollo. Las propuestas revisadas se centran en obtener la información, basadas en los conocimientos del interesado. Sin embargo, es relevante que el resultado del proceso lo valide quien posee los conocimientos y el criterio para ello y esta persona es el interesado. No obstante, son pocas las propuestas que son para uso del interesado y lo involucran directamente en el proceso. Por el contrario, la mayoría de propuestas se presentan como una estrategia para ayudar al analista a realizar su labor. Así, se mejora el proceso y se cuenta con el criterio de éste, para llevar a cabo de forma más eficiente la educación de requisitos.

Del mismo modo, en algunas propuestas se establece el uso de reglas para el desarrollo de la estrategia, la obtención de la información y la transformación de la misma en la especificación, aunque la mayoría de ellas no describe de forma clara las reglas que usan y otras se centran en las reglas de uso de la estrategia. Sin embargo, realizar la transformación de forma automática de la información en la especificación y el diseño de la propuesta contribuye con la completitud y consistencia de la información, entre las necesidades del interesado y las interpretaciones del analista en la especificación. En la tabla 3 se puede apreciar un consolidado de los trabajos consultados y sus características.



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

Wiliam A. Arévalo Camacho

Tabla 3. Análisis de los modelos de diálogo en Educación de Requisitos

Propuesta	Dominio de Aplicación	Lenguaje Controlado	Uso de Reglas	Valida el Interesado	Aporte Analista
FAES.	Particular	SADT	Si	Si	Exclusivo
Millard <i>et al.</i> , 1998.	Particular	No	No	No	Exclusivo
Lecoecuche <i>et al.</i> , 1999.	Particular	No	Si	Si	Mínimo
EXPECT.	Particular	No	No	No	Alto
Kato <i>et al.</i> , 2001.	Particular	No	Si	Si	Exclusivo
Kassel & Malloy, 2003.	Particular	XML	Si	Si	Medio
Kozima <i>et al.</i> , 2005.	Particular	No	Si	No	Alto
Al-Salem y Abu-Samaha, 2007.	Particular	No	No	No	Alto
FGDRE.	Particular	No	No	Si	Alto
Shahidi & Mohd, 2009.	Particular	No	No	No	Exclusivo
Zapata & Giraldo, 2009.	General	UN-Lencep	No	No	Exclusivo
Villaseñor <i>et al.</i> , 2004.	Particular	No	No	Si	Mínimo
Garmer <i>et al.</i> , 2004	Particular	No	No	Si	Exclusivo
Wang & Zeng, 2009.	General	ROM	Si	Si	Exclusivo



IV. DEFINICION DEL PROBLEMA DE INVESTIGACION

Consecuentemente con la literatura consultada, los problemas que afectan la comunicación y la relación analista-interesado interfieren en el entendimiento y en el desarrollo del proceso de educación de requisitos. Para solucionar estos problemas se encuentran propuestas que disminuyen su impacto y mejoran el proceso. No obstante, no se logra una solución efectiva, pues se deben sacrificar algunos elementos a fin de obtener los resultados esperados.

Pese a los diferentes avances tecnológicos y a los esfuerzos de la comunidad científica para mejorar la relación analista–interesado y disminuir los defectos en la comunicación, estos persisten y generan errores en la especificación (Shahidi & Mohd, 2009). De otro lado, a estos problemas de comunicación se suma la dificultad del interesado para expresar sus necesidades pues, en algunas ocasiones, no poseen una idea clara de lo que desea, son poco específicos o introducen información innecesaria en su discurso. Todo esto, genera especificaciones erróneas que se convierten en defectos que pueden producir sobrecostos en el desarrollo de una aplicación (Leffingwell & Widrig, 2003). Adicionalmente, estos defectos se incrementan en la relación humano-máquina, debido a la dificultad que representa para la máquina resolver ciertas características del diálogo como anáforas, elipsis, composición y ambigüedades, entre otras (Villaseñor *et al.*, 2004).

4.1 Limitaciones encontradas

El análisis del capítulo III presenta algunos de los usos y características de los modelos de diálogo, los cuales se agrupan de acuerdo a estos elementos. Así, se encuentran diálogos para obtención de información (Gilvaz & Leite, 1995) (Leite & Gilvaz, 1996) (Millard *et al.*, 1998) (Lecoeuche *et al.*, 1998; 1999) (Gil *et al.*, 2000) (Blythe *et al.*, 2001) (Kato *et al.*, 2001) (Kassel & Malloy, 2003) (Kozima *et al.*, 2005) (Al-Salem & Abu-Samaha, 2007)



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

(Mohd & Salwah, 2008) (Shahidi & Mohd, 2009) (Zapata & Giraldo, 2009) (Villaseñor *et al.*, 2004) (Garmer *et al.*, 2004) y (Wang & Zeng, 2009). Otra clasificación son los diálogos de presentación de información (Hwang *et al.*, 2006) (Caglioti *et al.*, 2009) (Williams *et al.*, 2007) (Prendinger *et al.*, 2007) (Nakagawa *et al.*, 2000) (Umeda *et al.*, 2003) y (Varges *et al.*, 2008). Por último, se propone la simulación de la relación humano-humano (López-Cózar & Rubio, 1997) (López-Cózar *et al.*, 2000) (Camacho, 2004) (Griol *et al.*, 2005) (Hurtado *et al.*, 2005) (Griol *et al.*, 2006) (Prendinger *et al.*, 2005) (González, 2006; 2008) (Liu *et al.*, 2008) (Peña *et al.*, 2002) (Kruijff *et al.*, 2007) y (Zender & Kruijff 2007). A continuación se presenta un compendio de las limitaciones encontradas en estos trabajos.

Mantener la consistencia y completitud entre el discurso del interesado y la especificación de la información es uno de los objetivos por alcanzar en los proyectos estudiados. Para ello, se propone realizar la educación de requisitos directamente con el interesado. Sin embargo, la mayoría de ellos no presentan una estrategia clara de validación, del interesado, de la información resultante. De otro lado algunos trabajos dejan en manos del analista la interpretación de las necesidades, desde la información obtenida en la entrevista. Así, la especificación de requisitos se realiza de forma semiautomática o, en algunos casos, manualmente. Sin embargo, se encuentran trabajos que implementan el uso de reglas, para la transformación automática de la información obtenida, en la especificación de requisitos, aunque no se presenta como una tendencia general en el proceso.

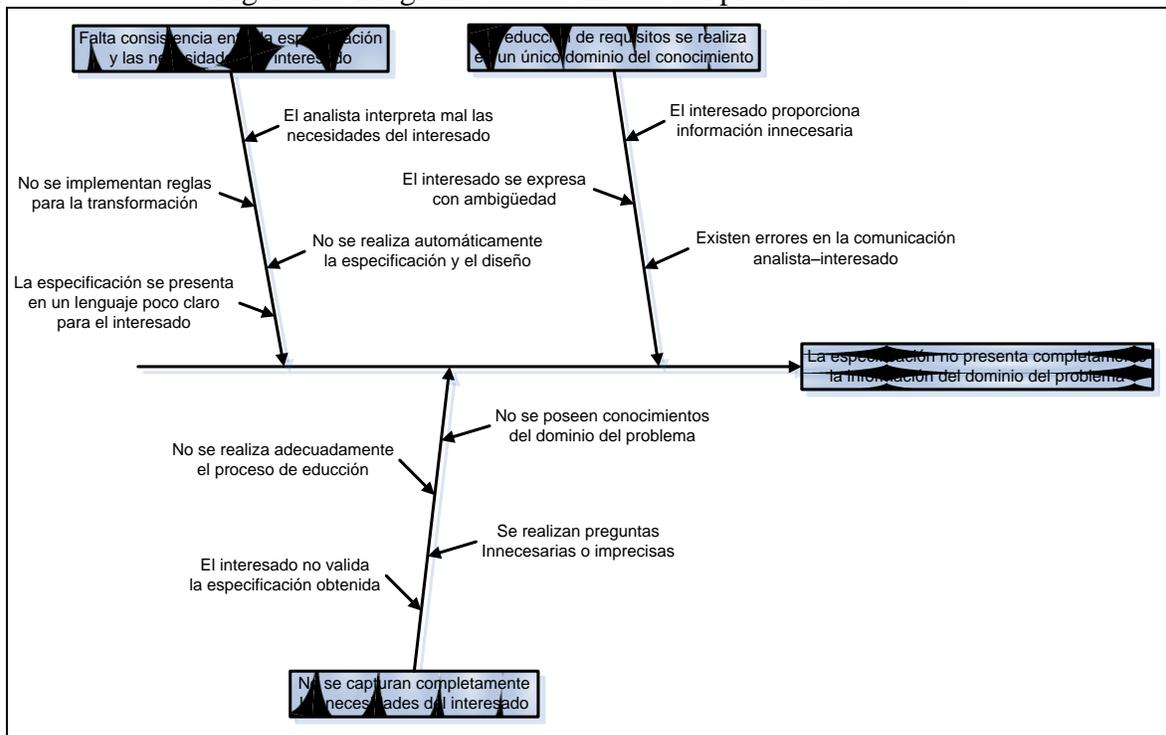
La principal limitación encontrada, en la literatura especializada, se refiere al dominio de aplicación. Esto, se debe a la dificultad en la resolución del discurso del interesado, de forma inambigua, clara y concisa, a causa de los problemas inherentes al lenguaje natural. De esta forma, es común utilizar una base de conocimientos con la información del dominio del problema. Así, se ayuda a la máquina a encontrar y resolver algunos defectos del diálogo, mejorando la interacción con el interesado. Adicionalmente, se debe establecer un formato y secuencia de preguntas para evitar que el interesado entregue información innecesaria y lograr un discurso concreto y coherente. No obstante, algunas propuestas



incluyen el uso de lenguajes controlados, como una forma de hacer más clara la información para la máquina, aunque estos lenguajes, usualmente, no son muy claros, ni fáciles de entender para el interesado.

Adicionalmente, por las restricciones del dominio de aplicación de una propuesta, no se suelen obtener completamente las necesidades del interesado, ya que la interacción con los modelos estudiados depende de diferentes experiencias almacenadas en la base de conocimientos. En la figura 17 se propone un diagrama causa-efecto que detalla las limitaciones encontradas en la literatura consultada.

Figura 17. Diagrama Causa-Efecto del problema estudiado



Todo lo anterior conduce a especificaciones incompletas y defectos en los productos desarrollados. Adicionalmente, algunas propuestas requieren el acompañamiento de un experto del dominio del problema o que esta experiencia se almacene en la base de



conocimientos. Todo esto, ocasiona que el proceso no se lleve de forma adecuada y no se logre obtener completamente las necesidades del interesado.

4.2 Objetivos

4.2.1 Objetivo General

Generar un método para, desde un diálogo controlado con el interesado durante la fase de educación de requisitos, obtener la especificación de requisitos del software, en un discurso en UN-Lencep.

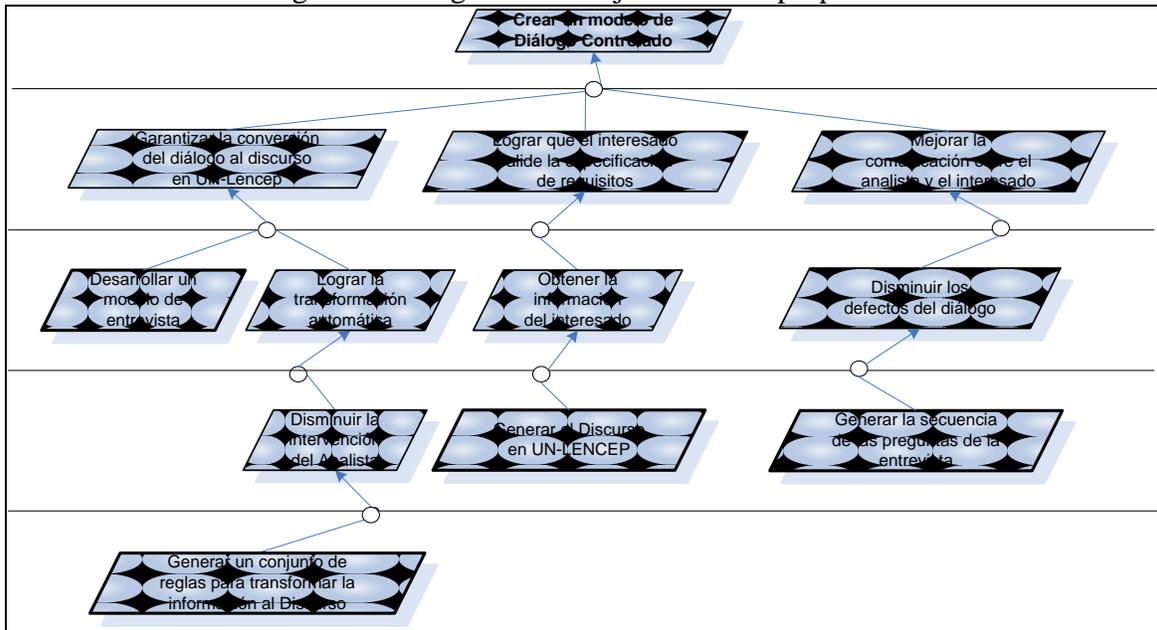
4.2.2 Objetivos Específicos

- Delimitar las características del diálogo para el proceso de educación de requisitos.
- Definir un conjunto de reglas que permitan identificar los elementos de UN-Lencep desde ese diálogo controlado.
- Incorporar el método así definido en el prototipo de una herramienta computacional que lo automatice.
- Ejemplificar el método propuesto y el prototipo construido con un caso de estudio.



Para cumplir con estos objetivos y de acuerdo a las limitaciones de la literatura especializada, se proyectan algunos objetivos deseables para la solución propuesta. La figura 18 representa estas características.

Figura 18. Diagrama de Objetivos de la propuesta





V. PROPUESTA DE SOLUCION

De acuerdo con los planteamientos del capítulo anterior, en esta Tesis se pretende definir un modelo para convertir, de forma automática, un diálogo controlado de educación de requisitos, en un discurso en UN–Lencep para la especificación de requisitos de software, de forma que se elimine la restricción del dominio del conocimiento y, además, para que la información se presente en un formato que el interesado pueda comprender y validar.

La propuesta de esta Tesis se basa en el uso del lenguaje controlado UN–Lencep, para lo cual se identifican los elementos que lo componen, que se presentaron en la tabla 1. Adicionalmente, se establece un conjunto de preguntas que permiten obtener del interesado estos elementos, empleando el diálogo. Finalmente, se define un conjunto de reglas para convertir las respuestas del interesado en las frases del discurso. Todas estas características se incluyen en un prototipo funcional para la automatización del proceso.

5.1 Método Propuesto

5.1.1 Modelo de Entrevista

En el modelo propuesto, se establece un compendio de preguntas que facilitan la obtención de los diferentes elementos del dominio del problema, para la construcción del discurso del interesado. Estas preguntas, se definieron tomando en consideración el análisis hecho, en el numeral 2.3.1, del lenguaje controlado UN-Lencep. De tal forma, en este modelo se utiliza una entrevista semiestructurada, según los planteamientos de Clark-Carter (2002). En la tabla 4 se listan las preguntas establecidas en el modelo, el orden en que se realizan y las preguntas que varían el flujo de la entrevista.



