

# Desarrollo de un Sistema Experto para la Especificación de Requerimientos

Alexandra Corral y José Luis Carrillo-Medina

*Departamento de Eléctrica y Electrónica de la Universidad de las Fuerzas Armadas*

*ESPE Extensión Latacunga*

**Abstract**— La complejidad de los proyectos de software, así como el carácter multidisciplinar de ingeniería de requisitos obliga a los desarrolladores a seleccionar cuidadosamente las técnicas y prácticas de especificación de requerimientos durante el proceso de desarrollo de software. Uno de los mayores problemas en la especificación de requerimientos es trasladar de un lenguaje natural a una especificación, en la que el requerimiento tenga aplicación en el dominio, este traslado se ve comúnmente afectado por la mala comprensión del dominio de la aplicación, el mayor trabajo para el ingeniero de software es explorar sobre ámbitos desconocidos y trabajar sobre un lenguaje en el que, tanto el usuario como el ingeniero, mantienen un grado de abstracción diferente. En este artículo nos centraremos en los problemas de la especificación de requerimientos asistidos y/o apoyados por un sistema experto: indicando lo que se debe especificar. Una de las técnicas utilizadas para la especificación de requerimientos es determinar los tipos de marcos de problemas con la finalidad de obtener la tipificación de requerimientos. Hemos implementado un sistema experto el cual permite obtener un documento de especificación de requerimientos basado en el estándar IEEE 830, documento entendible y legible para las fases iniciales del desarrollo de un producto software.

**Palabras claves**— Ingeniería de requisitos, Análisis de problemas, Dominio de la Aplicación, Especificación de requerimientos, Marcos de problema, Tipificación de requerimientos, Sistemas Expertos.

## I. INTRODUCCIÓN

EL reto de las empresas desarrolladoras de software es implementar aplicaciones que sean entregadas a tiempo, que no involucren presupuestos elevados y que satisfagan las necesidades del usuario, utilizando para su efecto metodologías y herramientas que guíen el proceso de desarrollo de Software.

La ingeniería de requisitos es la ciencia y la disciplina que se ocupa del análisis, documentación y validación de los deseos, necesidades y/o requerimientos de las partes interesadas para el desarrollo de un sistema [1]. Jiang et al. [2], considera que la mala práctica de la ingeniería de requisitos contribuye al fracaso de proyectos de software. Macaulay [3] ha demostrado los beneficios de la ingeniería de requisitos, mostrando muchas evidencias empíricas, mientras Nuseibe et al. [4] ha mostrado el impacto positivo de la utilización de técnicas de ingeniería de requisitos adecuadas para mejorar la calidad del software.

Siendo la ingeniería de requisitos una de las actividades más

complicadas e importantes para construir un producto software, hablamos específicamente de la elicitación de requerimientos que es una subactividad de la ingeniería de requisitos la cual tiene un alto impacto en el diseño y en las fases posteriores del ciclo de vida de un producto software. Si esta se realiza apropiadamente, ayuda a reducir cambios y correcciones en los requisitos. La elicitación de requerimientos se considera como la primera etapa en el proceso de abstracción, comprensión y concepción de las funcionalidades que un producto software debe cumplir. Definiéndose como requerimiento una condición o capacidad que debe cumplir o poseer un sistema para satisfacer un contrato, estándar, especificación u otro documento formalmente impuesto.

La elicitación esencialmente es una actividad humana donde se identifican las partes interesadas y se establecen las relaciones entre el comprador, el cliente, los usuarios y el equipo de desarrollado. De acuerdo a estudios realizados en [5] el 79% de todos los documentos de requisitos están escritos en lenguaje natural común. Trasladar de un lenguaje natural común a una especificación, en donde el requerimiento tenga aplicación en el dominio, es uno de los problemas más comunes que enfrentan los analistas, la relación entre analistas y diseñadores también son susceptibles a producir errores en la interpretación de los requerimientos, razón por la cual se ha ampliado líneas de investigación con el objetivo de acortar brechas entre la especificación de requerimientos y el diseño a través de la conversión de modelos verbales a modelos conceptuales [6], visto de diversos enfoques como heurísticas a ser usadas a discreción del analista o a través de la utilización de herramientas de apoyo tanto para la interpretación automática del lenguaje natural como la aplicación automática de las heurísticas [7].

La mayoría de las herramientas existentes de acuerdo a estudios realizados por [8], se basan en sistemas centralizados de gestión de bases de datos que almacenan información de los requisitos, que suelen consistir en párrafos de texto libre con una serie de atributos predefinidos. Cabe notar que estándares para la especificación de requerimientos de software, como el estándar 830 de la IEEE, son poco utilizados por empresas desarrolladoras de software, siendo este estándar una guía de especificación de requerimientos de software. Mediante este estándar se puede obtener un documento que sirve de contrato sobre especificaciones del

producto a construirse, tanto para la empresa consumidora como para la empresa proveedora, en la que las partes mantengan un mutuo acuerdo. Una de las principales razones por lo cual las empresas no documentan es el tiempo y costos empleados para esta actividad, centrándose directamente en el desarrollo del producto de software.

Nuestro trabajo de investigación tiene como objetivo la construcción de un sistema experto que permita asistir al ingeniero de software en la obtención del documento de especificación de requerimientos de software, basado en el análisis de marcos de problemas propuestos por Michael Jackson [10], mediante el empleo del estándar IEEE 830.

Un Sistema Experto es un conjunto de programas computacionales, que constan de hardware y software que imita el proceso de pensamiento de un experto para resolver problemas complejos en un campo o dominio en particular.

Un marco de problema caracteriza a una clase de problema simple. Los problemas realistas son vistos como composiciones de problemas sencillos de clases reconocidas correspondientes a marcos elementales conocidos, es decir, la complejidad del problema se aborda por descomposición en subproblemas, y así sucesivamente de forma recursiva, hasta que los subproblemas obtenidos son lo suficientemente simples para ser entendidos y resueltos sin más descomposición.

El propósito de este artículo es presentar y motivar el enfoque de marcos de problemas aplicado a la generación del documento de especificación de requerimientos de software mediante un sistema experto. La sección II proporciona un esquema muy general sobre ingeniería de requisitos, análisis de dominios, marcos de problemas y tipificación de requerimientos, explicando e ilustrando sus principios y motivaciones subyacentes. Además se explica la metodología a utilizarse para el desarrollo de un sistema experto. Un análisis más detallado sobre estos principios son aplicados en la construcción de un sistema experto para asistir al ingeniero de conocimientos en la especificación de requerimientos de software en la sección III, en donde se examina los procesos de análisis basados en marcos de problemas y cómo los marcos de problemas encajan en el contexto más amplio de las técnicas de especificación de requerimientos para el desarrollo de productos software utilizando la metodología propuesta en la sección 3. En la sección IV las ideas, principios y prácticas subyacentes son revisados y discutidos en el contexto del sistema desarrollado.

## II. MARCO CONCEPTUAL

### A. Ingeniería de requisitos

La ingeniería de requisitos provee el mecanismo para la comprensión de las necesidades de los usuarios utilizando para ello métodos, técnicas y herramientas que permiten analizar, documentar y especificar requerimientos de software.

El proceso de determinación de requerimientos empieza con la extracción y análisis de requisitos, considerándose una parte especial y crítica en el proceso, ya que es la fase en donde se determina que es lo que realmente los clientes, los usuarios e

interesados quieren que realice el sistema.

Un buen documento de ingeniería de requisitos puede reducir la carga de trabajo de los desarrolladores en las fases posteriores del desarrollo del producto software, mientras que una buena referencia en la documentación ayudará a prevenir problemas de mantenimiento[9].

