



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN INNOVACIÓN Y
TRANSFERENCIA DE TECNOLOGÍA

MAESTRÍA EN INGENIERÍA DE SOFTWARE
I PROMOCIÓN

TESIS DE GRADO MAESTRÍA EN INGENIERÍA DE SOFTWARE

**TEMA: “CREACIÓN DE UN SISTEMA EXPERTO PARA
ASISTIR AL INGENIERO EN SOFTWARE EN LA
ELABORACIÓN DE DOCUMENTOS DE REQUERIMIENTOS”**

AUTORES: ING. CORRAL DÍAZ, MARÍA ALEXANDRA

ING. CARRILLO MEDINA, JOSÉ LUIS

DIRECTOR: MSC. EDDIE GALARZA ZAMBRANO

LATACUNGA, MARZO 2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
MAESTRÍA EN INGENIERÍA DE SOFTWARE
CERTIFICADO
MSC. EDDI GALARZA ZAMBRANO
CERTIFICA

En mi calidad de tutor del trabajo de grado, titulado: “CREACIÓN DE UN SISTEMA EXPERTO PARA ASISTIR AL INGENIERO EN SOFTWARE EN LA ELABORACIÓN DE DOCUMENTOS DE REQUERIMIENTOS”, presentado por la Ing. MARÍA ALEXANDRA CORRAL DÍAZ y el Ing. JOSÉ LUIS CARRILLO MEDINA, requisito previo para la obtención del título de MAGÍSTER en Ingeniería de Software, doy fe de que dicho trabajo reúne los requisitos y los méritos suficientes para ser sometido a presentación pública y evaluación por parte del jurado examinador que se designe.

En la ciudad de Latacunga, a los 10 días del mes de marzo del 2015.

MSc. Eddie Galarza Zambrano

**UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
MAESTRÍA EN INGENIERÍA DE SOFTWARE**

DECLARACIÓN DE RESPONZABILIDAD

ING. MARÍA ALEXANDRA CORRAL DÍAZ

ING. JOSÉ LUIS CARRILLO MEDINA

DECLARAMOS QUE

El contenido e información que se encuentra en ésta Tesis denominada “CREACIÓN DE UN SISTEMA EXPERTO PARA ASISTIR AL INGENIERO EN SOFTWARE EN LA ELABORACIÓN DE DOCUMENTOS DE REQUERIMIENTOS” es responsabilidad exclusiva del autor y ha respetado derechos intelectuales de terceros, conforme a las fuentes que se incorporan en la bibliografía.

En la ciudad de Latacunga, a los 10 días del mes de marzo del 2015.

Ing. María Alexandra Corral Díaz

Ing. José Luis Carrillo Medina

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
MAESTRÍA EN INGENIERÍA DE SOFTWARE

AUTORIZACIÓN

ING. MARÍA ALEXANDRA CORRAL DÍAZ

ING. JOSÉ LUIS CARRILLO MEDINA

Autorizo a la Universidad de las Fuerzas Armadas- ESPE, la publicación, en la biblioteca virtual de la Institución del trabajo de grado denominado **“CREACIÓN DE UN SISTEMA EXPERTO PARA ASISTIR AL INGENIERO EN SOFTWARE EN LA ELABORACIÓN DE DOCUMENTOS DE REQUERIMIENTOS”**, cuyo contenido, ideas y criterios son de nuestra responsabilidad y autoría.

En la ciudad de Latacunga, a los 10 días del mes de marzo del 2015.

Ing. María Alexandra Corral Díaz

C.C 0501970487

Ing. José Luis Carrillo Medina

C.C 0501553788

DEDICATORÍA

Dedico esta tesis en primer lugar a mis padres Angelito y Piedacita por ser los pilares fundamentales en mi vida, quienes me enseñaron a ser quien soy, por lo que doy gracias a Dios por tenerlos aún conmigo.