Tabla 4. Preguntas del Modelo y su secuencia

Pregunta	Precondición
P1 ¿Cuál es el nombre de la organización?	Inicio
P2 ¿Cuál es el área o subdivisión de la organización donde se presentan los problemas?	P1
P3 Elabore una lista de los roles o actores que intervienen en el desarrollo de los procesos del área.	P2
P4 ¿Considera que los roles o actores mencionados anteriormente poseen características similares que permitan agruparlos por conjuntos o categorías?	P3
P5 ¿Qué actores podemos categorizar y qué nombre se le puede dar a la categoría?	P4 = ‘SI’
P6 ¿Qué características tiene cada categorías y por ende, todos los actores que se agrupan en ella?	P5
P7 Si algunas de las características mencionadas tienen sus propias subcaracterísticas en particular, lístelas.	P6
P8 ¿Existen ejemplos de las características mencionadas anteriormente dentro de un rango específico?	P7
P9 Liste los valores, ejemplos o instancias de cada característica.	P8 = ‘SI’
P10 ¿Qué actores o roles tienen características en particular y cuáles son éstas?	P4 = ‘NO’ V P8 = ‘NO’ V P9
P11 Liste las funciones, acciones o tareas que realiza cada uno de los actores y los objetos involucrados.	P10
P12 ¿De los objetos mencionados anteriormente, existen algunos que se pueden agrupar por conjuntos o categorías?	P11
P13 ¿Cuáles objetos se pueden agrupar y en cuál categoría?	P12 = ‘SI’
P14 ¿Cuáles objetos tienen características en particular?	P12 = ‘NO’ V P13
P8 ¿Existen ejemplos... dentro de un rango específico?	P14
P9 Liste los valores... de cada característica.	P8 = ‘SI’
P15 Liste la secuencia en la cual se realizan las funciones durante el proceso.	P8 = ‘NO’ V P9
P16 ¿Alguna de las funciones de los diferentes actores debe cumplir con una condición o restricción para que se pueda llevar a cabo?	P15
P17 ¿Para cumplir los objetivos de la organización y los requisitos concernientes de dichos procesos que funciones se deben tener en cuenta?	P16



El conjunto de preguntas y el orden a realizarlas le permite, a quien aplique el modelo, realizar algunas tareas del analista. Sin embargo, se recomienda el acompañamiento de éste durante el desarrollo proceso. Para el conjunto de preguntas definidas en el modelo, se tomaron como base las establecidas en el juego de educación de requisitos de Zapata y Giraldo (2009). No obstante, cuando surgen nuevos elementos en el discurso, se requiere repetir algunas preguntas para obtener la información completa. De este modo, desde la P11 hasta la P14 ingresan nuevos elementos al discurso, los cuales pueden presentar la información de la P9. Por ello, es necesario repetir la P8 y la P9 de acuerdo a las condiciones establecidas en la tabla 4.

5.1.2 Reglas de Transformación del diálogo

Con la información de la tabla 4 el interesado, preferiblemente en compañía del analista, adelanta el proceso de educación de requisitos. Posteriormente, mediante el uso de las reglas de traducción que se presentan a continuación, se convierten las respuestas del interesado en el discurso de especificación de requisitos expresado en UN-Lencep. El valor de los subíndices de las reglas, se toma del orden en que el usuario da las respuestas.

5.1.2.1 Regla para obtención de Roles como Conceptos (R1).

La *R1* se utiliza con la *P3* para identificar los roles o actores que se involucran en los procesos de la organización o el área problemática. La respuesta del usuario se convierte en una lista *A* con *n* elementos *C*. Entonces, por cada actor de la lista debe existir un concepto en el discurso. La tabla 5 muestra la regla sus elementos y un ejemplo de su aplicación.



Tabla 5. Regla para la obtención de Roles como Conceptos

Pregunta	Elemento Obtenido	Regla de Transformación
P3	$A_n = Actor_n$	$A \in Concepto$ $\forall A_n \exists! C_m \rightarrow C_m = A_n$
	$A_1 = Secretaria$ $A_2 = Mensajero$ $A_3 = Administrador$ $A_4 = Cliente$	$C_{21} = A_1 = Secretaria$ $C_{22} = A_2 = Mensajero$ $C_{23} = A_3 = Administrador$ $C_{24} = A_4 = Cliente$

5.1.2.2 Regla para la obtención de la Tríada Estructural (TE) ‘es’ (R2).

La R2 se utiliza con la P5 y P13 para construir una TE que emplea la RE ‘es’. De la respuesta del usuario se obtiene una lista de n elementos C que se denomina Concepto Origen (Co). Entonces, para cada elemento de la lista Co debe existir una categoría, la cual se denomina Concepto Destino (Cd), la unión de estos se asocia a una TE. La tabla 6 muestra la regla sus elementos y un ejemplo de su aplicación.

Tabla 6. Regla para la obtención de la Tríada Estructural ‘es’

Pregunta	Elemento Obtenido	Regla de Transformación
P5, P13	$Co_i = Subcategoría_i$ $Cd_j = Categoría_j$ $RE = "es"$	$Co, Cd \in Concepto$ $\forall (Co_i, Cd_j) \exists! TE_{(i,j)} \rightarrow$ $TE_{(i,j)} = Co_i + RE + Cd_j$
	$Co_{21} = Secretaria$ $Co_{22} = Mensajero$ $Cd_{10} = Empleado$	$TE_{(21,10)} = "Secretaria es Empleado"$ $TE_{(22,10)} = "Mensajero es Empleado"$



Continuación Tabla 6. Regla para la obtención de la Tríada Estructural ‘es’

Pregunta	Elemento Obtenido	Regla de Transformación
	Co_{31} = Casa Co_{32} = Local Co_{33} = Lote Cd_{21} = Inmueble	$TE_{(31,21)}$ = “Casa es Inmueble” $TE_{(32,21)}$ = “Local es Inmueble” $TE_{(33,21)}$ = “Lote es Inmueble”

5.1.2.3 Regla para la obtención de la Tríada Estructural ‘tiene’ (R3).

La R3 se utiliza con la P6, P7, P10 y P14 para construir una TE que emplea la RE ‘tiene’. De la respuesta del usuario se obtiene una lista de n elementos C llamados Cd , los cuales son las subcaracterísticas. Cada elemento de la lista Cd se asocia con una característica, llamada Co . Con la unión de estos elementos se genera la TE. La tabla 7 muestra la regla sus elementos y un ejemplo de su aplicación.

Tabla 7. Regla para la obtención de la Tríada Estructural ‘tiene’

Pregunta	Elemento Obtenido	Regla de Transformación
	Co_i = Característica _{i} Cd_n = Subcaracterística _{n} RE = “tiene”	$Co, Cd \in Concepto$ $\forall (Co_i, Cd_n) \exists TE_{(i,n)} \rightarrow$ $TE_{(i,n)} = Co_i + RE + Cd_n$
P6, P7, P10, P14	Co_{10} = Empleado Cd_{33} = Nombre Cd_{34} = Identificación	$TE_{(10,33)}$ = “Empleado tiene Nombre” $TE_{(10,34)}$ = “Empleado tiene Identificación”
	Co_{34} = Identificación Cd_{48} = Tipo Cd_{49} = Número	$TE_{(34,48)}$ = “Identificación tiene Tipo” $TE_{(34,49)}$ = “Identificación tiene Número”



5.1.2.4 Regla para la obtención del Posible Valor (R4).

La R4 se utiliza con la P9 para construir un PV, la cual contiene un texto fijo en su estructura, que se define en Tx. La respuesta del usuario es una lista de n valores llamada Vr, los cuales son características fijas de un elemento C. A cada elemento de la lista Vr se le asocia un elemento C. Esta unión permite la aparición del PV. La tabla 8 muestra la regla sus elementos y un ejemplo de su aplicación.

Tabla 8. Regla para la obtención del Posible Valor

Pregunta	Elemento Obtenido	Regla de Transformación
P9	$C_i = \text{Concepto}_i$ $Vr_n = \text{Ejemplo}_n$ Tx = “es un posible valor de”	$\forall (Vr_i, C_j) \exists ! PV_{(i,j)} \rightarrow$ $PV_{(i,j)} = Vr_i + Tx + C_j$
	$C_{48} = \text{Tipo}$ $Vr_{19} = \text{NIT}$ $Vr_{20} = \text{NIE}$ $Vr_{21} = \text{CC}$ $Vr_{22} = \text{TI}$	$PV_{(19,48)} = \text{“NIT es un posible valor de Tipo”}$ $PV_{(20,48)} = \text{“NIE es un posible valor de Tipo”}$ $PV_{(21,48)} = \text{“CC es un posible valor de Tipo”}$ $PV_{(22,48)} = \text{“TI es un posible valor de Tipo”}$

5.1.2.5 Regla para la obtención de la Tríada Dinámica (TD) (R5).

La R5 se utiliza con la P11 para obtener una TD. Esta emplea dos C, uno de la lista actores llamado Co y otro de la lista de características o subcaracterísticas llamado Cd. Las acciones o procesos que realiza el actor es una lista de RD. A cada elemento de la lista Cd se le asocia una acción RD. Un actor de la lista Co debe estar vinculado con mínimo una acción de la lista RD. Cada conjunto formado como



respuesta a la *P11*, da lugar a una *TD*. La tabla 9 muestra la regla sus elementos y un ejemplo de su aplicación.

Tabla 9. Regla para la obtención de la Tríada Dinámica

Pregunta	Elemento Obtenido	Regla de Transformación
P11	$Co_i = Actor_i$ $Cd_n = Característica_n$ $RD_x = Verbo_x$	$Co \wedge Cd \in Concepto$ $RD \in relación Dinámica$ $\forall (Co_i, RD_j, Cd_k) \exists ! TD_{(i,j,k)} \rightarrow$ $TD_{(i,j,k)} = Co_i + RD_j + Cd_k$
	$Co_{23} = Administrador$ $Co_{24} = Cliente$ $Cd_{61} = Factura$ $Cd_{62} = Pago$ $RD_{14} = Elabora$ $RD_{16} = Realiza$	$TD_{(23,14,61)} = \text{“Administrador Elabora Factura”}$ $TD_{(24,16,61)} = \text{“Cliente Realiza Pago”}$

5.1.2.6 Regla para la obtención de la Implicación (R6).

La *R6* se utiliza con la *P14* para obtener una *I*, la cual emplea dos *TD* que indican la secuencia en que se realizan los procesos o acciones de la organización o el área problemática. La tabla 10 muestra la regla sus elementos y un ejemplo de su uso.

Tabla 10. Regla para la obtención de la Implicación

Pregunta	Elemento Obtenido	Regla de Transformación
P15	TD_1, \dots, TD_n	$I \in implicación$ $\forall (TD_n, TD_{n+1}) \exists ! I_m \rightarrow$ $I_{(n,n+1)} = \text{“Cuando”} + TD_n + \text{“,”} + TD_{n+1}$



Continuación Tabla 10. Regla para la obtención de la Implicación

Pregunta	Elemento Obtenido	Regla de Transformación
P15	TD_{33} = “Administrador Elabora Factura” TD_{34} = “Cliente Realiza Pago”	$I_{(33,34)}$ = “Cuando Administrador Elabora Factura, Cliente Realiza Pago”

5.1.2.7 Regla para la obtención del Condicional (R7).

La R7 se utiliza con la P15 para obtener una CN, la cual emplea una TD y una instrucción llamada Cond (en ámbito de la regla), la cual indica la condición o restricción que se debe cumplir llevar a cabo la TD. Esta instrucción, generalmente, se expresa en términos de lógica matemática. La tabla 11 muestra la regla sus elementos y un ejemplo de su uso.

Tabla 11. Regla para la obtención del Condicional

Pregunta	Elemento Obtenido	Regla de Transformación
P16	TD_n = Triada Dinámica $Cond_m$ = Instrucción	$CN \in Condicional$ $\forall (Cond_m, TD_n) \exists ! CN_{(m,n)} \Rightarrow$ $CN_{(m,n)} = "Si" + Cond_m + "entonces" + TD_n$
	TD_{28} = “Secretaria Registra Pago” $Cond_{13}$ = Valor de Pago \geq Valor Pagado	$CN_{(13,28)}$ = “Si Valor de Pago \geq Valor Pagado entonces Secretaria Registra Pago”



5.1.2.8 Regla para la obtención del Conjunto de Logro (R8).

La R8 se utiliza con la P16 para obtener un CL, éste emplea una RL y un elemento E, el cual puede ser un C, TE o un TD. Los valores de RL se limitan al conjunto establecido en la tabla 2. El CL sirve para representar los objetivos de la organización. La tabla 12 muestra la regla sus elementos y un ejemplo de su uso.

Tabla 12. Regla para la obtención del Conjunto de Logro

Pregunta	Elemento Obtenido	Regla de Transformación
P17	$RL_i = \text{Relación de Logro}$ $E_n = \text{Elemento}$	$CL \in \text{Conjunto de Logro}$ $\forall E_m \exists! (C_i \vee TE_j \vee TD_k) \Rightarrow$ $\forall (RL_n, E_m) \exists! CL_{(n,m)} \Rightarrow$ $CL_{(n,m)} = RL_n + E_m$
	$RL_{24} = \text{“Garantizar”}$ $E_{47} = TD = \text{“Mensajero Entrega Recibo”}$	$CL_{(24,47)} = \text{“Garantizar Mensajero Entrega Recibo”}$

5.1.3 Integración del Modelo

Los elementos descritos en los numerales anteriores, se integran en un prototipo funcional que automatiza el proceso. De este modo, se disminuye la intervención del analista y se reducen los posibles errores humanos en la transformación de la información. Además, se mejora el tiempo de desarrollo del proceso. Otro aspecto a tener en cuenta es que, mediante el uso de las diferentes herramientas basadas en UN-Lencep, se pueden obtener algunos esquemas conceptuales de UML (Zapata & Arango, 2007). Así, se logra consistencia entre la especificación, el diseño y las necesidades del interesado.



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

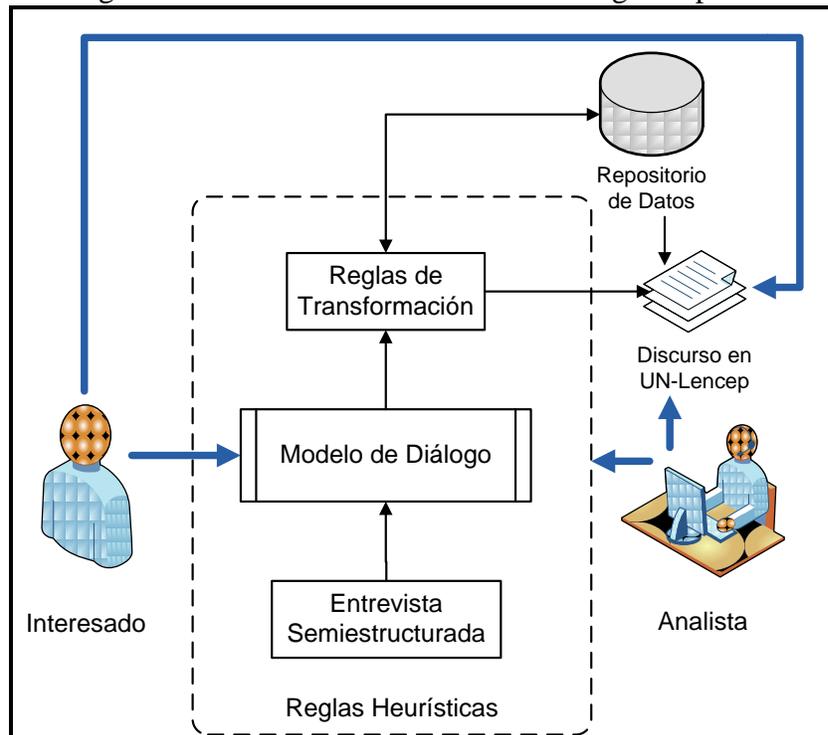
Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

Wiliam A. Arévalo Camacho

La integración de la propuesta, se detalla en la figura 19, en la cual se presenta la interacción de los actores con el modelo y de los componentes entre sí. Asimismo, se incluyen elementos externos necesarios para el cumplimiento del objetivo propuesto. De este modo, el interesado interactúa directamente con el modelo de diálogo, mientras que el analista lo hace con todos los elementos que lo componen. El modelo de diálogo selecciona las preguntas y las presenta al usuario, del cual recibe las respuestas. Posteriormente, se aplican las reglas de transformación y genera se obtienen los diferentes elementos del UN-Lencep, las cuales se guardan en un repositorio de datos. Al finalizar la entrevista, se genera el discurso para que el interesado lo valide.