### B. Análisis del dominio

El análisis del dominio se puede definir como "el proceso por el cual la información utilizada en el desarrollo de sistemas de software dentro del dominio es identificado, capturado, y organizado con el fin de hacerla reutilizable (para crear activos) en la construcción de nuevos productos" [10]. Este proceso puede ser utilizado para identificar elementos comunes y la variabilidad en los requerimientos y capturar las decisiones sobre los rangos y las interdependencias de variabilidad. Si el análisis del dominio no se lleva apropiadamente está termina en su definición ya sea por ser demasiado amplia o por ser demasiado restrictiva al alcance de la línea de productos. Lo que implica que los grandes beneficios como son la reutilización, la reducción de costes y mejora de la calidad del producto software no se pueden realizar [11].

El análisis de dominios se considera como una herramienta útil para el desarrollo de sistemas en dominios estables, en los cuales la información de contexto puede ser fácilmente deducible a partir de expertos de dominio [12].

### C. Marcos de Problemas y Tipificación de requerimientos

Los marcos de problemas, dan mayor énfasis a la comprensión de problemas reales [13], representan, clasifican y transforman los problemas de software a través de la descomposición del problema en subproblemas más simples. Los marcos de problema permiten clasificar y caracterizar clases de problema que ordinariamente ocurren como subproblemas de otros problemas más grandes y reales. Un problema que pertenece a una clase caracterizada por un marco elemental podrá ser definido mediante descripciones apropiadas a un marco. Un problema de una clase compuesta podrá primero ser descompuesto en subproblemas caracterizados por marcos elementales. La descomposición permite detectar e identificar las partes de un marco y las diferentes aproximaciones a la descomposición de un problema real en subproblemas ajustables a marcos elementales. Jackson utiliza dos técnicas para la descomposición de problemas: interior-exterior y exterior-interior. La técnica interior-exterior permite ajustar al problema aproximadamente a un marco de problema conocido, pero exhibe dificultades que frustran la aplicación pura del marco, tales dificultades generan por sí mismas subproblemas que pueden reconocerse ajustándose a otros marcos. De ahí la técnica exterior-interior permite al usuario identificar partes del problema (subproblemas) para deducir un tipo de marco de problema. Una vez identificado el marco de problema por una de las dos técnicas se puede deducir el tipo de requerimiento (tipificación de requerimientos) mismo que será especificado con mayor detalle. La Tabla I muestra los tipos de requerimientos asociados a los tipos de marcos de

problemas descritos por Jackson. Por lo tanto cuando se detecta un problema que se ajusta a uno de los marcos, se conoce cómo debe documentarse sistemáticamente el problema de una manera que sea útil para el resto del

desarrollo. Los problemas reales involucran distintos tipos de requerimientos a la vez, como es el caso de marcos compuestos.

TABLA I  
Tipificación de Requerimientos asociados a los tipos de Marcos de Problemas [13].

Tipo de Requerimiento	Descripción	Marco de Problema	Descripción
Consultas	Requerimiento de información sobre alguna parte del dominio del problema	Información	Todo software que resuelve un problema de información, contesta consultas acerca de cierta parte del mundo real. Documentar un problema de este tipo involucra describir los tipos de requerimientos de información a ser satisfechos, la parte del mundo real sobre la cual aplican tales consultas y cómo el software puede tener acceso a dicha parte del mundo.
Reglas de comportamiento	Reglas que debe seguir el comportamiento del dominio del problema	Control	En un problema de control, el software es responsable de asegurar que alguna parte del mundo se comportará de acuerdo a ciertas reglas. Documentar un problema de este tipo involucra describir los objetos que habitan en tal parte del mundo, las reglas causales a las que obedecen, las reglas que rigen a dichos objetos por efecto de su propia naturaleza, y los fenómenos compartidos con el software a través de los cuales el mismo puede monitorear el estado del mundo e iniciar cadenas causales que resulten en las reglas que se están siguiendo.
Mapeos	Mapeos sobre datos de entrada y de salida del software	Transformación	Para resolver un problema de transformación, el software genera datos de salida que están mapeados contra datos de entrada de acuerdo a reglas específicas. Documentar un problema de este tipo involucra la descripción del conjunto completo de todas las posibles entradas y las reglas de mapeo que prescribe, para cada posible entrada, la salida correcta.
Operaciones sobre dominios creados	Operaciones que realizan los usuarios sobre objetos que existen solo dentro del software	Workpieces	En un problema del tipo Workpieces, el software actúa como una herramienta para carear objetos que existen solo dentro del mismo software. Documentar un problema de este tipo consiste en la descripción del objeto que existirá dentro del software y las operaciones que los usuarios pueden realizar sobre el mismo.
Correspondencias entre dominios	Mantenimiento de dominios que no poseen fenómenos compartidos en sus estados correspondientes.	Conexión	En un problema de conexión, hay dominios que no comparten fenómenos directamente, sino que lo hacen a través de otro dominio entre ellos, el dominio de conexión. El problema consiste en lograr que dos dominios indirectamente conectados se comporten como si estuvieran directamente conectados.

Cabe notar que las relaciones existentes entre los requerimientos, las propiedades del dominio y las especificaciones de la máquina a construirse caracterizan a las clases de problemas. La Figura 1 muestra que un marco de problema consiste en:

- Un dominio del problema que describe el subconjunto del mundo real, el cual es motivo de análisis,
- Un requerimiento que describe lo que se espera sea verdadero acerca de los fenómenos en el dominio del problema, y
- Una máquina que describe el comportamiento que debe ser mostrado por un sistema informático para satisfacer los requerimientos.



Figura 1: Estructura general de un problema de software [10]

D. Metodología para la construcción de un Sistema Experto

De acuerdo a las características del sistema y a la necesidad de garantizar la fiabilidad del sistema experto (SSEE) que se desea construir, resulta adecuado adoptar la Metodología IDEAL para el desarrollo del sistema experto. La Metodología IDEAL propone un ciclo de vida en espiral en tres

dimensiones, y se ajusta a la tendencia del software actual, esto es: ser reutilizable, ser integrable, poseer requisitos abiertos y posee diversidad de modelos computacionales.

Los requisitos están sometidos a constantes cambios, por ende el sistema también, teniendo como resultado un sistema en constante evolución por lo que se considera como un prototipo en constante perfeccionamiento, mediante el agregado de nuevos marcos compuestos, mediante nuevas técnicas de descomposición del problema, mediante nuevas formas de documentación o estándares a los que debe ajustarse.

En la Tabla II se expone las fases y etapas que componen la metodología I.D.E.A.L. y que guiarán el desarrollo del sistema experto:

TABLA II  
Fases de Desarrollo de la Metodología IDEAL [10].

Fase	Descripción
Estudio de Viabilidad	Cuando se intenta resolver un problema con la tecnología de Sistemas Expertos, previamente debe evaluarse si la tarea es abordable en el campo de la Ingeniería del Conocimiento. Es decir, debe dirimirse si el desarrollo es Plausible, Justificable, Adecuada y procura garantizar su Éxito.

Adquisición de Conocimientos	Una de las actividades que requiere mayor esfuerzo, por su complejidad es la extracción y educación de conocimientos, por medio de la cual se intenta descubrir el dominio de la aplicación, el problema y el proceso de solución del mismo.
Conceptualización	Esta actividad está constituida por dos tareas fundamentales: el Análisis, basado en la detección de conocimientos estratégicos, tácticos y fácticos, y la Síntesis donde quedan expresados dichos conocimientos en forma estructurada.
Formalización	Pretende encontrar una adecuada representación de los conocimientos, garantizando su correcta manipulación.
Implementación	Desarrolla la transformación de los conocimientos representados en el modelo formal en un modelo computacional
Evaluación	Establece el grado de experiencia alcanzado por el sistema. Al evaluar el desempeño del sistema, tratando de vislumbrar la calidad de asistencia que brinda el sistema experto ante diferentes casos de problema a resolver por software..