A mi querido esposo Germán compañero de vida que me apoyó moralmente y me dio la fortaleza necesarias para culminar con este proyecto.

A mis hijos Emily y Germancito que son la inspiración de mi vida por quienes lucho día a día, hijos de mi alma que con su inocencia y pureza han visto y han comprendido la ausencia de su madre en momentos difíciles en el largo trajinar de ésta etapa.

A mis hermanas Sofía y Claudia que juntas hemos sabido salir en las buenas y en las malas y sabemos que la vida es una constante lucha, y que con la gracia de Dios todo se puede. A mis tíos Víctor y Carmelita, a mis cuñados Rodney y Cali, y a mis sobrinos que siempre me han servido de apoyo en el caminar de mi vida.

Dedico también este trabajo a la Mami Gordita forjadora de un gran hombre como es mi esposo, y a toda mi familia política que de alguna manera han estado presentes.

ALEXANDRA

AGRADECIMIENTO

A Dios por la fortaleza que nos brinda día a día, y por darnos la gracia de tenernos con vida para seguir adelante, sabemos que si nos tiene aquí, será porque algo bueno nos espera en nuestro caminar.

El agradecimiento reiterado a nuestros familiares por el apoyo constante en todas las metas en las que nos hemos trazado.

Agradecemos de manera especial el apoyo incondicional y dedicación del Msc. Eddie Galarza por su acertada dirección en éste trabajo y por habernos permitido adquirir conocimientos y sobre todo su amistad.

Agradecemos a la Institución por brindar y abrirnos sus puertas y enseñarnos no solo conocimientos enfocados a nuestra carrera, sino también por enseñarnos a ser personas más humanas.

Alexandra, José Luis

ÍNDICE GENERAL

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1 Planteamiento del problema	1
1.2 Formulación del problema.....	2
1.3 Objetivos.....	4
1.4 Justificación e importancia	4
1.5. Planteamiento de hipótesis	5
1.5.1 Hipótesis general	5
1.5.2 Hipótesis específicas	5
1.6 Variables de investigación	6
1.6.1 Variable independiente	6
1.6.2 Variable dependiente	6
1.7 Operacionalización de variables	6
1.7.1. Variable Independiente.	6
1.7.2. Variable Dependiente	7

CAPÍTULO II

MARCO TEÓRICO

2.1 Introducción	8
2.2. Los requerimientos	9
2.3. Ingeniería de requerimientos	10
2.4 Dificultades para definir los requerimientos.....	13
2.5. Requerimientos funcionales y no funcionales	13
2.5.1 Requerimientos funcionales.....	14
2.5.2 Requerimientos no funcionales.....	14
2.6. El documento de requerimientos del software.....	15
2.6.1 Características de un buen ERS.....	16
2.7. Documento de requerimientos propuesto para el S.E.....	19
2.7.1 Dominio del problema	20
2.7.2 Requerimiento	20

2.7.3 Interfaces Externas	21
2.7.4 Restricciones de Diseño	21
2.7.5 Características atributos del sistema	21
2.7.6 Glosario	22
2.8 Marcos de problema	22
2.8.1 Marcos de Problemas elementales	24
2.9. Sistemas basados en el conocimiento	26
2.9.1. Sistemas expertos	27
2.9.2 Arquitectura	28
2.10. Metodología para la construcción del sistema experto	30
2.10.1 Fases y etapas de la metodología ideal	30

CAPÍTULO III

ESTUDIO DE VIABILIDAD

3.1 Introducción	35
3.2 Test de viabilidad	36
3.2.1 Cálculo del valor	39
3.3 Análisis de viabilidad para el proyecto	40
3.3.1 Dimensión de Plausibilidad	40
3.3.2 Dimensión de Justificación.....	42
3.3.3. Dimensión de Adecuación	43
3.3.4 Dimensión de Éxito	47
3.4 Cálculo de viabilidad del proyecto.....	51
3.4.1 Determinación del valor por Dimensión.....	52
3.4.2 Cálculo del valor de cada dimensión.....	55
3.4.3 Cálculo del valor final.....	59