Figura 19. Estructura del Modelo de Diálogo Propuesto

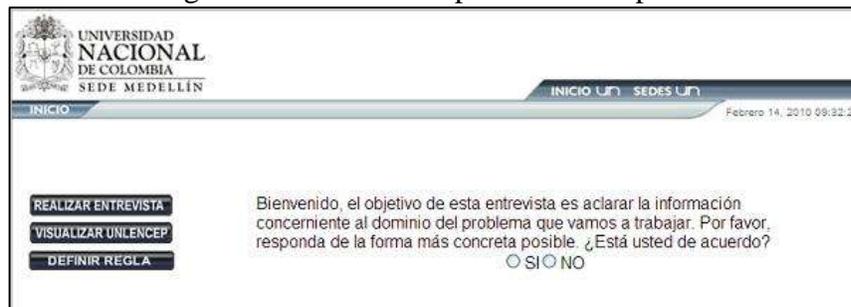




5.1.4 Prototipo Funcional

La integración descrita en el numeral anterior se implementa en un prototipo funcional, el cual permite realizar la entrevista, generar el discurso y definir las reglas. El interesado puede utilizar, directamente, este prototipo. Sin embargo, se recomienda la compañía del analista, para un mejor resultado del proceso. En el prototipo, la presentación de las preguntas incluye un ejemplo de la respuesta, lo cual facilita la comprensión del usuario y ayuda a obtener respuestas precisas y acordes con la pregunta realizada (según recomienda Clark-Carter, 2002, para este tipo de elementos). La figura 20 muestra el menú principal del prototipo.

Figura 20. Menú Principal del Prototipo



5.1.4.1 Realizar Entrevista.

Esta opción presenta al usuario las preguntas definidas para el modelo y, de acuerdo con las respuestas, se guía el desarrollo del proceso, conforme con la secuencia que se establece en la tabla 4. Algunas preguntas se presentan con técnicas que facilitan la obtención de la información y mejoran la interacción con el usuario.

Las respuestas se almacenan en una base de datos mediante la aplicación de las reglas definidas en el numeral 5.1.2. En la preguntas abiertas, el usuario digita la



respuesta y, en algunas ocasiones, se deben separar con punto y coma (;), guión (-) o coma (,), pero esto se le indica al usuario en el enunciado. Otras respuestas se pueden seleccionar de acuerdo con la información que se presenta. También, existen preguntas cerradas para definir el flujo de la entrevista.

De otro lado, en algunas preguntas se utiliza la información que suministra el interesado para complementarlas y mejorar la comunicación humano-máquina. Además, si la información que se suministra se encuentra en el repositorio de datos, ésta no se repite, sino que se vincula con la entrevista actual. Esto se realiza de acuerdo con las buenas prácticas de desarrollo de software.

La figura 21 muestra la interfaz de inicio de la entrevista y permite registrar el proyecto. En ésta se realizan las preguntas P1 y P2 de la tabla 4. Para continuar el proceso, se dispuso la opción “Registrar Proyecto”, mientras que la opción “Restablecer” borra las repuestas de las preguntas que se presentan.

Figura 21. Interfaz de las preguntas P1 y P2 de la entrevista

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN

INICIO UN SEDES UN

Febrero 14, 2010 09:32:52

¿Cual es el nombre de la organización?

¿Cual es el área o subdivisión de la organización donde se presentan problemas?

¿Cual es el nombre que le quiere dar al proyecto?

Registrar Proyecto Restablecer

Luego de registrar el proyecto, se presenta la pregunta P3. En esta interfaz se ejemplifica la respuesta, con el propósito que el usuario tenga claridad sobre cómo ingresar la respuesta. La figura 22 presenta esta opción.



Figura 22. Interfaz de la pregunta P3 de la entrevista

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN

INICIO UN SEDES UN

INICIO Febrero 14, 2010 09:32:19

Elabore una lista de las **personas o actores** que intervienen en el desarrollo de los procesos del área en cuestión
(Separados por ", " por ";" o por "." y en singular).

(Ejemplo: "Secretaria, mensajero, asistente, gerente")

Registrar Actores Restablecer

Una vez registrados los actores, se presenta la pregunta P4, la cual es de tipo cerrado, por lo cual, una vez el usuario selecciona la respuesta, el software presenta la opción que corresponda de acuerdo con lo que se establece en la tabla 4. En la figura 23 muestra esta interfaz.

Figura 23. Interfaz de la pregunta P4 de la entrevista

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN

INICIO UN SEDES UN

INICIO Febrero 14, 2010 09:32:01

¿Considera usted que los actores mencionados anteriormente poseen características similares que permitan agruparlos por **conjuntos o categorías**?

(EJEMPLO: Para calcular una nomina, los actores 'Administrador', 'Vendedor', 'Bodeguero', 'Cajero', 'Aseador', etc. se pueden agrupar en la categoría "Empleado" ya que comparten las características que nos interesan para tal fin, como 'Nombre', 'Apellido', 'Cedula', 'Salario', etc. De ese modo, un Administrador es un Empleado, un Vendedor es un Empleado, un Bodeguero es un Empleado, etc.).

SI NO

Cuando la opción es “Si” se presenta la pregunta P5, la cual se puede apreciar en la figura 24. Cuando la opción es “No” se presenta la pregunta P10, la cual se puede apreciar en la figura 29.



Figura 24. Interfaz de la pregunta P5 de la entrevista

Después de categorizar los actores se realiza la pregunta P6. En esta opción se presentan las categorías incluidas en la pregunta anterior. De este modo, se centra al entrevistado respecto de las respuestas que suministra. La figura 25 muestra el ejemplo de esta interfaz. Asimismo, se presenta la pregunta P7, la cual también utiliza las respuestas de la pregunta anterior para solicitar la información. Esta interfaz se puede apreciar en la figura 26.

Figura 25. Interfaz de la pregunta P6 de la entrevista



Figura 26. Interfaz de la pregunta P7 de la entrevista

La figura 27 muestra la pregunta P8, la cual tiene una función similar a la P4. De este modo, cuando la respuesta del usuario es “Si”, se presenta la pregunta P9, que se puede ver en la figura 28, pero, cuando la respuesta es “No” se presenta la pregunta P10, que se detalla en la figura 29.

Figura 27. Interfaz de la pregunta P8 de la entrevista

Figura 28. Interfaz de la pregunta P9 de la entrevista



Figura 29. Interfaz de la pregunta P10 de la entrevista

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLIN

INICIO UN SEDES UN

Febrero 14, 2010 09:34:25

¿Cuáles actores tienen características en particular?
Por favor escríbalas separadas por ",", por "-," o por "-".

Si no hay características que deban ser tomadas en cuenta, deje el espacio de la respuesta vacío. Si hay categorías, no tenga en cuenta las características que ya fueron ingresadas en la categoría a la que pertenece el actor por el que se está preguntando.

(EJEMPLO: "¿Cuales son las características de Empleado? Nombre, carnet, sueldo, cargo")

¿Cuáles son las características de ADMINISTRADOR?

¿Cuáles son las características de CONTADOR?

Al registrar las características de cada actor, se inicia la construcción de las tríadas dinámicas, para lo cual se presenta la pregunta P11 que se ilustra en la figura 30. Sin embargo, con las respuestas de esta pregunta pueden surgir nuevos elementos del dominio, entonces se realiza la pregunta P12, la cual se aprecia en la figura 31.

Figura 30. Interfaz de la pregunta P11 de la entrevista

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLIN

INICIO UN SEDES UN

Febrero 14, 2010 09:35:00

Ahora vamos a continuar con **las funciones** de los actores. Nos interesa saber las acciones o tareas, en forma de tríadas, que realiza un actor sobre un objeto.

(EJEMPLO: En la nomina, "Contador calcula nomina", "Gerente autoriza nomina", "administrador recibe Salario", "Vendedor recibe Comisión", etc. son funciones de Contador, Gerente, Administrador y Vendedor respectivamente).

¿Que **funciones** realiza el ADMINISTRADOR?

ADMINISTRADOR

¿Que **funciones** realiza el SECRETARIA?

SECRETARIA

Figura 31. Interfaz de la pregunta P12 de la entrevista

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLIN

INICIO UN SEDES UN

Febrero 14, 2010 09:32:46

¿Considera usted que los objetos mencionados anteriormente se pueden agrupar por **conjuntos o categorías**?

(EJEMPLO: 'Libro' y 'Revista' pueden ser agrupados bajo la categoría 'Documento')

SI NO



La P12 es similar a la P4, con la diferencia que en la P4 se asocian actores y en la P12 se asocian objetos. Entonces, si la respuesta a la P12 es “Si” se realizan las pregunta concernientes a ello y que son similares a las interfaces que se presentan en las figuras 24, 25, 26 y 27. De lo contrario, se realiza la pregunta P15 como se muestra en la figura 32.

Figura 32. Interfaz de la pregunta P15 de la entrevista

UNIVERSIDAD NACIONAL DE COLOMBIA SEDE MEDELLÍN

INICIO UN SEDES UN

Febrero 14, 2010 09:32:35

De acuerdo, ahora tratemos de establecer una secuencia de las funciones de los diversos actores para saber si una función **implica** la realización de otra dentro de un proceso, es decir, nos interesa saber si una función solamente se realiza luego de que ya se ha empezado a realizar una o se está realizando otra desde antes.

(EJEMPLO: en la nomina, "CUANDO Contador calcula Nomina, Gerente autoriza Nomina", y luego "CUANDO Gerente autoriza Nomina, Administrador recibe Salario").

Acciones que hay en el sistema		Secuencia ordenada de implicaciones
CONTADOR REGISTRA CONTABILIDA	>>	Agregar a la Secuencia de Implicaciones
CONTADOR VALIDA PAGO		
REVISOR FISCAL VERIFICA CONTAB		Eliminar de la Secuencia de Implicaciones
REVISOR FISCAL VERIFICA PAGO	<<	

Registrar otra Secuencia de Implicaciones Registrar Secuencia de Implicaciones y continuar No registrar implicaciones

Para responder la pregunta P16 se recomienda poseer conocimientos de lógica, pero la interfaz presenta las opciones necesarias para definir la instrucción, de acuerdo con la estructura del UN-Lencep. En la figura 33 se muestra la interfaz de esta pregunta.



Figura 33. Interfaz de la pregunta P16 de la entrevista

Para concluir la entrevista se realiza la pregunta P17, la cual presenta un conjunto limitado de verbos de acuerdo con lo que establecen Zapata y Lezcano (2009). Además, se presentan las posibles uniones que se puede tener el conjunto de logro, con todos los elementos del discurso, tomando en cuenta las respuestas que suministra el entrevistado. La figura 34 presenta la interfaz con la que se realiza esta pregunta.

Figura 34. Interfaz de la pregunta P17 de la entrevista

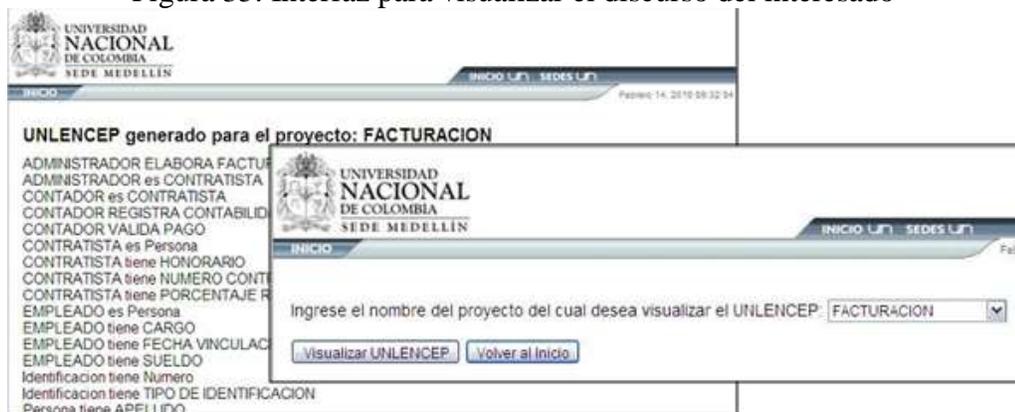


5.1.4.2 Visualizar UN-Lencep.

Esta opción le permite al usuario generar y consultar un discurso luego de realizar la entrevista. El discurso se genera en asocio con las reglas definidas en el numeral 5.1.2, pero el texto explícito del discurso no se guarda en la base de datos. Por lo tanto, se establece una opción para exportar el discurso, de forma que permita su uso con otras herramientas que contribuyen en el proceso de especificación y diseño.

Esta utilidad permite consultar los discursos que se generen en el modelo. Así, el usuario puede lograr una mejor comprensión de la herramienta e, incluso, apreciar la utilidad del discurso. Asimismo, se pueden validar los contenidos del mismo o interiorizar los resultados de forma que se comprendan completamente e identifiquen todos los elementos del dominio. Aún más, se puede detectar una incompletitud en el discurso, de forma que estos errores se detecten en una etapa temprana, de acuerdo con las expectativas de esta Tesis. La figura 35 muestra las interfaces de la opción Visualizar UN-Lencep.

Figura 35. Interfaz para visualizar el discurso del interesado





5.1.4.3 Definir Regla.

Esta opción permite incluir nuevas reglas, como una alternativa para la mejora del modelo. Así, se presentan las tablas de la base de datos, para que se defina que atributos de esta involucra la regla. También, se presenta la lista de las interfaces del prototipo para, de este modo, establecer en cuál de ellas se ejecutara dicha regla. Además, se definen 2 tipos de regla, de inserción y de lectura. La primera se conforma con las instrucciones para guardar los datos de la interfaz en el repositorio de datos. La segunda constituye la forma como se recuperarán los datos del repositorio y permite crear las frases del discurso.

Estas opciones hacen más interactivo y dinámico el modelo. Sin embargo, esta utilidad requiere de un amplio conocimiento técnico de la estructura del prototipo, para realizar modificaciones o inclusiones en las reglas establecidas. De otro lado, cualquier modificación que se haga en las reglas afectará, directamente, el discurso que se obtiene con el modelo de diálogo. La figura 36 presenta la interfaz de esta opción, para una mayor comprensión del contenido de la misma.

Figura 36. Interfaz para definir regla



VI. VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

Con el propósito de demostrar la funcionalidad y el aporte metodológico del modelo propuesto, en este capítulo se presentan los casos de estudio. En este sentido, se presentan las respuestas a cada una de las preguntas, el discurso en UN–Lencep generado y el esquema preconceptual obtenido con el discurso.

6.1 Caso de estudio: Facturación en Propiedad Horizontal

Realizar la facturación y llevar el estado de cuenta de cada propietario es una necesidad sentida en el Conjunto Residencial Senderos de San Jacinto. Con el propósito de ayudar en el manejo de esta información, se realiza la entrevista al administrador, utilizando el modelo propuesto en esta Tesis. En el desarrollo de este proceso, que para este caso de estudio se llama “Facturación PH”, se obtiene la información que se presenta en la tabla 13.