Sin embargo no todos los problemas pueden ser resueltos mediante la tecnología de sistemas expertos, solo aquellos que se justifiquen y sean apropiados para un fin. Para que un sistema experto sea viable debe existir un experto en el dominio, caso contrario el sistema fracasará. La viabilidad del sistema se plantea de acuerdo a la metodología IDEAL evalúa 4 dimensiones: plausibilidad, adecuación, justificación y éxito que caracterizan al problema. Para ello se utiliza el test de viabilidad, que consiste en asignar valores a las dimensiones anteriormente indicadas. El método es de tipo métrico, usa ponderaciones, utiliza la media armónica e incorpora la manipulación de valores lingüísticos mediante intervalos difusos, con los que, además, se pueden definir operaciones básicas de cálculos para cada dimensión. La Tabla III muestra el análisis final de las dimensiones la cual consiste en el cálculo de la media aritmética ponderada del resultado obtenido para cada dimensión, para más detalle revisar referencia [18].

TABLA III  
Análisis de las dimensiones plausibilidad, adecuación, justificación y éxito para validez el SSEE [14].

Dimensión	Peso	Valores Intervalo				Peso*Valor			
Plausibilidad	8	7,9	8,3	8,7	8,9	63	66	70	71
Justificación	3	7,8	8,8	10	10	23	26	30	30
Adecuación	8	5,2	6,4	7,6	8,46	41	51	60	68
Éxito	5	2,4	2,8	3,3	3,64	12	14	16	18
		24				140	158	177	187
		Intervalo Resultado Final				6	7	7	8
		Resultado Final:				6,90			

Un proyecto es viable si el cálculo de la media aritmética ponderada del análisis de dimensiones del sistema es mayor a 6. Una vez calculado estos valores, como puede observarse en la Tabla III, obteniéndose el valor de 6.9 lo cual implica que es viable el desarrollo del sistema experto, desde la perspectiva de la Ingeniería del Conocimiento [14].

### III. DESARROLLO DE UN FRAMEWORK PARA LA ESPECIFICACIÓN DE REQUERIMIENTOS MEDIANTE UN SISTEMA EXPERTO

Para el desarrollo del sistema experto es importante entender las características y atributos que tendrán influencia para su construcción, ver Tabla III. A través de la comprensión de estos criterios podemos definir requerimientos más claros para el desarrollo. Además, se analiza el proceso ingeniería de requisitos identificando los objetivos y requisitos. Para luego estudiar las técnicas específicas que se pueden aplicar en el desarrollo del sistema experto justificando las técnicas seleccionadas.

TABLA IV  
Características y Atributos que influyen en el desarrollo de un SSEE

Factores	Descripción
Tamaño de Proyecto	Diferentes tamaños de proyectos necesitan diferentes técnicas de requerimientos
Volatilidad de Requerimientos	Proyectos con requerimientos de alta volatilidad necesitan técnicas más flexibles de ingeniería de requerimientos.
Categoría del Proyecto	Se necesitan diferentes técnicas para diferentes categorías de proyectos.
Seguridad Critica	La aplicación de técnicas más serias son necesarias para sistemas con seguridad crítica
Complejidad del Proyecto	Los proyectos complejos necesitan diferentes técnicas de ingeniería requerimientos
Restricciones de Costo del Proyecto	Los ingenieros del conocimiento tendrán dificultades para completar los proyectos si el presupuesto no es suficiente
Restricciones de Tiempo del Proyecto	El sistema experto no podría satisfacer los requerimientos si el tiempo es limitado.

Para el desarrollo del sistema experto hemos delimitado sus características y atributos en referencia a su tamaño y categoría para proyectos pequeños, la volatilidad de los requerimientos se considera estática. Estas restricciones se deben al mercado de desarrollo de productos de software cubierto por esta investigación y por los casos propuestos y evaluados por el sistema experto, ver sección IV.

La tabla V muestra las actividades desarrolladas por el ingeniero de conocimiento para la elicitación de requerimientos del sistema experto.

TABLA V  
Fases y Actividades de la Ingeniería de Requisitos utilizadas para la construcción del SEE

Fases	Actividades
	Se identifican todos los actores del sistema. Usuarios, Clientes y otras partes Interesadas.
Obtención de requisitos	Se realizan entrevista, encuestas, grupos de enfoque y de intercambio de ideas. Se identifican todos los requisitos para alcanzar el objetivo del sistema experto.
Modelado	Se organizan los datos de manera que puedan ser entendidos por todos los grupos de interés. Se crear el modelo conceptual a través del modelo Entidad-Relación. El modelo pueden ser utilizados para ser referenciados por el equipo de desarrollo y por cada equipo de desarrollo deben tener acceso de la misma. Se crea un resumen de la actividad que es susceptible a interpretación.

Análisis de requerimientos	<p>Se identifica, entiende y comprenden las funciones, los datos y las interfaces y sus relaciones en el desarrollo del sistema.</p> <p>Se identifican ambigüedades, inconsistencias e incompletitud para ser resueltos.</p> <p>Se analizan los requisitos no funcionales del sistema experto.</p> <p>Se gestiona conflictos, la tarjeta de clasificación y análisis basado en un punto de vista.</p>
Validación y Verificación	<p>Se garantiza que los requerimientos del sistema experto son válidos de manera que sean ejecutables para las siguientes fases de desarrollo.</p> <p>Se aclara los requisitos ambiguos.</p> <p>Se realiza una inspección formal, se chequea la lista de verificación de requisitos, se realiza las pruebas de requisitos en el prototipo desarrollado.</p>
Gestión de Requisitos	<p>Incluye las tareas de gestión de los requisitos en todo el tiempo de desarrollo</p> <p>Se documentan los requisitos.</p>

La Figura 2 muestra la base para la construcción del sistema experto para asistir al ingeniero de conocimiento en la especificación de requerimientos. Describe cómo cualquier metodología de elicitación de requerimientos puede ser representada en términos de un modelo. El modelo basado en [16] representa una generalización de la metodología utilizada, las técnicas de elicitación conocidas y la asociación de marcos de problemas en el proceso de elicitación de requerimientos. Explícitamente se destaca el rol que juega el conocimiento en el desempeño de la obtención y selección de técnicas de elicitación. Analizándose para ello el dominio del problema y el dominio del proyecto. Lo que permite proporcionar un marco unificado con el fin de entender el propósito y la función de la obtención de requerimientos en el desarrollo de software. Además este modelo permite describir como un ingeniero en conocimiento puede realizar la elicitación de requerimientos moviéndose a través de una serie de actividades. El propósito de cada actividad es acercar a las partes más y más a un entendimiento común de los requerimientos que se quieren tratar. De acuerdo al modelo propuesto existen tres formas de obtener requerimientos de software:

1. Una vez analizado el problema este dará guías para describir tanto el dominio del problema, como el dominio del proyecto, según la descomposición del problema se seleccionará la técnica de elicitación a aplicarse de acuerdo a requerimientos conocidos, aplicada la técnica y analizado el dominio de conocimiento se traspasa la información al sistema obteniéndose requerimientos funcionales.
2. Una vez analizado el problema y de acuerdo a los conocimientos obtenidos de los usuarios, clientes y/o partes interesadas se traspasa la información al sistema obteniéndose requerimientos funcionales y no funcionales.
3. De acuerdo al análisis del problema y la descomposición del problema por parte de los usuarios, el sistema ajusta esta información a un marco de problema, para luego tipificarlos en requerimientos funcionales sugeridos por el sistema experto.