CAPÍTULO IV

ADQUISICIÓN DEL CONOCIMIENTO

4.1 Introducción	60
4.2 Educción del Conocimiento.....	62
4.2.1 Primera reunión con el Experto.....	62
4.2.2 Segunda reunión con el experto	64
4.2.3 Tercera reunión con el experto	66
2.2.4 Cuarta reunión con el Experto	68

4.2.5 Quinta reunión con el experto	71
4.2.6 Sexta reunión con el Experto	78
4.3 Extracción del conocimiento	83
4.3.1 Análisis estructural de textos	83
4.3.2 Evaluación del análisis estructural de textos	86
4.4 Evaluación de la adquisición de conocimientos	86

CAPÍTULO V

CONCEPTUALIZACIÓN

5.1 Introducción	88
5.2 Análisis de la información y de los conocimientos.....	89
5.2.1 Conocimientos fácticos	89
5.2.1.1 Glosario de términos.....	89
5.2.1.2 Diccionario de Conceptos	92
5.2.2. Conocimientos estratégicos	98
5.2.2.3 Comprobación de los conocimientos estratégicos.....	105
5.2.3 Conocimientos tácticos	105
5.2.4 Conocimientos fácticos	125
5.2.4.1 Comprobación de los conocimientos fácticos.....	130
5.2.5 Modelo dinámico o de procesos.....	130
5.2.6 Modelo estático.....	134
5.3 Comprobación de la conceptualización.....	134

CAPÍTULO VI

FORMALIZACIÓN

6.1 Introducción	136
6.2 Selección de las Técnica de Formalización	136
6.2.1 Sistemas de producción.....	136
6.2.2 Representación de los conocimientos en marcos	136
6.3 Herramienta utilizada para el Sistema Experto	139

CAPÍTULO VII

SELECCIÓN DE LAS HERRAMIENTAS E IMPLEMENTACIÓN DEL SISTEMA

7.1 Introducción	142
7.2 Selección de la Herramienta	142
7.3 Interfaz con el usuario e implementación	145

7.3.1 Gestión de Usuarios	145
7.3.2 Mantenimiento de Catálogos	148
7.3.4 Administración de Proyectos.....	149
8.1 Introducción	174
8.2 Evaluación de la Adquisición de los conocimientos.....	175
8.3 Evaluación de la conceptualización de conocimientos.....	175
8.4 Evaluación de la Formalización de Conocimientos	176
8.5 Evaluación de la Implementación de conocimientos	176
8.6 Caso ejemplo 1.1: Facturación Electrónica	178

CAPÍTULO VIII

Conclusiones	203
---------------------------	------------

CAPÍTULO IX

Bibliografía	205
---------------------------	------------

CAPÍTULO X

ANEXOS

ANEXO 1	207
ANEXO 2.....	208

ÍNDICE DE FIGURAS

Figura 1. 1. Diagrama Causa Efecto de creación de un Sistema Experto.....	2
Figura 2. 1. Fenómenos compartidos	23
Figura 2. 2. Forma General de un Problema de software	23
Figura 2. 3. Arquitectura Básica de un Sistema Experto	29
Figura 3. 1. Valores Lingüísticos	37
Figura 3. 2 Intervalos de Dimensión de Plausibilidad	56
Figura 3. 3. Intervalos de Dimensión de Justificación	57
Figura 3. 4. Intervalos de Dimensión de Justificación	58
Figura 3. 5. Intervalos de Dimensión de Éxito	59
Figura 5. 1. Modelo Relacional	97
Figura 5. 2. Árbol de descomposición Funcional	98
Figura 5. 3. Diagrama jerárquico de tareas.....	131
Figura 5. 4. Jerarquía de tareas del documento de requerimientos	132