Tabla 13. Respuestas de la entrevista con el Administrador de San Jacinto

Pregunta	Respuesta
P1	C.R. Senderos de San Jacinto
P2	Facturación
P3	Secretaría, Administrador, Contador, Propietario
P4	Si
P5	Secretaria, Administrador, Contador, se agrupan en: Empleado Propietario, Empleado, se agrupan en: Persona
P6	Persona: Nombre, Identificación, Teléfono, Dirección, Sexo, E-mail Empleado: Cargo, Sueldo, Tipo de Contrato
P7	No
P8	Si
P9	Sexo: Masculino, Femenino Tipo de Contrato: Honorarios, Servicios, Indefinido
P10	Propietario: Inmueble
P7	Identificación: Tipo, Numero de Identificación
P8	Si
P9	Tipo: Cedula de Ciudadanía, Cedula de Extranjería, Pasaporte, NIT



Continuación Tabla 13. Respuestas de la entrevista con el Administrador de San Jacinto

Pregunta	Respuesta
P11	El Administrador elabora la Factura, el Administrador envía la Factura, el Propietario recibe la Factura, el Propietario paga la Factura, el Propietario envía el Soporte de Pago, la Secretaria recibe el Soporte de Pago, la Secretaria elabora el Recibo de Pago, la Secretaria envía el Recibo de Pago, el Propietario recibe el Recibo de Pago, el Contador verifica el Recibo de Pago, el Contador concilia la Cuenta Bancaria
P12	No
P14	Cuenta Bancaria: Número de Cuenta, Banco, Mes del Movimiento, Movimiento Factura: Fecha de Elaboración, Propietario, Valor de la Cuota del Mes, Saldo Vencido, Intereses, Valor Total, Fecha de Vencimiento, Número de Factura Recibo de Pago: Fecha del Recibo, Número del Recibo, Factura, Valor Pagado, Fecha de Pago, Soporte de Pago Soporte de Pago: Movimiento Movimiento: Fecha, Sucursal, Valor, Número de Inmueble
P15	Primero el Administrador elabora la Factura Luego el Administrador envía la Factura Primero el Propietario recibe la Factura Luego el Propietario paga la Factura Primero el Propietario paga la Factura Luego el Propietario envía el Soporte de Pago Primero el Propietario envía el Soporte de Pago Luego la Secretaria recibe el Soporte de Pago Primero la Secretaria recibe el Soporte de Pago Luego la Secretaria elabora el Recibo de Pago Primero la Secretaria elabora el Recibo de Pago Luego la Secretaria envía el Recibo de Pago Primero la Secretaria elabora el Recibo de Pago Luego el Contador verifica el Recibo de Pago Primero el Contador verifica el Recibo de Pago Luego el Contador concilia la Cuenta del Banco
P16	Secretaria elabora Recibo de Pago si Valor Pagado = Valor
P17	Se debe Producir la Factura, Alcanzar que la Factura tiene Fecha de Vencimiento, Promover que la Factura tiene Propietario, Garantizar que el Administrador envíe la Factura, Fomentar que el Propietario reciba la Factura, Incrementar que el Propietario pague la Factura, Promover que el Propietario envíe el Soporte de Pago, Garantizar que la Secretaria elabora el Recibo de Pago, Lograr que Recibo de Pago tiene Valor Pagado, Hacer que la Secretaria envíe el Recibo de Pago, Fomentar que el Contador verifica el Recibo de Pago



6.1.1 Discurso en UN–Lencep del Proyecto Facturación PH

A continuación se encuentra el discurso en UN–Lencep, que se generó con las respuestas del entrevistado, mediante el uso del método propuesto, el cual permite ver de forma clara y en un lenguaje técnico la información suministrada. Usando este discurso se puede generar el esquema preconceptual, también algunos esquemas conceptuales de UML.

Administrador Elabora Factura

Administrador Envía Factura

Administrador es Empleado

Contador Concilia Cuenta Bancaria

Contador es Empleado

Contador Verifica Recibo de Pago

Cuenta Bancaria tiene Banco

Cuenta Bancaria tiene Mes del Movimiento

Cuenta Bancaria tiene Movimiento

Cuenta Bancaria tiene Número de Cuenta

Empleado es Persona

Empleado tiene Cargo

Empleado tiene Sueldo

Empleado tiene Tipo de Contrato

Factura tiene Fecha de Elaboración

Factura tiene Fecha de Vencimiento

Factura tiene Intereses

Factura tiene Número de Factura

Factura tiene Propietario

Factura tiene Saldo Vencido

Factura tiene Valor de la Cuota del Mes



Factura tiene Valor Total
Identificación tiene Número de Identificación
Identificación tiene Tipo
Movimiento tiene Fecha
Movimiento tiene Número de Inmueble
Movimiento tiene Sucursal
Movimiento tiene Valor
Persona tiene Dirección
Persona tiene Email
Persona tiene Identificación
Persona tiene Nombre
Persona tiene Sexo
Persona tiene Teléfono
Propietario Envía Soporte de Pago
Propietario es Persona
Propietario Paga Factura
Propietario Recibe Factura
Propietario Recibe Recibo de Pago
Propietario tiene Inmueble
Recibo de Pago tiene Factura
Recibo de Pago tiene Fecha de Pago
Recibo de Pago tiene Fecha del Recibo
Recibo de Pago tiene Numero del Recibo
Recibo de Pago tiene Valor Pagado
Recibo de Pago tiene Soporte de Pago
Secretaria Elabora Recibo de Pago
Secretaria Envía Recibo de Pago
Secretaria es Empleado
Secretaria Recibe Soporte de Pago



Soporte de Pago tiene Movimiento

Femenino es un posible valor de Sexo

Masculino es un posible valor de Sexo

Cedula de Ciudadanía es un posible valor de Tipo

Cedula de Extranjería es un posible valor de Tipo

NIT es un posible valor de Tipo

Pasaporte es un posible valor de Tipo

Honorarios es un posible valor de Tipo de Contrato

Indefinido es un posible valor de Tipo de Contrato

Servicios es un posible valor de Tipo de Contrato

Cuando Administrador Elabora Factura, Administrador Envía Factura

Cuando Administrador Envía Factura, Propietario Recibe Factura

Cuando Propietario Recibe Factura, Propietario Paga Factura

Cuando Propietario Paga Factura, Propietario Envía Soporte de Pago

Cuando Propietario Envía Soporte de Pago, Secretaria Recibe Soporte de Pago

Cuando Secretaria Recibe Soporte de Pago, Secretaria Elabora Recibo de Pago

Cuando Secretaria Elabora Recibo de Pago, Secretaria Envía Recibo de Pago

Cuando Secretaria Elabora Recibo de Pago, Contador Verifica Recibo de Pago

Cuando Contador Verifica Recibo de Pago, Contador Concilia Cuenta Bancaria

Si Valor = Valor Pagado Entonces Secretaria Elabora Recibo de Pago

Producir: Factura

Alcanzar: Factura tiene Fecha de Vencimiento

Fomentar: Contador Verifica Recibo de Pago

Fomentar: Propietario Recibe Factura

Garantizar: Administrador Envía Factura

Garantizar: Secretaria Elabora Recibo de Pago

Hacer: Secretaria Envía Recibo de Pago

Incrementar: Propietario Paga Factura

Lograr: Recibo de Pago tiene Valor Pagado



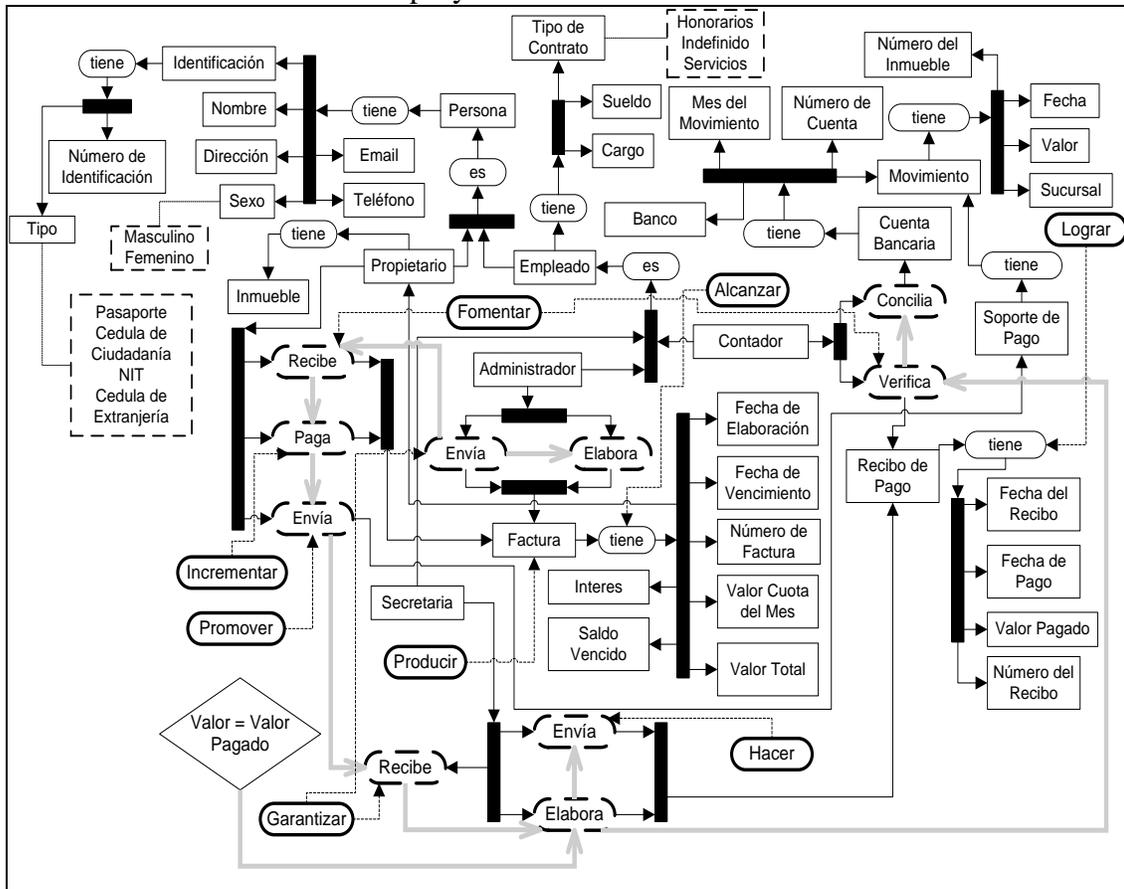
Promover: Factura tiene Propietario

Promover: Propietario Envía Soporte de Pago

6.1.2 Esquema Preconceptual del Proyecto Facturación PH

La figura 37 muestra el esquema preconceptual obtenido con el discurso del numeral anterior, el cual representa gráficamente la información del dominio del problema.

Figura 37. Esquema Preconceptual generado con el discurso en UN-Lencep del proyecto Facturación PH





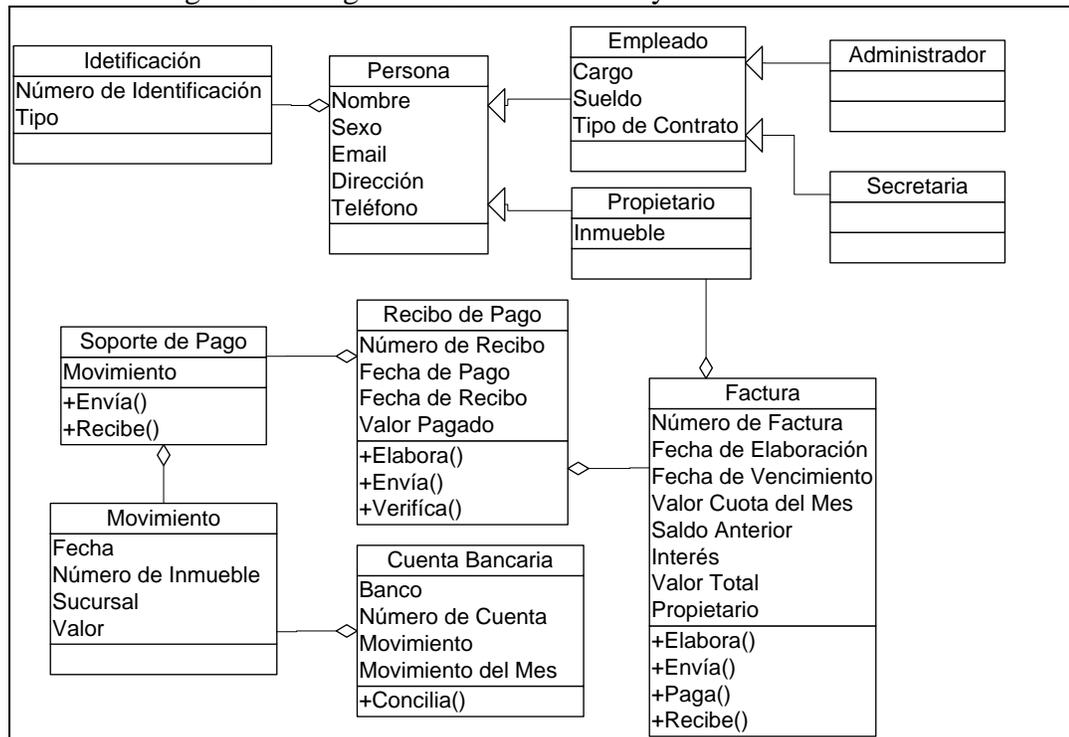
6.1.3 Esquemas Conceptuales de UML del Proyecto Facturación PH

Haciendo uso de las técnicas presentadas por Zapata *et al.* (2007) se obtienen los diagramas de clases, de comunicación, de máquina de estados y de casos de uso. A continuación se presentan estos diagramas, los cuales representan parte de la especificación y el diseño de la solución.

6.1.3.1 Diagrama de Clases.

La figura 38 muestra el diagrama de clases que se obtuvo con el discurso generado por el modelo de diálogo del proyecto de facturación PH.

Figura 38. Diagrama de Clases del Proyecto Facturación PH





6.1.3.2 Diagrama de Comunicación.

La figura 39 muestra el diagrama de Comunicación obtenido con el discurso presentado del proyecto de facturación PH.

Figura 39. Diagrama de Comunicación del Proyecto Facturación PH





6.1.3.3 Diagrama de Máquina de Estados.

La figuras 40, 41 y 42 muestran los diagramas de máquina de estados de los objetos Factura, Recibo de Pago y Soporte de Pago, los cuales se obtuvieron con el discurso generado por el modelo de diálogo para el proyecto de facturación PH.

Figura 40. Diagrama de Máquina de Estados de los objeto Factura

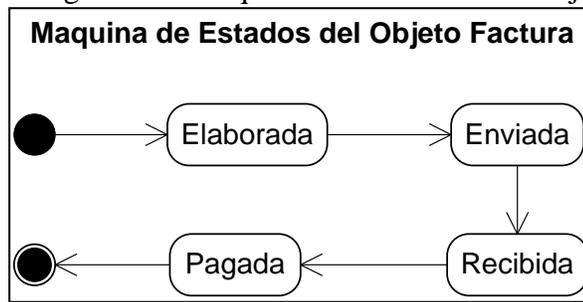


Figura 41. Diagrama de Máquina de Estados de los objeto Recibo de Pago

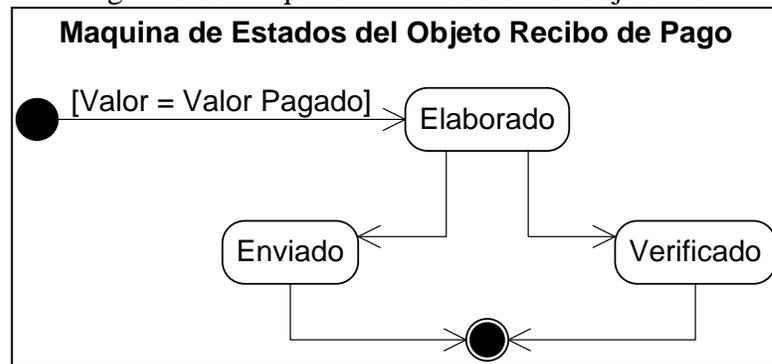
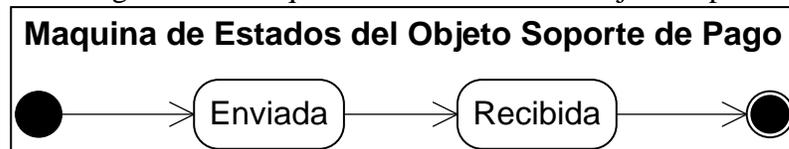