El objetivo primario es obtener un marco de problema adecuado al problema que el sistema intenta dar solución y, a partir del mismo, servir de guía para que toda la información necesaria haya sido obtenida con el fin de generar un documento preliminar de requerimientos.

La descomposición utilizada en esta investigación es la exterior-interior la cual empieza por encontrar aspectos del problema conocidos y/o reconocidos por los usuarios, las cuales se encaminan a la deducción de un tipo de marco de problema. La parte fundamental para el sistema es: saber hacer buenas preguntas y determinar las condiciones necesarias para encontrar lo desconocido del problema, es decir, generar y deducir correctamente un tipo de marco de problema, en el cual la especificación del requerimiento tenga sentido en el contexto del dominio tratado, sin olvidar las partes relevantes de los requerimientos funcionales, los cuales definen el comportamiento específico del software. En lo que se refiere a los requerimientos no funcionales o atributos de calidad, definen las características del sistema como la fiabilidad, la respuesta en el tiempo, la capacidad de almacenamiento, seguridad. Estos adquieren importancia en el momento de modelar la arquitectura del sistema. Siendo parte integrante del documento de especificación de requerimientos.

Cabe notar que existe una gran cantidad de técnicas en las diferentes actividades del proceso de ingeniería de requisitos que se puede utilizar para el desarrollo del sistema. En este trabajo, hemos identificado todas las técnicas específicas de la biblioteca técnica de ingeniería de requisitos propuesto en la investigación realizada por Jiang et al. [2]. Mientras tanto, los factores de selección se basan en el factor enumerado por Kheirkhah y Dareman [1]. La Tabla V [15] presenta el tipo de requerimiento cubierto a través de la técnica utilizada en cada una de las actividades de la ingeniería de requisitos.

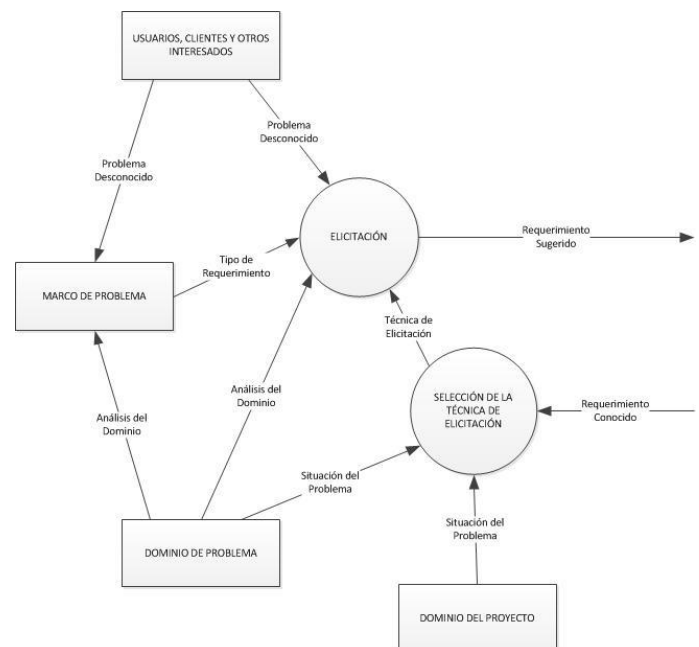


Figura 2: Modelo de elicitación de requerimientos del SSEE.

TABLA V  
Proceso de Ingeniería de Requisitos para la Educación de Requerimientos

Proceso	Técnica	Descripción de la Técnica	Requerimiento cubierto
Elicitación de Requerimientos	Reunión	Reunión de usuarios e interesados para discutir requerimientos	Funcional
	Cuestionario	Lista escrita de preguntas para ser respondidas por usuarios para proporcionar información básica sobre el producto de software	Funcional
	Lluvia de Ideas	Ideas brillantes repentinas	No Funcional
	Entrevistas	Reunión formal con diferentes grupos de interés por separado con el fin de obtener diferentes opiniones	Funcional y No Funcional
Modelamiento de Requerimientos	Diagrama Entidad-Relación	Descripción de la Base de Datos de una forma abstracta	Funcional
Análisis de Requerimientos	Manejo de Conflictos	Resolución de conflictos en el requerimiento de los interesados	Funcional
	Puntos de Vista	Identificación de puntos de vista de las partes interesadas	Funcional y No Funcional
Evaluación de Requerimientos	Inspección Formal	Identificación de requerimientos correctos y consistentes	Funcional y No Funcional
	Prueba de Requerimientos	Se evalúa una serie de casos destinados a averiguar la capacidad del producto	Funcional y No Funcional
Administración de Requerimientos	Gestión de Requerimientos	Proceso de identificación de la ambigüedad, inconsistencia y lo incompleto del sistema	Funcional y No Funcional

La Tabla VI describe el proceso de educación de requerimientos realizado por el sistema a través de marcos de problemas. Para ello se han formulado preguntas específicas sobre subproblemas identificados para los usuarios, que permitan inferir marcos de problemas conocidos para luego poder tipificarlos en requerimientos funcionales.

TABLA VI  
Descripción de Requerimientos Funcionales inferidos mediante Marcos de Problemas por el SSEE.

ID	Entrada/Pregunta	Descripción	Posibles Respuestas	Marco de Problema Inferido	Tipo de Requerimiento Inferido	Detalle de Especificación del Requerimiento
ER1	¿Existe Información requerida por los solicitantes de información (Usuario, Hardware, Software)?	Permite determinar si el sistema genera algún tipo de consulta solicitada por los usuarios	Si No	Marco de Información	Requerimiento de Consulta	Reporte detallado de consultas de la Base de Datos. La BDD donde se almacena la información debe estar indexada por campos claves para que la búsqueda tenga una respuesta óptima. Se debe colocar una clave principal dentro de cada tabla creada en la Base de Datos para mejorar el performance y la utilidad de las consultas.
ER2	¿Existe información entregada por el mundo real a través de formularios, pre impreso u otros?	Permite determinar si el sistema entrega información a través de formularios, impresos entre otros.	Si No			Reporte detallado de consultas, en varios formatos de salida (pdf, doc, xls, etc.), filtros de búsqueda de información deben tener varios criterios como rangos de fechas, claves principales o palabras claves.
ER3	¿Existen informaciones de cálculo que generen nuevos tipos de datos?	Permite determinar si el usuario realiza ciertos cálculos en un conjunto de datos de entrada que produzcan transformaciones a un conjunto de datos de salida.	Si No	Marco de Transformación	Requerimiento de Mapeo	Revisar los datos de entrada con los cuales se realizan cálculos generando nuevos tipos de datos en las salidas.
ER4	¿Existe operaciones de transformación que genere nuevos formatos de archivos?	Permite determinar si el sistema genera nuevos tipos de archivos.	Si No			Generar nuevos formatos de archivos de acuerdo con las necesidades de los clientes.
ER5	¿Existen dispositivos (Reloj Biométrico, cajero automático, dispensador de moneda o similares)?	Permite determinar si el sistema controla un dispositivo haciendo que cumpla una serie de reglas de comportamiento.	Si No	Marco de Control	Requerimiento de reglas de comportamiento	Utilizar los drives de Instalación que viene por defecto dentro del equipo que vamos a controlar dentro de nuestro software, procurar que la información que ingrese desde el dispositivo externo lleguen en archivos planos para que su

						manipulación sea transparente hacia nuestro software contar con documentación sobre la maquinaria que vamos a controlar para conocer su interfaz de comunicación y el manejo de su funcionalidad
--	--	--	--	--	--	--

Las siglas ER significa especificación de requerimientos.