Figura 5. 5. Jerarquía de tareas de Descripción de requerimientos	133
Figura 5. 6. Mapa Conceptual.....	134
Figura 7. 1. Pantalla principal.....	146
Figura 7. 2. Pantalla Gestión de Usuarios.....	146
Figura 7. 3. Crear Usuario	147
Figura 7. 4. Modificar Usuario	147
Figura 7. 5. Eliminar Usuario.....	147
Figura 7. 6. Pantalla de acceso.....	148
Figura 7. 7. Pantalla Principal Mantenimiento de Catálogos.....	149
Figura 7. 8. Pantalla de acceso a Gestión Proyectos	149
Figura 7. 9. Icono de acceso al escritorio.....	150
Figura 7. 10. Pantalla principal Gestión de Proyectos	150
Figura 7. 11. Información de un nuevo proyecto.....	151
Figura 7. 12. Información del Proyecto.....	154
Figura 7. 13. Información adicional de la Empresa	155
Figura 7. 14. Pantalla Información Adicional	156
Figura 7. 15. Administración de entidades.....	157
Figura 7. 16. Agregar entidad.....	157
Figura 7. 17. Edición de Entidades	158
Figura 7. 18. Agregar atributo	159
Figura 7. 19. Eliminar entidad	159
Figura 7. 20. Creación de relaciones	160
Figura 7. 21. Gráfica del modelo conceptual con relaciones	161
Figura 7. 22. Eliminación de Relaciones	162
Figura 7. 23. Requerimientos funcionales.....	164
Figura 7. 24. Requisitos funcionales adicionales.....	165
Figura 7. 25. Requerimientos no funcionales	166
Figura 7. 26. Información general Requerimientos no funcionales.....	166
Figura 7. 27. Restricciones de Diseño.....	169
Figura 7. 28. Árbol del documento de especificación de requerimientos.....	170
Figura 7. 29. Área de edición	172
Figura 7. 30. Documento de Especificación de Requerimientos.....	172
Figura 7. 31. Anexos del Documento de Requerimientos	173
Figura 8. 1. Información General del Proyecto.....	179
Figura 8. 2. Definiciones-Acrónimos-Abreviaturas	180
Figura 8. 3. Modelo Conceptual del Sistema Facturación Electrónica	180

ÍNDICE DE TABLAS

Tabla 1. 1 Operacionalización de la variable independiente.....	6
Tabla 1. 2 Operacionalización de la variable dependiente.....	7
Tabla 2. 1 Documento de Requerimientos.....	20