Figura 42. Diagrama de Máquina de Estados de los objeto Soporte de Pago

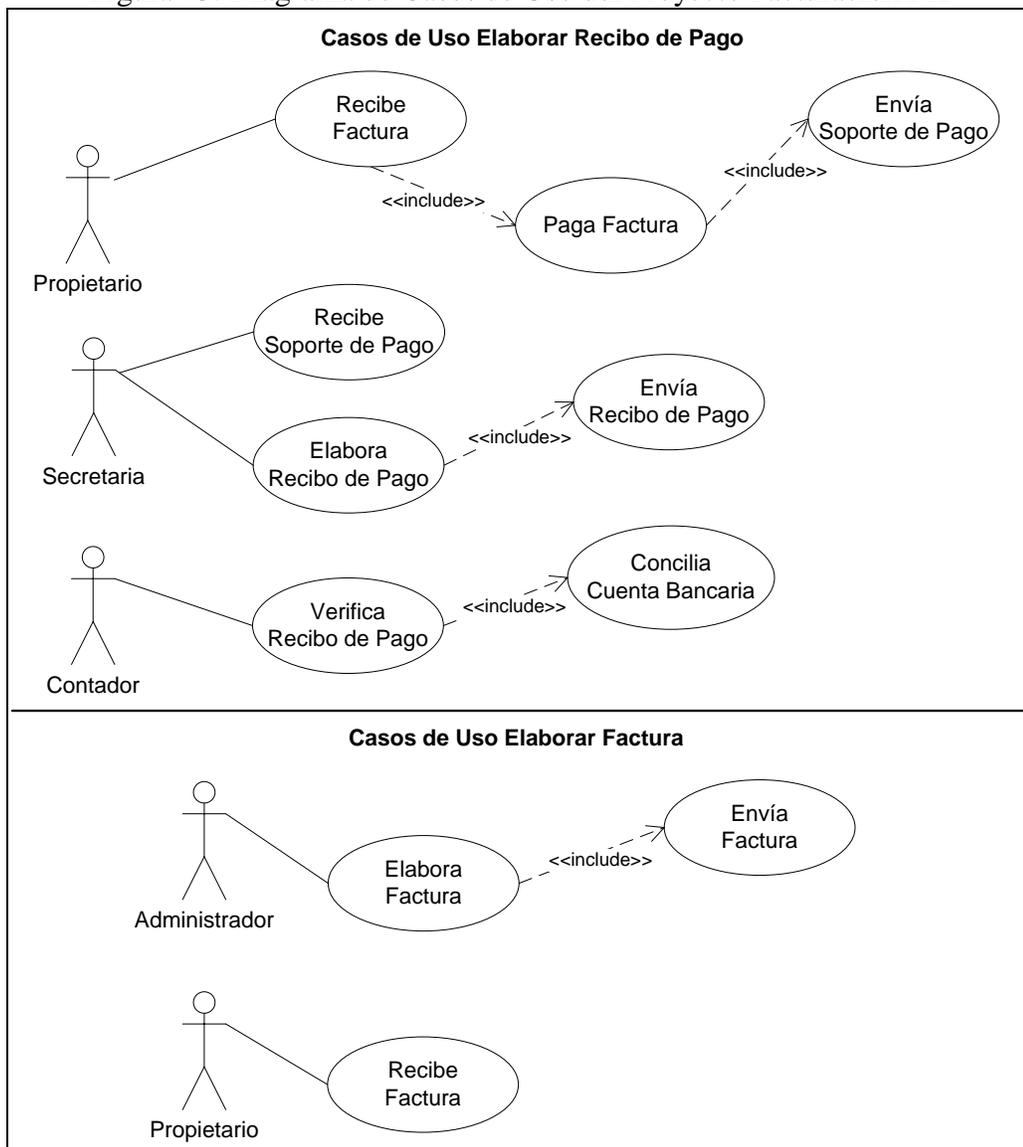




6.1.3.4 Diagrama de Casos de Uso.

La figura 43 muestra los diagramas de casos de uso, los cuales se obtuvieron con el discurso generado para el proyecto de facturación PH, en el cual se incluyen los roles Administrador y Contador, pero en este ejemplo se omitieron, pues no se consideran relevantes, para demostrar la utilidad del modelo y del proceso.

Figura 43. Diagrama de Casos de Uso del Proyecto Facturación PH





6.2 Caso de estudio: Modelo de diálogo para la educación de requisitos basado en UN-Lencep

Con el propósito de realizar la validación de la propuesta y obtener los requisitos del prototipo a desarrollar, que sirvan como base para la documentación del mismo, se utilizó la propuesta de esta Tesis. Así, se amplía la información técnica del prototipo, la cual no se incluyó en el numeral 5.1.4. Para llevar a cabo la entrevista se tomó en cuenta la información de esta investigación, además de la experiencia de diversos expertos en el manejo de lenguaje controlado UN-Lencep. La tabla 14 presenta las respuestas que se obtuvieron en la ejecución del proceso.

Tabla 14. Respuestas de la entrevista para prototipo del Modelo de Diálogo

Pregunta	Respuesta
P1	Grupo de Investigación Lenguajes Computacionales
P2	Educción de Requisitos de Software
P3	Analista, Interesado
P4	No
P10	No es relevante para el proceso a sistematizar
P11	Interesado responde Pregunta, Interesado realiza Entrevista, Analista registra Proyecto, Analista Define regla, Analista aplica regla, Analista construye Frase, Analista visualiza UN-Lencep, Analista realiza Pregunta, Analista registra Condicional, Analista registra Implicación, Analista registra Triada, Analista agrega Posible Valor, Analista registra Concepto, Analista registra Relación de logro, Interesado Valida UN-Lencep
P12	No
P14	Pregunta: Número de pregunta, Tipo de pregunta, Orden de la pregunta, Enunciado, Concepto, Triada Entrevista: Número de entrevista, Etiqueta, Pregunta Proyecto: Nombre del proyecto, Nombre del área, Nombre de la organización, Concepto, Triada, Implicación, Condicional, Relación de logro, Entrevista, UN-Lencep Regla: Número de regla, Tipo de regla, Interfaz, Elemento Frase: Descripción, Tipo de elemento, Código del elemento, Regla UN-Lencep: Frase Condicional: Número de condicional, Instrucción, Elemento condicional Implicación: Número de implicación, Triadaorigen, Triadadestino. Triada: Número de triada, Tipo triada, Conceptorigen, Conceptodestino, Verbo



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

Continuación Tabla 14. Respuestas de la entrevista para prototipo del Modelo de Diálogo

Pregunta	Respuesta
P14	Posible valor: Número del posible valor, Nombre del posible valor Concepto: Número de concepto, Nombre del concepto, Tipo de concepto, Posible valor Relación de logro: Número de relación de logro, Verbo, Elemento de logro Elemento: Número de elemento, Número de regla, Control, Almacén, Orden de regla, Atributo Elemento condicional: Orden, tipo de elemento, Número de elemento, elemento constante, Número de elemento condicional, operador Verbo: Nombre del verbo, Tipo de verbo, Número del Verbo
P8	Si
P9	Tipo de pregunta: Sentencia, Declarativa, Aclarativa, Si/No Tipo de concepto: Actor, Objeto, Categoría Tipo triada: Estructural, Dinámica Tipo de verbo: Característica, Categoría, Dinámica, Logro
P15	Primero Interesado realiza entrevista luego Analista registra Proyecto Primero Analista registra Proyecto luego Analista realiza Pregunta Primero Analista realiza Pregunta luego Interesado responde Pregunta Primero Interesado responde Pregunta luego Analista aplica Regla Primero Analista aplica Regla luego Analista registra Concepto Primero Analista aplica Regla luego Analista registra Triada Primero Analista aplica Regla luego Analista agrega Posible valor Primero Analista aplica Regla luego Analista registra implicación Primero Analista aplica Regla luego Analista registra Condicional Primero Analista aplica Regla luego Analista registra Relación de Logro Primero Analista aplica Regla luego Analista construye Frase Primero Analista construye Frase luego Analista visualiza UN-Lencep Primero Analista visualiza UN-Lencep luego Interesado Valida UN-Lencep
P16	No
P17	Se debe Lograr que Analista Registra Proyecto, Fomentar que Interesado realiza entrevista, Mejorar que Analista realiza Pregunta, Garantizar que Interesado responde Pregunta, Incrementar que Analista aplica Regla, Lograr que Analista registra Concepto, Lograr que Analista registra Triada, Lograr que Analista agrega Posible valor, Lograr que Analista registra implicación, Lograr que Analista registra Condicional, Lograr que Analista registra Relación de Logro, Mejorar que Analista construye Frase, Mejorar que Analista visualiza UN-Lencep, Incrementar que Interesado Valida UN-Lencep



6.2.1 Discurso en UN–Lencep del Proyecto Modelo de Diálogo

A continuación se presenta el discurso en UN–Lencep, que se generó con las respuestas de los entrevistados, mediante el uso de forma manual del método propuesto. De este modo, a las respuestas de la entrevista se le aplicaron las reglas definidas en el numeral 5.1.2. Así, se comprobó que la propuesta era funcional aún sin automatizar.

Proyecto tiene Nombre del proyecto

Proyecto tiene Nombre del área

Proyecto tiene Nombre de la organización

Proyecto tiene Concepto

Proyecto tiene Triada

Proyecto tiene Implicación

Proyecto tiene Condicional

Proyecto tiene Relación de logro

Proyecto tiene Entrevista

Proyecto tiene UN-Lencep

UN-Lencep tiene Frase

Frase tiene Descripción

Frase tiene Tipo de elemento

Frase tiene Código del elemento

Frase tiene Regla

Regla tiene Número de regla

Regla tiene Tipo de regla

Regla tiene Interfaz

Regla tiene Elemento

Elemento tiene Número de elemento

Elemento tiene Número de regla

Elemento tiene Control



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

Elemento tiene Almacén

Elemento tiene Orden de regla

Elemento tiene Atributo

Entrevista tiene Número de entrevista

Entrevista tiene Etiqueta

Entrevista tiene Pregunta

Pregunta tiene Número de pregunta

Pregunta tiene Tipo de pregunta

Pregunta tiene Orden de la pregunta

Pregunta tiene Enunciado

Pregunta tiene Concepto

Pregunta tiene Triada

Concepto tiene Número de concepto

Concepto tiene Nombre del concepto

Concepto tiene Tipo de concepto

Concepto tiene Posible valor

Posible valor tiene Número del posible valor

Posible valor tiene Nombre del posible valor

Condicional tiene Número de condicional

Condicional tiene instrucción

Condicional tiene Elemento condicional

Elemento condicional tiene Orden

Elemento condicional tiene tipo de elemento

Elemento condicional tiene Número de elemento

Elemento condicional tiene elemento constante

Elemento condicional tiene Número de elemento condicional

Elemento condicional tiene operador

Triada tiene Número de triada

Triada tiene Tipo triada



Triada tiene Conceptorigen
Triada tiene Concepto destino
Triada tiene Verbo
Verbo tiene Nombre del verbo
Verbo tiene Tipo de verbo
Verbo tiene Número del Verbo
Implicación tiene Número de implicación
Implicación tiene Triada origen
Implicación tiene Triada destino
Relación de logro tiene Número de relación de logro
Relación de logro tiene Verbo
Relación de logro tiene Elemento de logro
Sentencia es un posible valor de Tipo de pregunta
Declarativa es un posible valor de Tipo de pregunta
Aclarativa es un posible valor de Tipo de pregunta
Si/No es un posible valor de Tipo de pregunta
Actor es un posible valor de Tipo de concepto
Objeto es un posible valor de Tipo de concepto
Categoría es un posible valor de Tipo de concepto
Estructural es un posible valor de Tipo triada
Dinámica es un posible valor de Tipo triada
Característica es un posible valor de Tipo de verbo
Categoría es un posible valor de Tipo de verbo
Dinámica es un posible valor de Tipo de verbo
Logro es un posible valor de Tipo de verbo
Interesado responde Pregunta
Interesado realiza Entrevista
Analista define regla
Analista aplica regla



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

Analista construye Frase

Analista visualiza UN-Lencep

Analista realiza Pregunta

Analista registra Proyecto

Analista registra Concepto

Analista registra Relación de logro

Analista registra Condicional

Analista registra Implicación

Analista registra Triada

Analista agrega Posible Valor

Cuando Interesado realiza entrevista, Analista registra Proyecto

Cuando Analista registra Proyecto, Analista realiza Pregunta

Cuando Analista realiza Pregunta, Interesado responde Pregunta

Cuando Interesado responde Pregunta, Analista aplica Regla

Cuando Analista aplica Regla, Analista registra Concepto

Cuando Analista aplica Regla, Analista registra Triada

Cuando Analista aplica Regla, Analista agrega Posible valor

Cuando Analista aplica Regla, Analista registra implicación

Cuando Analista aplica Regla, Analista registra Condicional

Cuando Analista aplica Regla, Analista registra Relación de Logro

Cuando Analista aplica Regla, Analista construye Frase

Cuando Analista construye Frase, Analista visualiza UN-Lencep

Cuando Analista visualiza UN-Lencep, Interesado Valida UN-Lencep

Fomentar que Interesado realiza entrevista

Lograr que Analista Registra Proyecto

Lograr que Analista registra Concepto

Lograr que Analista registra Triada

Lograr que Analista registra implicación

Lograr que Analista registra Condicional



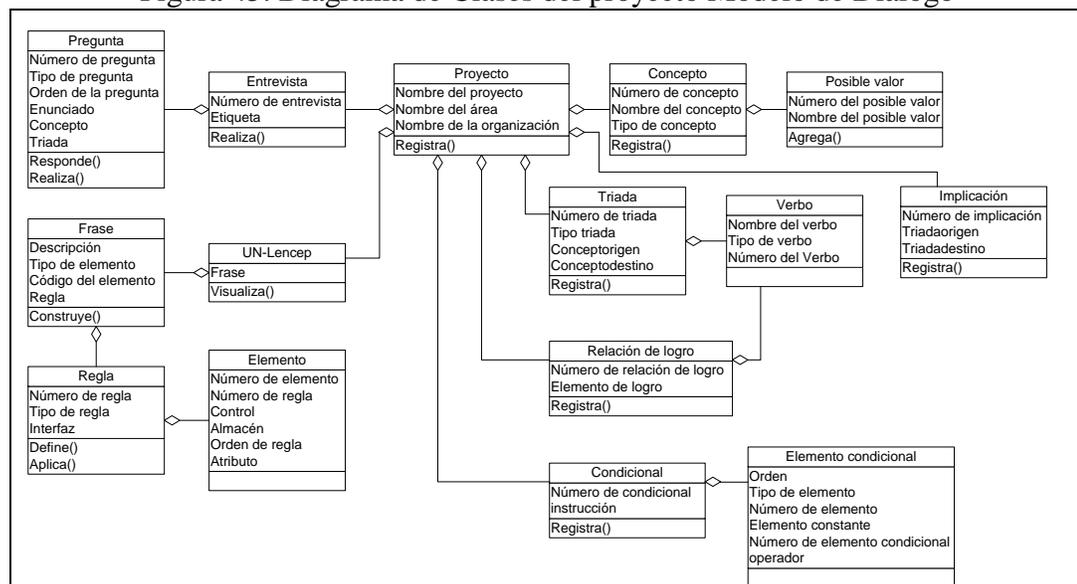
6.2.3 Esquemas Conceptuales de UML del Proyecto Modelo de Diálogo

Para complementar la información técnica de la propuesta, a continuación se presentan diversos esquemas generados haciendo uso de la propuesta de Zapata *et al.* (2007). Así, se pueden observar el diagrama de clases, el diagrama de comunicación, el diagrama de máquina de estados, el diagrama de secuencias y el diagrama de casos de uso, los cuales hacen parte de la especificación y el diseño del prototipo que automatiza el proceso de esta propuesta.

6.2.3.1 Diagrama de Clases.

La figura 45 muestra el diagrama de clases que se obtuvo con el discurso del numeral anterior, el cual permite establecer características funcionales del software y hace parte de la estructura estática del modelo (Fowler & Scott, 1999). Además, es una base para el diseño de la base de datos.

Figura 45. Diagrama de Clases del proyecto Modelo de Diálogo

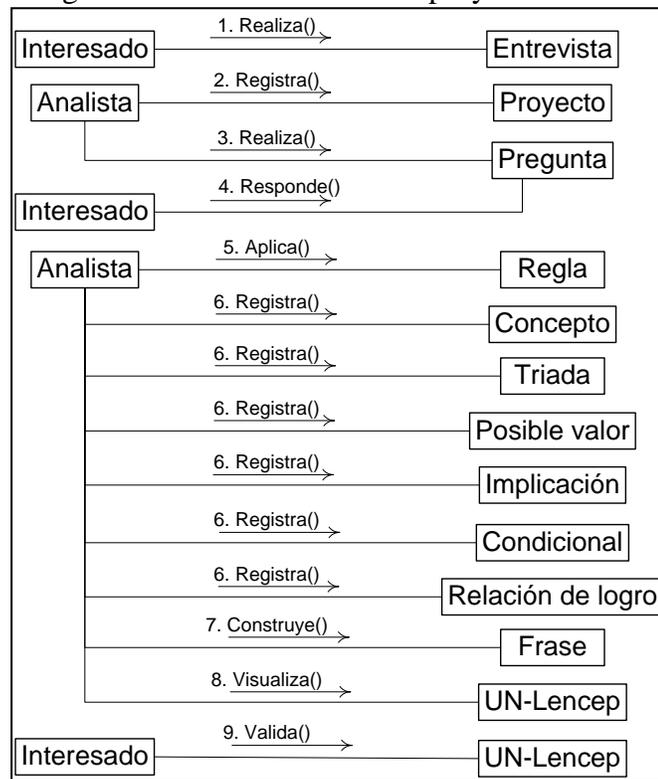




6.2.3.2 *Diagrama de Comunicación.*

La figura 46 muestra el diagrama de comunicación del proyecto modelo de diálogo. En este diagrama se modela la secuencia de las comunicaciones entre los objetos o sus partes y hace parte de los diagramas de interacción (Fowler & Scott, 1999).