Para el caso de los requerimientos no funcionales este se diseña las mismas que se detallan en la Tabla VII. basa en una serie de preguntas puntuales sobre restricciones de

**TABLA VII**  
Entradas para la Educación de Requerimientos no Funcionales: Restricciones de Diseño

ID	Entrada/Pregunta	Descripción	Opciones
ER7	La política de Adquisición de Software para toda la empresa es	Permite seleccionar el tipo de Lenguaje con el que la empresa cuenta o sugiere	<ul style="list-style-type: none"> <li>• Software Libre</li> <li>• Ambos</li> </ul>
ER8	El sistema necesita tener varios lenguajes de programación	Permite determinar si el sistema funcionará con múltiples lenguajes de programación	<ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul>
ER9	Alcance de acceso de su sistema	Permite determinar el acceso de comunicación del sistema.	<ul style="list-style-type: none"> <li>• Utiliza solo dentro de la empresa</li> <li>• Utiliza desde Internet con computador de escritorio</li> <li>• Utiliza desde Internet utilizando dispositivos como tablets, celulares etc.</li> </ul>
ER10	El Sistema necesita tener bases de datos	Permite que el sistema almacene la información permanentemente	<ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul>
ER11	El sistema necesita compartir información con otros sistemas	Permite determinar si existe interconexión con sistemas externos	<ul style="list-style-type: none"> <li>• Si</li> <li>• No</li> </ul>
ER11.1	Como gestiona la Información de entrada y salida	Si existe interconexión con sistemas externos, permite gestionar la información de entrada salida	<ul style="list-style-type: none"> <li>• Servicios Web</li> <li>• Archivos planos</li> <li>• Tramas</li> <li>• Clonando estructuras de Base De Datos</li> <li>• Consultas</li> </ul>
ER12	En cantidad de horas al día, que tiempo deberá estar la aplicación disponible	Permite determinar la disponibilidad del sistema	<ul style="list-style-type: none"> <li>• 1-8 horas</li> <li>• 8-más horas</li> </ul>
ER13	En una escala de 1 a 4 siendo 1 nada sensible y 4 muy sensible en que escala colocaría la información que manejará el sistema	Permite determinar el nivel de seguridad del sistema	<ul style="list-style-type: none"> <li>• 1</li> <li>• 2</li> <li>• 3</li> <li>• 4</li> </ul>
ER14	Coloque un estimado en GB de la información que manejará el sistema en un año de funcionamiento	Permite determinar la capacidad de los servidores para almacenar información.	<ul style="list-style-type: none"> <li>• 1-5 GB</li> <li>• 5-10 GB</li> <li>• 10-más GB</li> </ul>

Las siglas ER significa especificación de requerimientos.

El sistema experto se encuentra embebido en un sistema de gestión. Este sistema permite entender la conceptualización del dominio del problema, detallándose entidades, atributos y relaciones del sistema. Una vez instanciado el sistema experto permite generar el documento de especificación de requerimientos de software. Además el sistema de gestión administra dos tipos de usuarios: los usuarios que gestionan los proyectos y los administradores que a más de gestionar proyectos dan mantenimiento al sistema.

El sistema experto se genero en la herramienta Clips (C Language Integrated Production System) por cuanto presenta características adaptables al trabajo realizado de acuerdo a las siguientes consideraciones: permite la representación del conocimiento humano a través de reglas heurísticas, dispone de un intérprete que permite escribir comandos de manera interactiva, permite mostrar información sobre el estado actual de la memoria de trabajo y las reglas que las va activando en cada momento, el motor de inferencias es un algoritmo de encadenamiento hacia delante, en el que a partir de los hechos se obtiene los resultados, implementa un algoritmo

denominado Rete, que mantiene en memoria condiciones y acciones que hace más eficiente la comprobación de qué reglas pueden ser activadas en cada momento, es una herramienta de libre distribución que puede interactuar con otras herramientas de desarrollo.

#### IV. EVALUACIÓN DEL SISTEMA EXPERTO

La evaluación de un sistema consiste en comprobar que la aplicación funcione correctamente y responda a las necesidades del usuario. Por lo que resulta una de las partes fundamentales en la construcción de un sistema experto, en nuestro caso, para asistir al ingeniero en conocimiento en la especificación de requerimientos de software. En este proceso el experto debe asegurar que los requerimientos sean ejecutables para el desarrollo del producto software deseado. Cabe notar que un experto involucra directamente a usuarios y/o personas interesadas para asegurar que los requerimientos establecidos por ellos sean validados y verificados mediante la visualización de los artefactos de requerimientos. Por ejemplo podemos requerir que la parte interesada despeje interrogantes

y ambigüedades en el proceso de verificación. Para ello la evaluación se apoya en algunas técnicas como son: la inspección formal, lista de verificación de requerimientos, pruebas de requerimientos y pruebas del prototipo. En nuestro caso particular nos apoyamos por lo general en la técnica de inspección formal validada por el experto del dominio.

De acuerdo con la metodología utilizada (Metodología Ideal) para la construcción del Sistema Experto hemos realizado evaluaciones en cada una de las fases de desarrollo: Adquisición de conocimientos, Conceptualización de Conocimientos, Formalización de conocimientos e Implementación de conocimientos.

#### A. Evaluación de la Adquisición del Conocimiento.

La fase de adquisición de conocimientos permite educir los conocimientos del experto acerca del dominio de la aplicación. Para obtener dichos conocimientos se realizaron sesiones de trabajo entre el experto del dominio y el ingeniero del conocimiento a través de reuniones, entrevistas, preguntas, intercambio de ideas y/o análisis de textos. En esta evaluación, se utilizó dos técnicas de validación y verificación de Sistemas Expertos: las inspecciones formales y pruebas de requerimientos, realizándose una retroalimentación de conceptos, definiciones, casos de estudio, validándose y verificándose todos los requisitos con la finalidad de lograr el objetivo del sistema experto, lo que permitió obtener un documento comprensible, consistente, completo y no redundante, de acuerdo con los lineamientos del experto y del ingeniero en conocimientos.

#### B. Evaluación de la Conceptualización del Conocimiento.

Para esta evaluación hemos utilizado cinco tipos de comprobaciones:

- Comprobación de los conocimientos estratégicos para lo cual se evaluó el modelo de descomposición funcional mediante inspecciones formales a través de revisiones parciales por parte del experto.
- Comprobación de los conocimientos tácticos, se revisaron en detalle las seudoreglas, mediante inspecciones formales a través de revisiones parciales aceptadas por el experto.
- Comprobación de los conocimientos fácticos, se revisaron glosario de términos, diccionario de conceptos, tabla de concepto-atributo-valor y definición de atributos mediante revisiones parciales avaladas por el experto e ingeniero del conocimiento.
- Comprobación del modelo dinámico y estático, fueron revisados y aceptados por el experto.
- Comprobación de la fase de conceptualización, para realizar la evaluación de esta fase se seleccionó un caso de prueba, misma que se referencia en la evaluación de la implementación de conocimientos.

#### C. Evaluación de la Formalización del Conocimiento.

En ésta evaluación se realizó la formalización de

conocimientos con respecto a la conceptualización, seleccionándose el mismo caso de prueba anterior. Esta evaluación se realizó teniendo en cuenta dos criterios, según [17]:

**Marcos:** Se ha revisado la coincidencia con el modelo conceptual así como también se ha analizado que no exista redundancias, incompletitud e inconsistencias de datos e información.

**Reglas:** Se ha revisado y analizado que no existan reglas duplicadas, reglas idénticas, reglas embebidas, condiciones si innecesarias, callejones sin salida, reglas ausentes, reglas inalcanzable inconsistentes, incompatibles y reglas circulares.

#### D. Evaluación de la Implementación del Conocimiento.

Para realizar esta evaluación se han comparado los resultados obtenidos por el sistema con respecto a los documentos generados por el ingeniero de conocimiento. En este artículo evaluamos cuatro casos de prueba en donde se puede encontrar resultados particulares de requerimientos funcionales y no funcionales. Los casos de prueba se describen en la Tabla VIII. Cabe notar que en este artículo solo se realiza la evaluación para el primer caso de estudio: Sistema de Facturación Electrónico.

TABLA VIII  
Casos de prueba para evaluar el Sistema Experto

CASO	PROYECTO	DESCRIPCIÓN
1	Facturación Electrónica (SFE)	Sistema de Información que permite cubrir actividades de gestión de Facturación Electrónica.
2	Sistema de Información de Gestión Académica (SIGEAC)	Sistema de Información que permite cubrir actividades de gestión de Facultades e Institutos de una Institución Educativa.
3	Oci-Tienda para RedSpain	Oci-Tienda es una tienda online para la venta a distancia de diversos productos. Se solicita realizar un sistema de información que permita cubrir actividades de gestión de empresas, gestión de productos o servicios mediante las funciones de compra, venta, contrato y consulta de productos, a demás deberá permitir el pago a través de tarjetas de crédito.
4	Sistema de Información de PCGeek	Sistema de Información que permite cubrir actividades de gestión del almacenaje, compras y construcción de PCs en la empresa PCGeek.
5	Sistema Experto para asistir al IC en la especificación de requerimientos.(SE AICER)	Sistema de Información que permite cubrir actividades de gestión de proyectos y la deducción de la especificación de requerimientos funcionales y no funcionales.

Para evaluar el sistema nos hemos basado en las Tablas IX y X en donde se analiza los tipos de requerimientos funcionales y en las Tablas XI y XII en donde se analiza los requerimientos no funcionales encontrados y/o deducidos por el sistema experto para el primer caso de prueba.

La Tabla IX muestra las preguntas que realiza el sistema experto a los usuarios para educir el/los marcos de problema y así poder tipificar los requerimientos funcionales, mediante las inferencias a las reglas, las cuales se presentan en la Tabla X.



TABLA IX  
Entradas para la Educación de Requerimientos Funcionales

ID	Entrada/Pregunta	Descripción	Opciones
ER1	¿Existe Información requerida por los solicitantes de información (Usuario, Hardware, Software)	Permite determinar si el sistema genera algún tipo de consulta solicitada por los usuarios	Si
ER2	¿Existe información entregada por el mundo real a través de formularios, pre impreso u otros?	Permite determinar si el sistema entrega información a través de formularios, impresos entre otros.	Si
ER3	¿Existen informaciones de cálculo que generen nuevos tipos de datos?	Permite determinar si el usuario realiza ciertos cálculos en un conjunto de datos de entrada que produzcan transformaciones a un conjunto de datos de salida.	Si
ER4	¿Existe operaciones de transformación que genere nuevos formatos de archivos?	Permite determinar si el sistema genera nuevos tipos de archivos.	Si
ER5	¿Existen dispositivos (Reloj Biométrico, cajero automático, dispensador de moneda o similares?)	Permite determinar si el sistema controla un dispositivo haciendo que cumpla una serie de reglas de comportamiento.	No
ER6	¿Los usuarios requieren herramientas para crear y editar artefactos intangibles tales como textos o gráficos?	Permite determinar si el sistema requiere de una herramienta para crear y editar artefactos intangibles tales como texto, gráficos o estructuras similares para que puedan ser posteriormente copiadas, impresas, analizadas o usadas de otra manera	No

Las siglas ER significa especificación de requerimientos.

TABLA X

Reglas de Inferencia para la Educación de Requerimientos Funcionales a través de Marcos de Problemas

<pre> Regla #21 (defrule Req-func-Consultas   (declare (salience -1))   (Es SolicitantesInformacion ?ent1)   (Es InformacionEntregada ?ent2)   ?ret1 &lt;- (Es SolicitantesInformacion ?ent1)   ?ret2 &lt;- (Es InformacionEntregada ?ent2) =&gt;   (open "outReqnoFunc.txt" outdata "a")   (if (eq ?ent1 Si) then     (if (eq ?ent2 Si) then       (assert (Es UnRequerimiento Consulta))       (printout outdata "Requerimiento de Consulta" crlf)       (printout outdata "Marco de Problema de Información " crlf)       (retract ?ret1)       (retract ?ret2)))     )   (close outdata) ) ;Regla #23 (defrule Req-func-Mapeo   (declare (salience -1))   (Es OperacionesTransformacion ?ent1)   (Es OperacionesdeCalculo ?ent2)   ?ret1 &lt;- (Es OperacionesTransformacion ?ent1) </pre>
--

<pre> ?ret2 &lt;- (Es OperacionesdeCalculo ?ent2) =&gt;   (open "outReqnoFunc.txt" outdata "a")   (if (eq ?ent1 Si) then     (if (eq ?ent2 Si) then       (assert (Es UnRequerimiento Mapeo))       (printout outdata "Requerimiento de Mapeo" crlf)       (printout outdata "Marco de Problema de Transformación" crlf)       (retract ?ret1)       (retract ?ret2)))     )   (close outdata) ) ;Regla #25 (defrule Req-func-MarcoProblema   (declare (salience -10))   (Es UnRequerimiento ?ent1)   ?ret1 &lt;- (Es UnRequerimiento ?ent1) =&gt;   (open "outReqnoFunc.txt" outdata "a")   (if (eq ?ent1 Consulta) then     (assert (Es UnMarcoProblema Informacion))     (printout outdata "reporte detallado de consultas, en varios formatos de salida (pdf, doc, xls, etc.), filtros de búsqueda de información deben tener varios criterios como rangos de fechas, claves principales o palabras claves .La Base de Datos donde se almacena la información debe estar indexada por campos claves para que la búsqueda tenga una respuesta óptima. Colocar una clave principal dentro de cada tabla creada en la Base de Datos para mejora el performance y la utilidad de las consultas." crlf))   )   (if (eq ?ent1 Mapeo) then     (assert (Es UnMarcoProblema Transformacion))     (printout outdata "Revisar los datos de entrada con los cuales se realizan cálculos generando nuevos tipos de datos en las salidas o nuevos formatos de archivos de acuerdo con las necesidades de los clientes" crlf))   )   (if (eq ?ent1 OperacionesDominios) then     (assert (EsUn MarcoProblema WorkPiece))     (printout outdata "Utilizar librerías nativas del lenguaje de programación, procesar las imágenes y documentos en el servidor de aplicaciones y no almacenarlas directamente en la Base de Datos. No realizar el procesamiento de imágenes dentro de la capa de usuario sino en la capa de negocios para que no se afecte a la performance de la aplicación. Los documentos generados pueden ser impresos en varios formatos para facilidad del cliente final." crlf))     (retract ?ret1)     (close outdata)   ) ) </pre>
--

De acuerdo a las inferencias realizadas por el sistema, basadas en las Tablas IX y X, podemos indicar que el dominio de la aplicación se ajusta a dos marcos de problemas: de información y de transformación, lo que permite tipificar los requerimientos de consulta y mapeo.

El sistema debe tomar en cuenta los siguientes detalles para la especificación de requerimientos:

- Reporte detallado de consultas de la Base de Datos. La BDD donde se almacena la información debe estar indexada por campos claves para que la búsqueda tenga una respuesta óptima. Se debe colocar una clave principal dentro de cada tabla creada en la Base de Datos para mejorar el performance y la utilidad de las consultas.
- Reporte detallado de consultas, en varios formatos de salida (pdf, doc, xls, etc.), filtros de búsqueda de información deben tener varios criterios como rangos de fechas, claves principales o palabras claves.
- Revisar los datos de entrada con los cuales se realizan cálculos generando nuevos tipos de datos en las salidas.
- Generar nuevos formatos de archivos de acuerdo con las

necesidades de los clientes.