Figura 46. Diagrama de Comunicación del proyecto Modelo de Diálogo



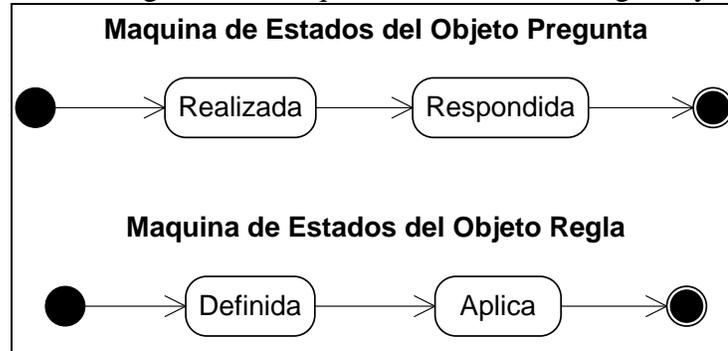
6.2.3.3 *Diagrama de Máquina de Estados.*

La figura 47 muestra los diagramas de máquina de estados de los objetos Pregunta y Regla. Estos ilustran y explican el movimiento y el comportamiento de un objeto,



para determinar cómo este se mueve entre sus diferentes estados y hacen parte de los diagramas de comportamiento (Fowler & Scott, 1999).

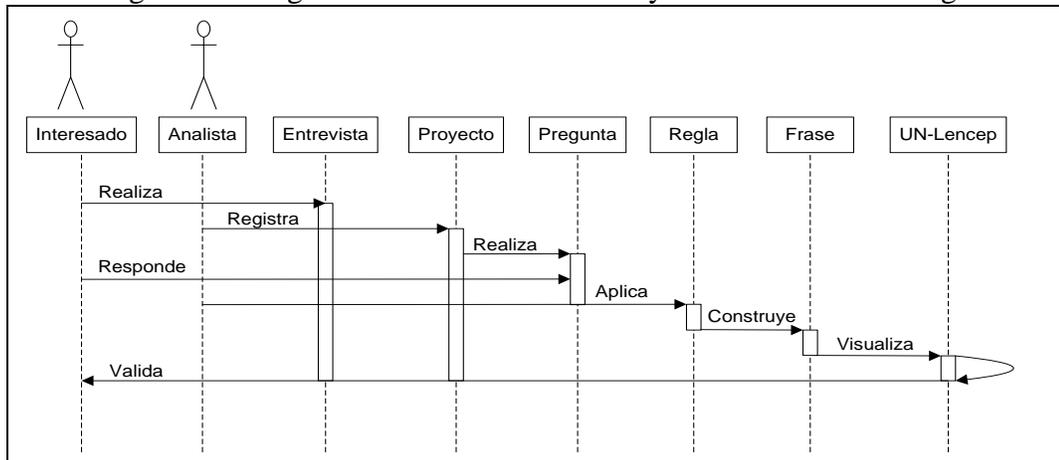
Figura 47. Diagrama de Máquina de Estados de Pregunta y Regla



6.2.3.4 Diagrama de Secuencias.

La figura 48 presenta el diagrama de secuencias, el cual describe de forma estructurada el comportamiento y flujo del trabajo, el paso de mensajes y cómo los elementos cooperan, en el desarrollo del proceso, para lograr un resultado. Este diagrama hace parte de los diagramas de interacción (Fowler & Scott, 1999).

Figura 48. Diagrama de secuencias del Proyecto Modelo de Diálogo





6.2.3.5 Diagrama de Casos de Uso.

Las figuras 49, 50 y 51 muestran los diagramas de casos de uso para el proyecto modelo de diálogo. Estos diagramas capturan las interacciones de los objetos con los actores y la respuesta del sistema (Fowler & Scott, 1999).

Figura 49. Diagrama de Casos de Uso Realizar Entrevista

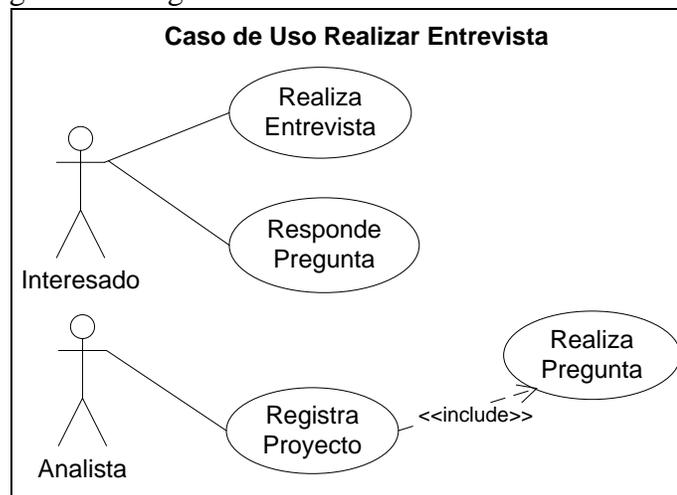


Figura 50. Diagrama de Casos de Visualizar UN-Lencep

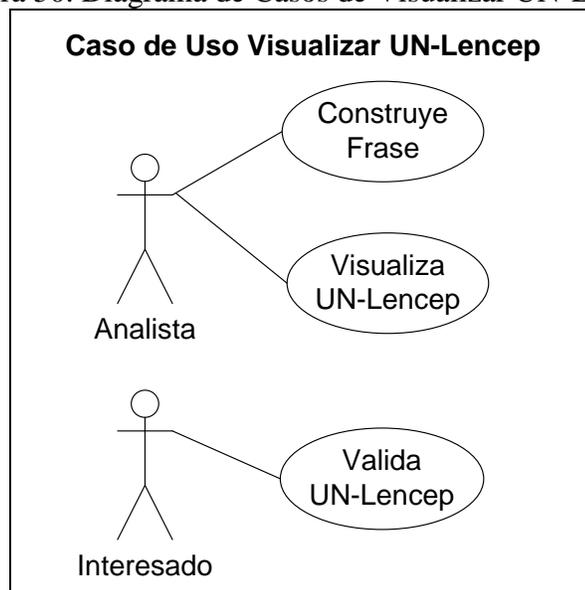
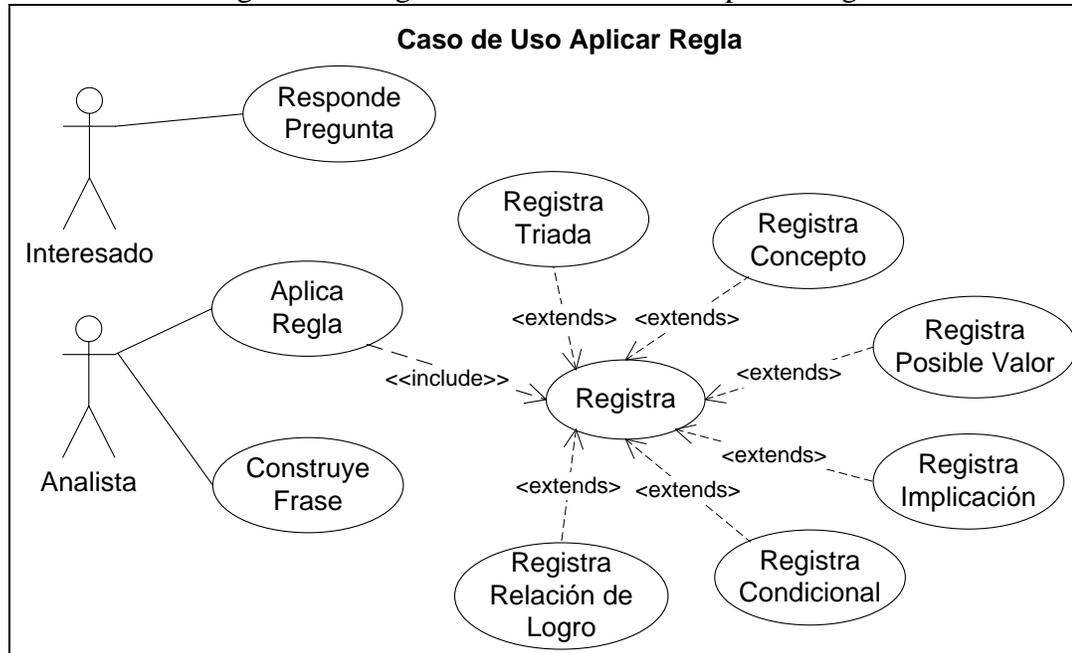




Figura 51. Diagrama de Casos de Uso Aplicar Regla



6.3 Caso de estudio: Herramienta de Apoyo a la Gestión de Proyectos Basada en el Modelo de Proceso Competisoft

Competisoft es una iniciativa para mejorar los procesos y fomentar la competitividad en la MiPyME (Micro, Pequeña y Mediana Empresa), o áreas internas de la empresa, dedicadas al desarrollo o mantenimiento de software, mediante la creación y difusión de un marco metodológico común, ajustado a las necesidades específicas del sector y que se soporta en las mejores prácticas internacionales. Para ello, Competisoft presenta un modelo de referencia, un modelo de evaluación y un modelo de mejora de procesos fácil de comprender e implementar (Piattini *et al.*, 2008).

En este orden de ideas, Competisoft se proyecta como base para establecer un mecanismo de evaluación y certificación de la industria del software, el cual sea reconocido y acogido



en toda Iberoamérica. Así, se encuentra una iniciativa para establecer un estándar ISO para la certificación de las empresas, con base en el modelo competisoft (Oktaba *et al.*, 2007).

Las respuestas que presenta la tabla 15 son tomadas de la experiencia en la implementación del modelo en una MiPyME, como una alternativa con la cual se pueda definir y hacer seguimiento a los procesos dentro de la misma.

Tabla 15. Respuestas de la entrevista para el proyecto herramienta de apoyo basada en el Modelo Competisoft

Pregunta	Respuesta
P1	Competisoft
P2	Mejora de procesos
P3	Usuario, Grupo Directivo, Responsable del Proceso, Involucrado
P4	Si
P5	Usuario se agrupan en Cliente Cliente, Grupo Directivo, Responsable del Proceso, Involucrado se agrupan en Rol
P6	Rol: Abreviatura, Nombre Rol
P7	No
P8	No
P10	Ninguno
P11	Cliente Solicita Software, Cliente Financia Software, Usuario Utiliza Software, Grupo Directivo Establece Patrón de Proceso, Responsable del Proceso Utiliza Patrón de Proceso, Rol Realiza Actividad, Responsable del Proceso Evalúa Actividad, Responsable del Proceso Elabora Guía de Ajuste
P12	No
P14	Software: Nombre, Cliente, Costo, Fecha de Inicio, Fecha de Finalización Patrón de Proceso: Nombre del Proceso, Categoría, Propósito, Detalle, Objetivo, Responsable del Proceso, Proceso Relacionado, Producto Interno, Actividad, Diagrama de Flujo, Guía de Ajuste Guía de Ajuste: Identificación Guía de Ajuste, Descripción Actividad: Identificación Actividad, Nombre de la Actividad, Sub actividad, Objetivo de la actividad, Entrada, Salida, Recursos de infraestructura Entrada: Nombre de la Entrada, Fuente Salida: Nombre de la Salida, Destino Sub actividad: Rol, Identificación Sub actividad, Nombre Sub actividad, Descripción sub actividad, Entrada, Salida
P8	Si
P9	Categoría: Alta Dirección (DIR), Gerencia (GER), Operación (OPE)



Continuación Tabla 15. Respuestas de la entrevista para el proyecto herramienta de apoyo basada en el Modelo Competisoft

Pregunta	Respuesta
P15	Primero Cliente Solicita Software luego Cliente Financia Software Primero Cliente Solicita Software luego Rol Realiza Actividad Primero Grupo Directivo Establece Patrón de Proceso luego Responsable del Proceso Utiliza Patrón de Proceso Primero Responsable del Proceso Utiliza Patrón de Proceso luego Rol Realiza Actividad Primero Responsable del Proceso Utiliza Patrón de Proceso Responsable del Proceso Evalúa Actividad Primero Responsable del Proceso Evalúa Actividad luego Responsable del Proceso Elabora Guía de Ajuste Primero Rol Realiza Actividad luego Responsable del Proceso Evalúa Actividad Primero Responsable del Proceso Elabora Guía de Ajuste luego Grupo Directivo Establece Patrón de Proceso
P16	No
P17	Se debe Desarrollar el Patrón de Proceso, Incrementar que Grupo Directivo Establece Patrón de Proceso, Fomentar que Responsable del Proceso Utiliza Patrón de Proceso, Garantizar que Rol Realiza Actividad, Garantizar que Patrón de Proceso tiene Actividad

6.3.1 Discurso en UN–Lencep del Proyecto Herramienta de Apoyo Basada en el Modelo Competisoft

El discurso que se presenta a continuación se generó con las respuestas de la entrevista que se presentan en la tabla 15. Para ello, se usó el prototipo que automatiza la propuesta de esta Tesis, con lo cual se demuestra la funcionalidad del método propuesto. Así, la aplicación del método, tanto de forma manual como automática, produce los resultados esperados. De otro lado, la información que presenta el discurso concuerda con las respuestas de la entrevista.

Actividad tiene Entrada

Actividad tiene Identificación Actividad

Actividad tiene Nombre de la Actividad



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

Actividad tiene Objetivo de la actividad
Actividad tiene Recursos de infraestructura
Actividad tiene Salida
Actividad tiene Sub actividad
Cliente es Rol
Cliente Financia Software
Cliente Solicita Software
Entrada tiene Fuente
Entrada tiene Nombre de la Entrada
Grupo Directivo es Rol
Grupo Directivo Establece Patrón de Proceso
Guía de Ajuste tiene Descripción
Guía de Ajuste tiene Identificación Guía de Ajuste
Involucrado es Rol
Patrón de Proceso tiene Actividad
Patrón de Proceso tiene Categoría
Patrón de Proceso tiene Detalle
Patrón de Proceso tiene Diagrama de Flujo
Patrón de Proceso tiene Guía de Ajuste
Patrón de Proceso tiene Nombre del Proceso
Patrón de Proceso tiene Objetivo
Patrón de Proceso tiene Proceso Relacionado
Patrón de Proceso tiene Producto Interno
Patrón de Proceso tiene Propósito
Patrón de Proceso tiene Responsable del Proceso
Responsable del Proceso Elabora Guía de Ajuste
Responsable del Proceso es Rol
Responsable del Proceso Utiliza Patrón de Proceso
Rol Realiza Actividad



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

Rol tiene Abreviatura

Rol tiene Nombre Rol

Salida tiene Destino

Salida tiene Nombre de la Salida

Software tiene Cliente

Software tiene Costo

Software tiene Fecha de Finalización

Software tiene Fecha de Inicio

Software tiene Nombre

Sub actividad tiene Descripción sub actividad

Sub actividad tiene Entrada

Sub actividad tiene Identificación Sub actividad

Sub actividad tiene Nombre Sub actividad

Sub actividad tiene Rol

Sub actividad tiene Salida

Usuario es Cliente

Usuario Utiliza Software

Alta Dirección (DIR) es un posible valor de Categoría

Gerencia (GER) es un posible valor de Categoría

Operación (OPE) es un posible valor de Categoría

Cuando Cliente Solicita Software, Rol Realiza Actividad

Cuando Cliente Solicita Software, Cliente Financia Software

Cuando Grupo Directivo Establece Patrón de Proceso, Responsable del Proceso Utiliza Patrón de Proceso

Cuando Responsable del Proceso Utiliza Patrón de Proceso, Responsable del Proceso Evalúa Actividad

Cuando Responsable del Proceso Utiliza Patrón de Proceso, Rol Realiza Actividad

Cuando Responsable del Proceso Elabora Guía de Ajuste, Grupo Directivo Establece Patrón de Proceso