De los resultados obtenidos podemos indicar que el sistema experto infiere y deduce satisfactoriamente los requerimientos funcionales mismos que fueron abalizados por el ingeniero en conocimiento y el experto.

TABLA IX  
Preguntas para la Educación de Requerimientos no funcionales

ID	Entrada/Pregunta	Descripción	Opciones
ER7	La política de Adquisición de Software para toda la empresa es?	Permite seleccionar el tipo de lenguaje con el que la empresa cuenta o sugiere	Software Libre
ER8	El sistema necesita tener varios lenguajes de programación?	Permite determinar si el sistema funcionará con múltiples lenguajes de programación	Si
ER9	Alcance de acceso de su sistema?	Permite determinar el acceso de comunicación del sistema.	Utiliza desde Internet con computadoras de escritorio
ER10	El Sistema necesita tener bases de datos?	Permite que el sistema almacene la información permanentemente	Si
ER11	El sistema necesita compartir información con otros sistemas?	Permite determinar si existe interconexión con sistemas externos	Si
ER11.1	Como gestiona la Información de entrada y salida?	Si existe interconexión con sistemas externos, permite gestionar la información de entrada salida	Servicios Web
ER12	En cantidad de horas al día, que tiempo deberá estar la aplicación disponible?	Permite determinar la disponibilidad del sistema	Mayor a 8
ER13	En una escala de 1 a 4 siendo 1 nada sensible y 4 muy sensible en que escala colocaría la información que manejará el sistema?	Permite determinar el nivel de seguridad del sistema	3
ER14	Coloque un estimado en GB de la información que manejará el sistema en un año de funcionamiento?	Permite determinar la capacidad de los servidores para almacenar información.	De 5- 10

Las siglas ER significa especificación de requerimientos.

TABLA XI  
Reglas de Inferencia para la Educación de Requerimientos no Funcionales

:Delimitación y restricciones: Herramientas de Desarrollo :Regla #2 (defrule Req-no-func-Lenguaje-BDD-Internet (Es Herramienta ?ent1) (Es Computadoras ?ent2) (Es BaseDatos ?ent3) (Es Internet ?ent4) =>
--

<pre> (open "outReqnoFunc.txt" outdata "a") (if (eq ?ent1 SoftwareLibre) then   (if (eq ?ent2 Escritorio) then     (if (eq ?ent3 Si) then       (if (eq ?ent4 Si) then         (printout outdata "PHP con Herramientas Web y MySQL" crlf)))))) (close outdata) ) :Regla #12 (defrule Req-no-func-Multiples-Lenguajes   (Es MultiplesLenguajes ?ent1)  =&gt;   (open "outReqnoFunc.txt" outdata "a")   (if (eq ?ent1 Si) then     (printout outdata "Archivos de Globalización" crlf))   (close outdata) )  :Delimitación y restricciones: Conexión con Sistemas Externos, Servicios Web :Regla #13 (defrule Req-no-func-Conexion-Externa-Servicio-Web   (Es ConexionSistemaExterno ?ent1)   (Es ServiciosWeb ?ent2)  =&gt;   (open "outReqnoFunc.txt" outdata "a")   (if (eq ?ent1 Si) then     (if (eq ?ent2 Si) then       (printout outdata "Enviar requisitos como dirección del servicio Web, Archivos WDSL" crlf)))   (close outdata) )  :Regla #18 (defrule Req-no-func-Seguridad   (Es Sensibilidad ?ent1)  =&gt;   (open "outReqnoFunc.txt" outdata "a")   (if (eq ?ent1 4) then     (printout outdata "Sensibilidad Alta: Manejar autenticación de usuarios, roles, perfiles, expiración de claves y reglas mínimas de clave, métodos para el control de auditoría y encriptación del canal de envío y BDD" crlf))     (if (eq ?ent1 3) then       (printout outdata "Sensibilidad Media: Manejar autenticación de usuarios, roles, perfiles, expiración de claves y reglas mínimas de clave, métodos para el control de auditoría y la comunicación se hará encriptando el canal de envío" crlf))     (if (eq ?ent1 2) then       (printout outdata "Sensibilidad Media: Manejar autenticación de usuarios                 </pre>
--

```

sin autorias, ni encriptación en la comunicación" crlf))
  (if (eq ?ent1 1) then
    (printout outdata "Sensibilidad Baja: Sin autenticación de usuarios, ni
auditorías de acceso y manipulación de datos" crlf))
  (close outdata)
)

:Regla #19
(defrule Req-no-func-Disponibilidad
  (Es Disponibilidad ?ent1)
=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (> ?ent1 8) then
    (printout outdata "Disponibilidad Alta: Habilitar Servidores Replica, para
evitar que la caída del Sistema afecte la disponibilidad" crlf))
  (if (< ?ent1 8) then
    (printout outdata "Disponibilidad Media: Utilizar un servidor robusto
para el alojamiento de la aplicación y programar puntos de restauración y
respaldo" crlf))
  (close outdata)
)

:Regla #20
(defrule Req-no-func-Capacidad
  (Es Capacidad ?ent1)
=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (or (> ?ent1 1) (< ?ent1 5)) then
    (printout outdata "Capacidad Baja: Su Software necesita un servidor
moderadamente robusto con 100GB de almacenamiento y 1 procesador
sin la necesidad de virtualización y sistemas de respaldo" crlf))
  (if (or (> ?ent1 5) (< ?ent1 10)) then
    (printout outdata "Capacidad Media: Su Software necesita un servidor
robusto de 500GB de almacenamiento y 2 procesadores sin la
necesidad de virtualización y sistemas de respaldo" crlf))
  (if (> ?ent1 10) then
    (printout outdata "Capacidad Alta: Su Software necesita un servidor
robusto con 2TB de almacenamiento y 4 procesadores con virtualización y
sistemas de respaldo" crlf))
  (close outdata)
)

```

De acuerdo a las inferencias realizadas por el sistema, basadas en las Tablas XI y XII, podemos indicar que el sistema experto infiere y deduce satisfactoriamente los requerimientos no funcionales de acuerdo a los resultados obtenidos y abalizados por el ingeniero en conocimiento y el experto, obteniéndose los siguientes requerimientos no funcionales:

- Capacidad Media: Su Software necesita un servidor robusto de 500GB de almacenamiento y 2 procesadores

sin la necesidad de virtualización y sistemas de respaldo.

- Sensibilidad Media: Manejar autenticación de usuarios, roles, perfiles, expiración de claves y reglas mínimas de clave. Se debe aplicar métodos para el control de auditoría. Además la comunicación se hará encriptando el canal de envío para mantener la seguridad de la información.
- Disponibilidad Alta: Habilitar Servidores Réplica, para evitar que la caída del Sistema afecte la disponibilidad
- Desarrollaremos un servicio Web para la comunicación y recibir los datos de acuerdo a los parámetros de entrada del servicio Web que se va a consumir. Requisitos: Dirección para el consumo del servicio Web, el archivo WSDL para el consumo y parámetros de entrada y salida. Enviar requisitos como dirección del servicio Web, Archivos WDSL.
- PHP con Herramientas Web y MySql
- Archivos de Globalización

Una vez realizada la evaluación de requerimientos funcionales y no funcionales podemos concluir que el sistema experto para asistir al ingeniero en software en la elaboración de documentos de requerimientos es funcional, de acuerdo al criterio y aval del experto del dominio.

Cabe notar que una vez obtenidos los requerimientos del proyecto, sean estos funcionales o no funcionales, el sistema genera automáticamente un documento de especificación de requerimientos de acuerdo al estándar 830 de la IEEE, el mismo que servirá como un artefacto inicial para el desarrollo de ciclo de vida del proyecto.