Cuando Rol Realiza Actividad, Responsable del Proceso Evalúa Actividad

Cuando Responsable del Proceso Evalúa Actividad, Responsable del Proceso Elabora
Guía de Ajuste

Desarrollar: Patrón de Proceso

Fomentar: Responsable del Proceso Utiliza Patrón de Proceso

Garantizar: Patrón de Proceso tiene Actividad

Garantizar: Rol Realiza Actividad

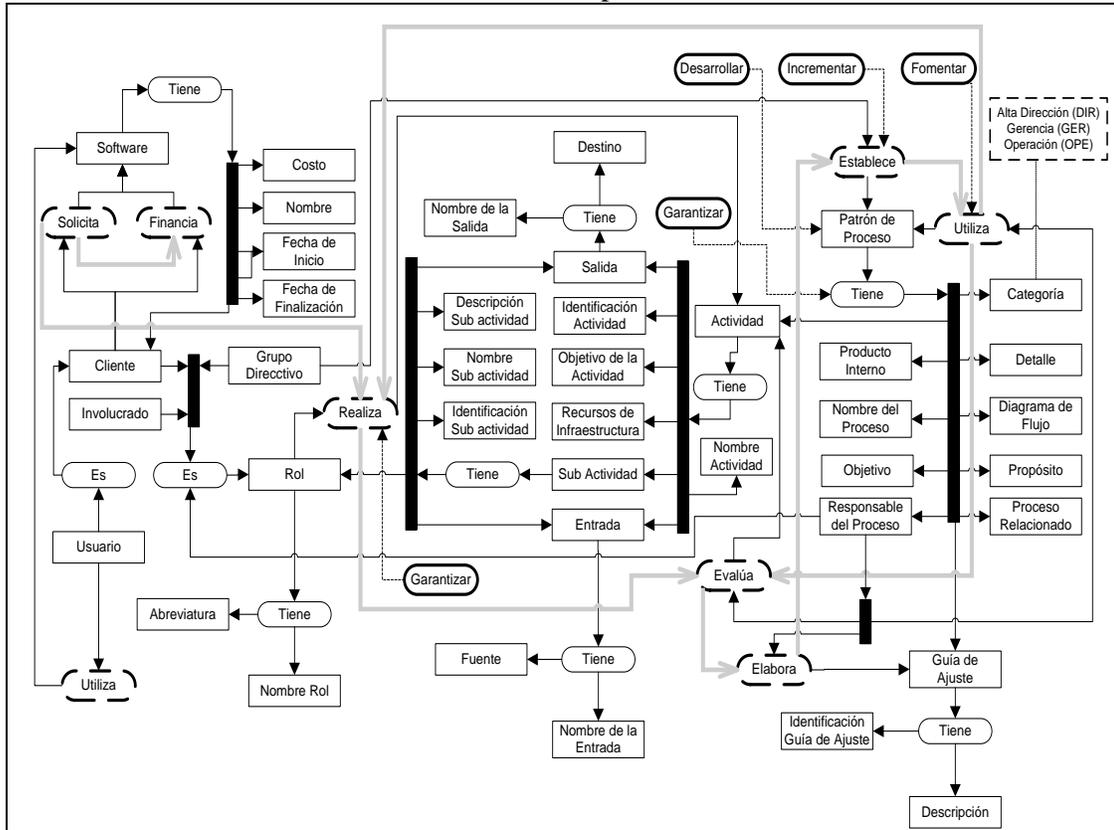
Incrementar: Grupo Directivo Establece Patrón de Proceso

6.3.2 Esquema Preconceptual del Proyecto Herramienta de Apoyo Basada en el Modelo Competisoft

Pese a que el alcance del método que se propone en esta Tesis no incluye la generación del EP, en la figura 52 ilustra el EP que se obtuvo con el discurso del numeral anterior. Así, se representa gráficamente la información que del dominio del problema. Cabe aclarar que tanto el EP como los esquemas conceptuales de UML que se presentan posteriormente no hacen parte de las funcionalidades incluidas en el prototipo, estos se obtienen utilizando la propuesta de Zapata *et al.* (2007). No obstante, con esta información se realiza la especificación y el diseño de un producto de software. Por esta razón, se incluyen dentro del caso de estudio.



Figura 52. Esquema Preconceptual del proyecto herramienta de apoyo basada en el Modelo Competisoft



6.3.3 Esquemas Conceptuales de UML del Proyecto Herramienta de Apoyo Basada en el Modelo Competisoft

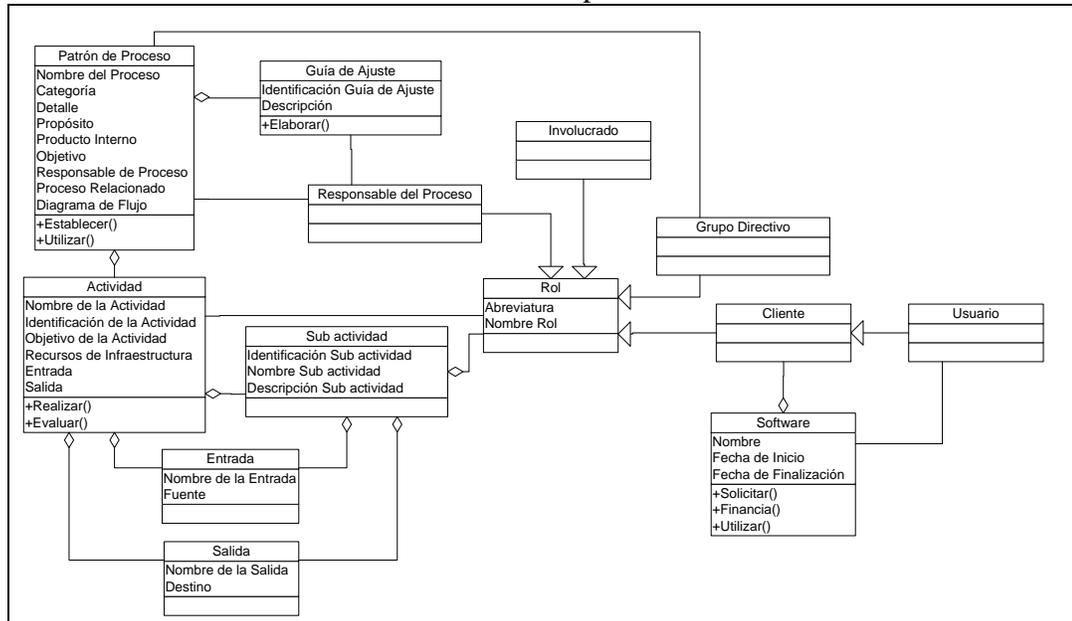
Del esquema que se presenta en el numeral anterior, mediante tratamiento computacional, se obtiene el diagrama de clases, el diagrama de comunicaciones, el diagrama de máquina de estados y el diagrama de casos de uso (Zapata *et al.*, 2007), los cuales se presentan a continuación.



6.3.3.1 Diagrama de Clases.

La figura 53 muestra el diagrama de clases, con el cual se describe los objetos que se identificaron en el discurso y que harán parte del sistema, asimismo, las relaciones entre ellos.

Figura 53. Diagrama de Clases del proyecto herramienta de apoyo basada en el Modelo Competisoft

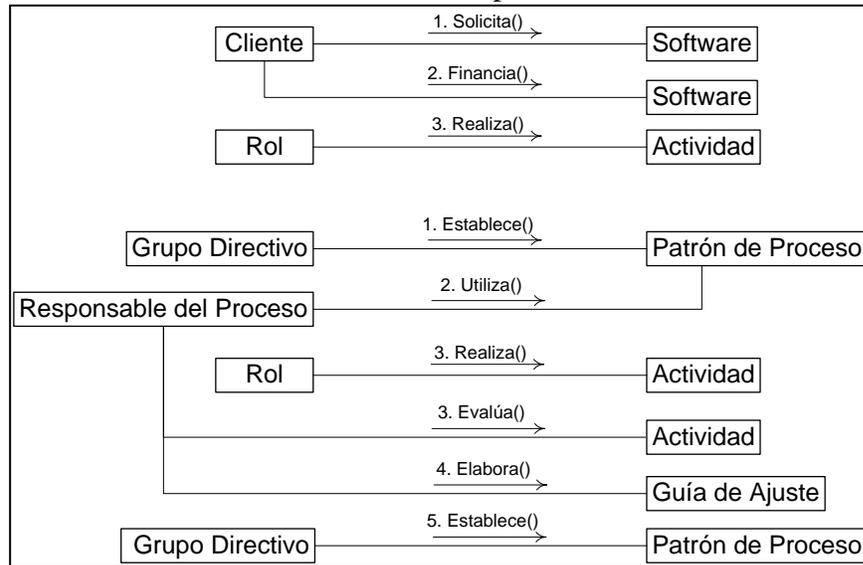


6.3.3.2 Diagrama de Comunicación.

La figura 54 muestra el diagrama de Comunicación obtenido que refleja la interacción de los actores con los objetos.



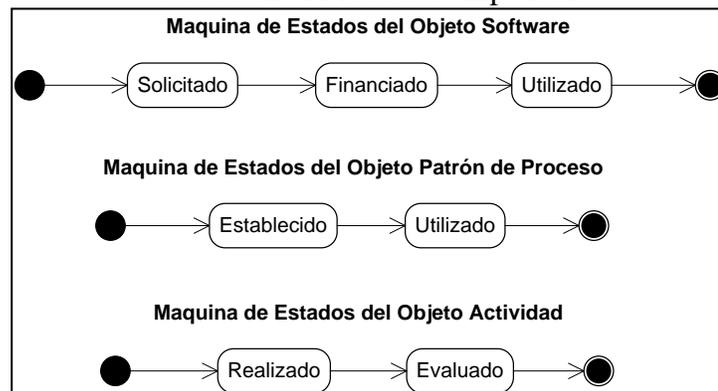
Figura 54. Diagrama de comunicación del proyecto herramienta de apoyo basada en el Modelo Competisoft



6.3.3.3 Diagrama de Máquina de Estados.

La figura 55 muestra los diagramas de máquina de estados de los objetos Software y Patrón de Proceso, los cuales detallan la secuencia de las actividades que se realizan sobre cada objeto.

Figura 55. Diagrama de Máquina de Estados para el proyecto herramienta de apoyo basada en el Modelo Competisoft

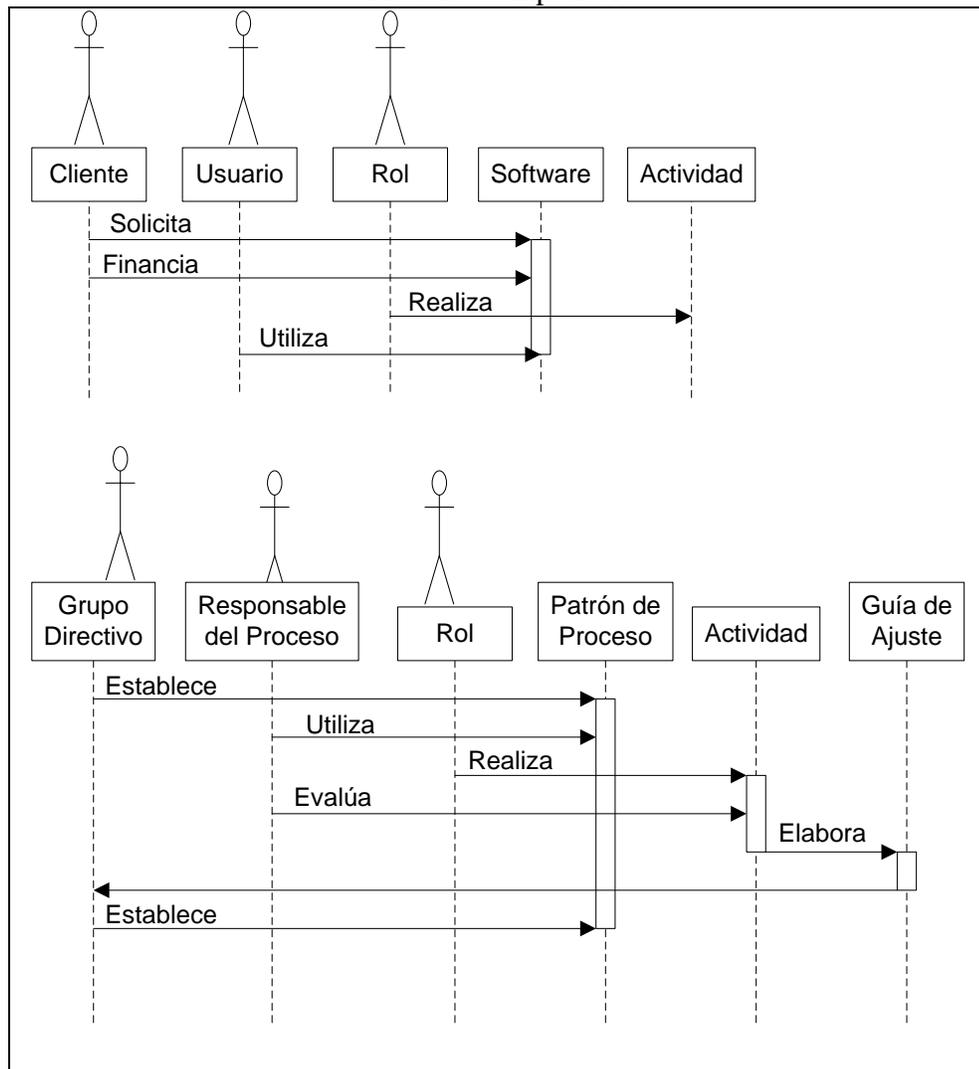




6.3.3.4 *Diagrama de Secuencias.*

El diagrama de la figura 56 representa de forma estructurada el comportamiento y flujo del trabajo, el paso de mensajes y la cooperación entre los objetos, en el desarrollo del proceso (Fowler & Scott, 1999) de uso de la herramienta de apoyo basada en el Modelo Competisoft.

Figura 56. Diagrama de secuencias del Proyecto herramienta de apoyo basada en el Modelo Competisoft





6.3.3.5 Diagrama de Casos de Uso.

Las figuras 57 y 58 muestran los diagramas de caso de uso Solicitar Software y Establecer Patrón de Proceso, los cuales se obtienen con el discurso del numeral 6.3.1.

Figura 57. Diagrama de Caso de Uso Solicitar Software

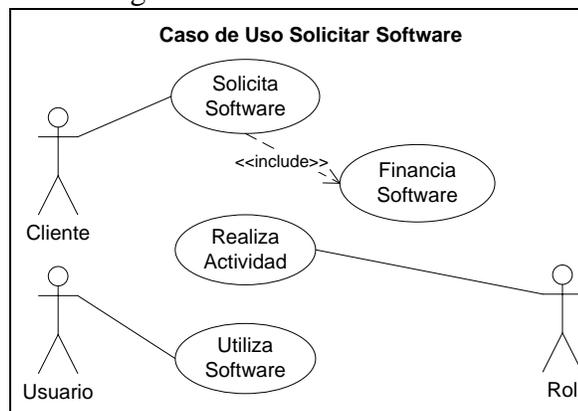
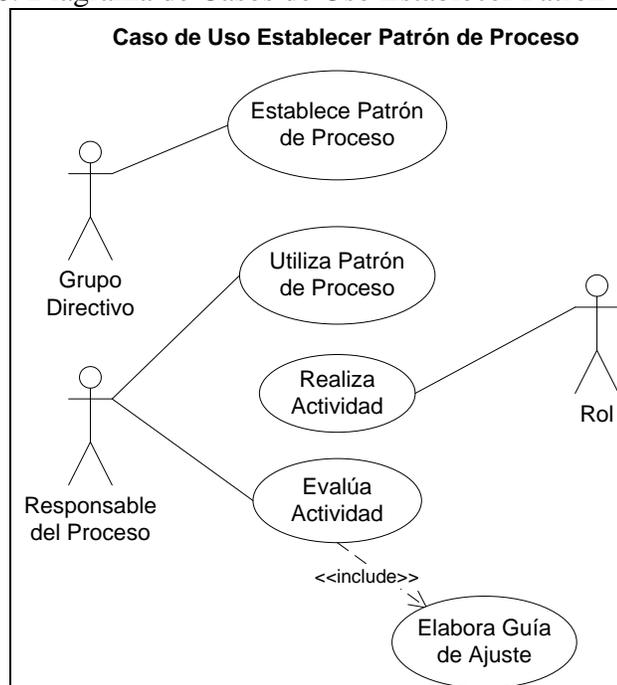


Figura 58. Diagrama de Casos de Uso Establecer Patrón de Proceso





VII. CONCLUSIONES

El uso del modelo de diálogo para la obtención del discurso del interesado en UN-Lencep para la especificación de requisitos de software, mejora la comunicación entre el analista y el interesado. Además, disminuye los errores de comunicación que plantea Ongallo (2007). De otro lado, el modelo de la entrevista estructurada permite que el interesado responda de forma concreta y concisa, eliminando información innecesaria en el diálogo. De este modo, se aporta completitud y consistencia en la especificación de requisitos, pues el discurso, resultado del proceso, lo puede validar el interesado.

Además, con el uso de los aportes realizados en trabajos como Zapata *et al.* (2006), Zapata y Arango (2007) y Zapata *et al.* (2007), entre otros, se pueden obtener diferentes artefactos UML, que se utilizan en la especificación, modelado y diseño de una aplicación de software. De esta forma, se automatizan estos procesos, disminuyendo la intervención del analista, con lo cual se reducen los posibles errores humanos y se mantiene la consistencia entre las necesidades del interesado y la especificación de requisitos.

La definición de reglas para la traducción de las respuestas del entrevistado a las frases del discurso en UN-Lencep, garantizan la consistencia de la información. Además, se plantea la posibilidad de incluir nuevas reglas que ayuden en la obtención de elementos que en el futuro hagan parte del discurso. Esto, teniendo en cuenta que el conocimiento es dinámico, entonces, si cambia la estructura del Lenguaje Controlado, el modelo se puede modificar para seguir cumpliendo la tarea para la cual se desarrolló.

El prototipo que se implementó ofreció entre otras, las siguientes ventajas: la posibilidad de guardar el discurso en UN-Lencep que se genera en cada entrevista. Adicionalmente, se reutiliza la información, que se encuentra en la base de datos, pero se adaptada al proyecto para el cual se realiza la entrevista, optimizando la funcionalidad del prototipo.



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

El modelo propuesto en esta Tesis, es una herramienta de mejora para la MiPyME (Micro, Pequeña y Mediana Empresa) productoras de software. Debido, precisamente, a que estas empresas presentan poco personal y recursos para el desarrollo de su objeto comercial. De este modo, con el uso de esta propuesta en la MiPyME, se disminuye los errores y por ende los costos de corrección de los mismos.

De otro lado, el uso del modelo como herramienta pedagógica, ayuda al estudiante en la identificación de las necesidades del interesado. Además, propone un punto de partida y de entendimiento, entre los actores del diálogo. Así, le indica al estudiante las preguntas que le permiten entender el dominio del problema, mientras que el interesado se limita a las posibles respuestas, las cuales se ejemplifican para un mejor entendimiento. Por otro lado, el estudiante desarrolla la habilidad de mantener la atención del interesado en el tema de discusión.



VIII. TRABAJO FUTURO

Pese a los diversos aportes que se realizaron con anterioridad y que sustentan todo el trabajo realizado en torno al Lenguaje Controlado UN-Lencep. Las líneas de investigación futura de esta Tesis que se deben considerar para mejorar la propuesta presentada son, entre otras, las siguientes.

Definir un modelo de preguntas para la obtención de los elementos del UN-Lencep avanzado, el cual es una versión de UN-Lencep con elementos más cercanos al lenguaje natural. Para llevar a cabo este complemento, se debe tener en cuenta la definición de las reglas que permitan la transformación de las respuestas a los elementos del UN-Lencep Avanzado.

En este sentido, otro aporte futuro es la inclusión de técnicas de procesamiento del habla, de forma que, al realizar la entrevista, el interesado pueda contestar en este formato, sin necesidad de digitar su respuesta. Así, se disminuye la participación del analista, pero el interesado podría ver el discurso y validarlo al final del proceso. Sin embargo, este aporte requiere definir las características específicas de la instrucción del condicional, además de otros elementos.

De otro lado, el prototipo desarrollado, presenta tres funcionalidades que se consideran necesarias para el objetivo propuesto. No obstante, una vez que se inicia la entrevista se debe concluir, pues no se presenta la posibilidad de guardar el avance para retomarlo posteriormente. Tampoco, se encuentra la opción para editar un discurso generado o guardado en la base de datos.



REFERENCIAS

- Al-Salem, L. & Abu-Samaha, A. (2007), Eliciting Web application requirements – an industrial case study. *The Journal of Systems and Software*, Vol. 80, pp. 294–313.
- Aguilera, M. & Mañas, S. (2001), Atravesando el espejo. *Comunicar: Revista Científica Iberoamericana de Comunicación y Educación*, No. 17, pp. 79–85.
- Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L. Stent, A. (2001), Towards Conversational Human-Computer Interaction. *AI Magazine*, Vol. 22, No 4, pp. 27–37.
- Bertrán, M. & Xampeny, J. (1978), Tratamiento unificado de la comunicación y transformación de procesadores. *Qüestiió*, Vol. 2, No. 3, pp. 181–194.
- Blythe, J., Kim, J., Ramachandran, S., Gil, Y. (2001), An Integrated Environment for Knowledge Acquisition. En: *International Conference on Intelligent User Interfaces*. New Mexico, USA.
- Caglioti, V., Giusti, A., Riva, A., Uberti, M. (2009), Drawing Motion Without Understanding It. En: *International Symposium on Advances in Visual Computing*, pp. 147–156. Las Vegas, USA.
- Clark-Carter, D. (2002), *Quantitative Psychological Research*. Oxford, Inglaterra: Oxford University press.
- Camacho, S.M. (2006), SAMI: Un SOFBOT de charla desarrollado con la técnica de razonamiento basado en casos. En: *Congreso Internacional de Telemática y Telecomunicaciones*. La Habana, Cuba.
- Fowler, M. & Scott, K. (1999), *UML gota a gota*. Ciudad de Mexico, Mexico: Adison Wesley.
- Garner, K., Ylvén, J., Karlsson, M. (2004), User participation in requirements elicitation comparing focus group interviews and usability tests for eliciting usability requirements for medical equipment: a case study. *International Journal of Industrial Ergonomics* Vol. 33, No. 2, pp. 85–98.
- Gil, Y., Blythe, J., Kim, J., Ramachandran, S. (2000), Acquiring procedural knowledge in EXPECT. En: *AAAI – 2000 Fall Symposium*. Massachusetts, USA.



- Gilvaz, A. & Leite, J. (1995), FAES: A Case Tool for Information Acquisition. En: 7th International Workshop on Computer-Aided Software Engineering, pp. 260–269. Toronto, Canada.
- González, C. (2008), Social Behavior Simulator. Generación y aplicación de un ser humano simulado para el estudio de la interacción social diádica. *Revista Iberoamericana de Inteligencia Artificial*, Vol. 12, No. 38, pp. 61–73.
- Griol, D., Hurtado, L., Sanchis, E., Segarra, E. (2005), Dos aproximaciones basadas en reglas para la gestión del diálogo. *Procesamiento del lenguaje natural*, Vol. 35, pp. 213–220.
- Griol, D., Torres, F., Hurtado, L., Grau, S., García, F., Sanchis, E., Segarra, E. (2006), A dialog system for the DIHANA Project. En: 11th International Conference Speech and Computer. San Petesburgo, Rusia.
- Hurtado, L., Blat, F., García, F., Grau, S., Griol, D., Sanchis, E., Segarra, E., Torres, F. (2005), Sistema de diálogo para el Proyecto DIHANA. *Procesamiento del Lenguaje Natural*, Vol. 35, pp. 453–454.
- Hwang, W., Lee, P., Chun, B., Ryu, D., Cho, H. (2006), Cinema comics: cartoon generation from video stream. En: 1st International Conference on Computer Graphics Theory and Applications, pp. 299-304. Setubal, Portugal.
- IEEE. (1998), Recommended Practice for Software Requirements Specifications Software, Standard 830, Engineering Standards Committee of the IEEE Computer Society.
- Kassel, N. & Malloy, B.A. (2003), An Approach to Automate Requirements Elicitation and Specification. En: International Conference Software Engineering and Applications. Marina del Rey, USA.
- Kato, J., Komiya, S., Saeki, M., Ohnishi, A., Nagata, M., Yamamoto, S., Horai, H. (2001), A model for navigating interview processes in requirement elicitation. En: 8th Asia-Pacific on Software Engineering Conference, pp. 141–148. Macao, China.
- Kozima, A., Kiguchi, T., Yaegashi, R., Kinoshita, D., Hayashi, Y., Hashiura, H., Komiya, S. (2005), A system to guide interview-driven requirements elicitation work: domain–



- specific navigation using the transition pattern of topics. *Journal of Integrated Design & Process Science*, Vol. 9, No. 4, pp. 27–39.
- Kruijff, G., Zender, H., Jensfel, P., Christensen, H. (2007), *Situated Dialogue and Spatial Organization: What, where... and why?*. *International Journal of Advanced Robotic Systems*, Special Issue on Human and Robot Interactive Communication, Vol. 4, No. 2, pp. 125–138.
- Lecoeuche, R., Mellish, C., Robertson, D. (1998), *A Framework for requirements elicitation through mixed-initiative*. En: *3rd International Conference on Requirements Engineering*, pp. 190–197. Colorado Springs, USA.
- Lecoeuche, R., Robertson, D., Barry, C. (1999) *Using Focus Rules in Requirements Elicitation Dialogues*. En: *6th International Joint Conference on Artificial Intelligence*. pp. 649–654, Tokyo, Japan.
- Leffingwell, D. & Widrig, D. (2003), *Software Requirements: A Use Case Approach*. Boston, USA: Pearson.
- Leite, J. (1987) *A survey on requirements analysis*. *Advanced Software Engineering Project*. Reporte Técnico RTP–071. University of California at Irvine.
- Leite, J. & Gilvaz, A. (1996), *Requirements Elicitation Driven by Interviews: The Use of Viewpoints*. En: *8th International Workshop on Software Specification and Design*, pp. 85–94. Schloss Velen, Alemania.
- Liu, C., Conn, K., Sarkar, N., Stone, W. (2008) *Online Affect Detection and Robot Behavior Adaptation for Intervention of Children with Autism*. *IEEE Transactions on Robotics*, Vol.24, No.4, p883–896.
- López-Cózar, R. & Rubio, A. (1997), *SAPLEN: un sistema de diálogo en lenguaje natural para una aplicación comercial*. *Procesamiento del lenguaje natural*, Vol. 20, pp. 65–81.
- López-Cózar, R., Rubio, A., Benítez, M., Milone, D. (2000), *Restricciones de Funcionamiento en Tiempo Real de un Sistema Automático de Diálogo*. *Procesamiento del lenguaje natural*, Vol. 26, pp. 169–174.



- Millard, N., Lynch, P., Tracey, K. (1998), Child's play: using techniques developed to elicit requirements from children with adults. En: 3rd Requirements Engineering International Conference, pp. 66–73. Colorado Springs, USA.
- Mohd, Z. & Salwah, S. (2008), Focus Group Discussion Model for Requirements Elicitation Activity. En: International Conference on Computer and Electrical Engineering, pp. 101–105. Phuket Islas, Tailandia.
- Nakagawa, S., Kogure, S., Itoh, T. (2000), A semantic interpreter and a cooperative response generator for a robust spoken dialogue system. *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 14, No. 5, pp. 553–569.
- Oktaba, H., García, F., Piattini, M., Ruiz, F., Pino, F., Alquicira, C. (2007), Software Process Improvement: The Competisoft Project. *IEEE Computer*, Vol 40, No.10, p. 21–28.
- Ongallo, C. (2007), Manual de comunicación. Madrid, España: Dykinson.
- Ross, D. (1977), Structured Analysis (SA): A Language for Communicating Ideas. *IEEE Transactions on Software Engineering*, Vol. 3, No. 1, pp. 16–34.
- Peña, M., López-Juárez, I., Corona, J., Ordaz, K. (2002) Visión para Robots en Tareas de Ensamble. En: 2^o Congreso Nacional de Mecatrónica. Ciudad de México, México.
- Piattini, M., Oktaba, H., Orozco, M., Alquicira, C. (2008), Competisoft. Mejora de procesos software para pequeñas y medianas empresas y proyectos. Mexico: RA-MA Editorial.
- Prendinger, H., Morib, J., Ishizuka, M. (2005), Using human physiology to evaluate subtle expressivity of a virtual quizmaster in a mathematical game. *International Journal of Human–Computer Studies*, Vol. 62, No. 2, p. 231–245.
- Prendinger, H., Piwek, P., Ishizuka M. (2007), Automatic Generation of Multi–Modal Dialogue from Text Based on Discourse Structure Analysis. En: 1st International Conference on Semantic Computing, pp. 27–36. Irvine, USA.
- Shahidi, S. & Mohd, Z. (2009), Using Ethnography Techniques in Developing a Mobile Tool for Requirements Elicitation. En: International Conference on Information Management and Engineering, pp. 510–513. Kuala Lumpur, Malaysia.



- Sommerville, I. (2007). *Software Engineering*. Londres, Inglaterra: Editorial Pearson.
- Suzuki, K., Umeda, M., Kogure, S., Nakagawa, S. (2004), Development of a Portable Spoken Dialogue System for Database Retrieval and Reservation. *Joho Shori Gakkai Kenkyu Hokoku*, Vol. 2004, No. 103, pp. 25–30.
- Varges, S., Weng, F., Pon-Barry, H. (2008), Interactive question answering and constraint relaxation in spoken dialogue systems. En: *Natural Language Engineering*, Vol. 15, No.1, p. 9-30.
- Villaseñor, L., Xuereb, A., Caelen, J., Montes, M. (2004), Un Modelo de Interpretación Pragmática basado en la SDRT para el Diálogo Hombre–Máquina. En: *3ª Jornadas en Tecnología del Habla*. Valencia, España.
- Wang, M. & Zeng, Y. (2009), Asking the right questions to elicit product requirements. *International Journal of Computer Integrated Manufacturing*, Vol. 22, No. 4, pp. 283–298.
- Williams, S., Piwek, P., Power, R. (2007), Generating monologue and dialogue to present personalised medical information to patients. En: *11th European Workshop on Natural Language Generation*, pp. 167-170. Schloss Dagstuhl, Alemania.
- Zapata, C. & Arango, F. (2007), Un ambiente para la obtención automática de diagramas UML a partir de un lenguaje controlado. *Revista Dyna*, Vol. 74, No. 153, pp. 223–236.
- Zapata, C., Gelbukh, A., Arango, F. (2006). Pre-conceptual Schema: a UML Isomorphism for Automatically Obtaining UML Conceptual Schemas. *Research in Computing Science: Advances in Computer Science and Engineering*, Vol. 19, pp. 3–13.
- Zapata, C. & Giraldo G. (2009), El juego del diálogo de educación de requisitos. *Revista Avances en Sistemas e Informática*. Vol. 6, No. 1, pp. 105–114.
- Zapata, C. & Lezcano, L. (2009), Caracterización de los verbos usados en el diagrama de objetivos. *Revista Dyna*, Vol. 76, No. 158, pp. 219–228.
- Zapata, C. & Rosero, R. (2008), Revisión crítica de la literatura especializada en lenguajes controlados. *Revista Avances en Sistemas e Informática*, Vol. 5, No. 3, pp. 27–33.



Método de conversión de un diálogo controlado a un discurso en UN-Lencep

Tesis de Maestría en Ingeniería – Ingeniería de Sistemas

Universidad Nacional de Colombia.

William A. Arévalo Camacho

Zapata, C., Ruiz, L., Villa, F. (2007). UNC–Diagramador una herramienta upper CASE para la obtención de diagramas UML desde esquemas preconceptuales. Revista Universidad EAFIT, Vol. 43, No. 143, pp. 68–80.

Zender, H. & Kruijff, G. (2007), Towards Generating Referring Expressions in a Mobile Robot Scenario. En: Symposium on Language and Robots, pp. 101–106. Aveiro, Portugal.