Con respecto a los otros casos de estudio la evaluación sirvió para validar y verificar las diferentes posibilidades de tipificación de requerimientos ya sea mediante marcos de problemas (requerimientos funcionales) o atributos de calidad (requerimientos no funcionales). Lo que permitió verificar el alcance del sistema experto. Estas evaluaciones fueron también abalizadas por el experto del dominio.

## V. DISCUSIÓN DE RESULTADOS

Una especificación es una herramienta fundamental para garantizar la calidad del proceso de desarrollo y del producto final entregado. La fiabilidad de los sistemas expertos ha sido un tema muy debatido, y la falta de consenso sobre el tema puede ser la razón por la cual los sistemas expertos han permanecido en gran medida un producto de la investigación y no un producto comercial.

Uno de los principales problemas en la construcción de un producto de software es entender y analizar las necesidades del usuario. Para poder especificar los requerimientos de un sistema es necesario hacer una aproximación al análisis del problema, en ésta artículo se planteó realizar éste análisis mediante la aplicación de Marcos de Problemas, sugeridos por Michael Jackson que es una técnica que permite mantener la integridad de los datos, a través del análisis de dominios y la tipificación de requerimientos.

En éste trabajo se desarrolló un sistema experto que de soporte al ingeniero de conocimiento para la especificación de

requerimientos basados en el estándar IEEE 830 y marcos de problemas, utilizando la metodología ideal para la construcción del sistema. De la evaluación del sistema podemos indicar que éste es un soporte para la generación de un documento legible y útil para el análisis y diseño de proyectos de software que puede ser utilizado como un referente de negociación.

El sistema experto permite inferir requerimientos funcionales utilizando la técnica interior-Exterior propuesta por Michael Jackson [10], en la que los requerimientos son identificados por los expertos para deducir un tipo de marco de problema, permitiendo al sistema dar recomendaciones sobre requerimientos funcionales, detallándolos de forma explícita, lo que asegura la confiabilidad de los requerimientos. En los que respecta a los requerimientos no funcionales fueron inferidos utilizando características propias de todos los sistemas software.

Antes de que los sistemas expertos puedan llegar a ser ampliamente comercializado estos productos de software deben ser confiables y deben utilizar metodologías adecuadas para su desarrollo. Esperamos haber contribuido a este esfuerzo por proponer un enfoque sobre la especificación de requerimientos de software, lo cual permitirá cubrir características importantes de una amplia gama de posibles aplicaciones de sistemas expertos.

#### REFERENCIAS

- [1] E. Kheirkhah, and A. Deraman, "Important factors in selecting Requirements Engineering Techniques", *Proceedings of International Symposium on Information Technology (ITSIM 2008)*, 2008, pp.1-5.
- [2] L. Jiang, A. Eberlein, B. H. Far, and M. Mousavi, "A Methodology for the selection of Requirement Engineering Techniques", *Software and Systems Modeling*, vol.7, Iss.3, 2008, pp. 303-328.
- [3] L. A. Macaulay, "Requirements Engineering. Applied Computing, Springer", 1996.
- [4] B. Nuseibeh, and S. Easterbrook, "Requirements engineering: a roadmap", in Finkelstein, A. (ed.) *the Future of Software Engineering*, ACM Press, and New York, 2000.
- [5] H. Meth, M. Brhel, and A. Maedche, "The state of the art in automated requirements elicitation," *Information and Software Technology*, vol. 55, pp. 1695-1709, 2013.
- [6] P. Colombo, F. Khendek, and L. Lavazza, "Bridging the gap between requirements and design: An approach based on Problem Frames and SysML," *Journal of Systems and Software*, vol. 85, pp. 717-745, 2012.
- [7] C. M. A. Álvarez, L. M. J. Hurtado, and C. M. Z. Jaramillo, "Estructuración del plan de gestión de la tecnología informática para el Departamento de Bibliotecas de la Universidad de Antioquia," Ed., 2004.
- [8] M. F. Rosique Contreras, M. Jiménez Buendía, and P. Sánchez Palma, "Evaluación de herramientas de gestión de requisitos," 2010.
- [9] J. K. Ang, S. B. Leong, C. F. Lee, and U. K. Yusof, "Requirement engineering techniques in developing expert systems," in *Computers & Informatics (ISCI)*, 2011 IEEE Symposium on, 2011, pp. 640-645.
- [10] America, P., Thiel, S., Ferber, S., & Mergel, M. (2001). *Introduction to domain analysis*. Eureka S, 2023.
- [11] P. Clements, L. Northrop. "Software Product Lines: Practices and Patterns", Addison-Wesley, 2001. 0-201-70332-7.
- [12] D. Díez Cebollero, "ComBLA: la aplicación del análisis de dominios al desarrollo de sistemas de aprendizaje asistido por ordenador," 2009.
- [13] M. Jackson. "Problem frames and software engineering". *Information and Software Technology*. 2005. 47(14), 903-912.
- [14] J. Pazos, "Análisis de Viabilidad en Sistemas Basados en Conocimiento", Máster en Ingeniería de software, Universidad Politécnica de Madrid, Instituto Tecnológico de Buenos Aires, 1997.

- [15] J. K. Ang, S. B. Leong, C. F. Lee, & U. K. Yusof. "Requirement engineering techniques in developing expert systems". In *Computers & Informatics (ISCI)*, March 2011 IEEE Symposium on pp. 640-645 IEEE.
- [16] , A. M. Hickey, & A. M. Davis. "Requirements elicitation and elicitation technique selection: model for two knowledge-intensive software development processes". In *System Sciences*, January 2003. *Proceedings of the 36th Annual Hawaii International Conference* on pp. 10-pp IEEE.
- [17] A. Gómez, N. Juristo, C. Montes, J. Pazos. "Ingeniería del Conocimiento, Editorial: Centro de Estudios Ramón Areces. 1997.
- [18] A. Corral, J. L. Carrillo. Tesis de Maestría: "Creación de un Sistema Experto para asistir al Ingeniero en Software en la elaboración de Documentos de Requerimientos" ,2015, Unpublished.



**Alexandra Corral D.** Nació en Latacunga Provincia de Cotopaxi en Ecuador. Es graduada de la Escuela Politécnica del Ejército, Latacunga-Ecuador en Sistemas e Informática en el año 1998 y cuenta con: Diplomado en Gestión del Desarrollo de Software en la Escuela Politécnica del Ejército, Diplomada en Gerencia de Marketing en la Universidad Autónoma de los Andes y actualmente está postulándose en el Doctorado en Ciencias de la Informática en la Universidad Nacional de

Argentina.

Actualmente Docente Tiempo Parcial del Departamento de Eléctrica y Electrónica de la ESPE Extensión Latacunga.

Email: [macorral@espe.edu.ec](mailto:macorral@espe.edu.ec).



**José Luis Carrillo M.** Nació en Latacunga Provincia de Cotopaxi en Ecuador. Es graduado de la Escuela Politécnica Nacional, Quito-Ecuador en Sistemas de Computación e Informática en el año 1997 y cuenta con un Diplomado en Gestión del Desarrollo de Software, dos Masterados: Máster en Informática Aplicada en la Escuela Politécnica del Chimborazo y Máster en Investigación e Innovación en Tecnologías de Información y Comunicación en la Universidad Autónoma de Madrid y actualmente es postulante a Doctorado en Informática y Telecomunicaciones.

Actualmente Docente Tiempo Completo del Departamento de Eléctrica y Electrónica de la ESPE Extensión Latacunga.

Email: [jlcarrillo@espe.edu.ec](mailto:jlcarrillo@espe.edu.ec)