Tabla 2. 2 Tipos de Marcos de Problemas	25
Tabla 3. 1 Valores lingüísticos – Puntos angulares.....	36
Tabla 3. 2 Puntos angulares para los valores booleano.....	37
Tabla 3. 3 Análisis del Test de Viabilidad según la Metodología Ideal.....	38
Tabla 3. 4 Dimensión de Plausibilidad.....	52
Tabla 3. 5 Dimensión de justificación	52
Tabla 3. 6 Dimensión de adecuación.....	53
Tabla 3. 7 Dimensión de éxito.....	54
Tabla 3. 8. Cálculo de la Dimensión Plausibilidad	55
Tabla 3. 9 Cálculo de la Dimensión Justificación.....	56
Tabla 3. 10 Cálculo de la Dimensión de Adecuación.....	57
Tabla 3. 11 Cálculo de la Dimensión de Éxito.....	58
Tabla 3. 12 Cálculo Valor Final.....	59
Tabla 5. 1 Glosario de términos.....	90
Tabla 5. 2 Diccionario de conceptos.....	92
Tabla 5. 3 Tabla Concepto/Atributo/valor.....	94
Tabla 5. 4 Diseño de seudoreglas.....	105
Tabla 5. 5 Análisis del Dominio: Marco de problema de Información	106
Tabla 5. 6 Análisis del Dominio: Marco de problema de control.....	107
Tabla 5. 7 Análisis del Dominio: Marco de problema de Workspiece.....	108
Tabla 5. 8 Análisis del Dominio: Marco de problema de conexión	108
Tabla 5. 9 Delimitaciones y restricciones: Herramientas de desarrollo.....	109
Tabla 5. 10 Delimitaciones y restricciones: Herramientas de desarrollo.....	109
Tabla 5. 11 Delimitaciones y restricciones: Herramientas de desarrollo.....	110
Tabla 5. 12 Delimitaciones y restricciones: Herramientas de desarrollo.....	110
Tabla 5. 13 Delimitaciones y restricciones: Herramientas de desarrollo.....	111
Tabla 5. 14 Delimitaciones y restricciones: Herramientas de desarrollo.....	111
Tabla 5. 15 Delimitaciones y restricciones: Herramientas de desarrollo.....	112
Tabla 5. 16 Delimitaciones y restricciones: Herramientas de desarrollo.....	112
Tabla 5. 17 Delimitaciones y restricciones: Herramientas de desarrollo.....	113
Tabla 5. 18 Delimitaciones y restricciones: Herramientas de desarrollo.....	113
Tabla 5. 19 Delimitaciones y restricciones: Herramientas de desarrollo.....	114
Tabla 5. 20 Conexión con sistemas externos: Servicios Web.....	114
Tabla 5. 21 Conexión con sistemas externos: Tramas.....	115
Tabla 5. 22 Conexión con sistemas externos: Bases de Datos	116
Tabla 5. 23 Conexión con sistemas externos: Bases de Datos	116
Tabla 5. 24 Conexión con sistemas externos: COM+.....	117
Tabla 5. 25 Definición requerimientos no Funcionales: Seguridad.....	117
Tabla 5. 26 Definición requerimientos no Funcionales: Seguridad.....	118
Tabla 5. 27 Definición requerimientos no Funcionales: Seguridad.....	119
Tabla 5. 28 Definición requerimientos no Funcionales: Seguridad.....	119
Tabla 5. 29 Definición requerimientos no Funcionales: Disponibilidad	120
Tabla 5. 30 Definición requerimientos no Funcionales: Disponibilidad	120
Tabla 5. 31 Definición requerimientos no Funcionales: Capacidad.....	121

Tabla 5. 32 Definición requerimientos no Funcionales: Capacidad.....	121
Tabla 5. 33 Definición requerimientos no Funcionales: Capacidad.....	122
Tabla 5. 34 Definición requerimientos Funcionales: Requerimiento de consultas	122
Tabla 5. 35 Definición requerimientos Funcionales: Requerimiento de Reglas de comportamiento.....	123
Tabla 5. 36 Definición requerimientos Funcionales: Operaciones sobre dominios creados.....	124
Tabla 5. 37 Descripción de Atributos.....	125
Tabla 5. 38 Descripción de Requerimientos Funcionales inferidos mediante Marcos de Problemas por el SSEE.....	128
Tabla 5. 39 Entradas para la Educción de Requerimientos no Funcionales: Restricciones de Diseño.....	129
Tabla 6. 1. Tabla-Concepto-Valor.....	137
Tabla 6. 2. Marco Clase Proyecto.....	137
Tabla 6. 3 Marco Clase Proyecto.....	137
Tabla 6. 4 Marco Clase Dominio.....	137
Tabla 6. 5 Marco Clase Requerimiento Funcional.....	138
Tabla 6. 6 Marco Clase Restricciones de diseño.....	138
Tabla 6. 7 Marco Clase Alcance del Sistema.....	138
Tabla 6. 8 Marco Clase Requerimiento no funcional.....	138
Tabla 6. 9 Marco Clase Requerimiento no funcional.....	139
Tabla 6. 10 Formulación de reglas en Clips.....	140
Tabla 6. 11 Ejemplo de una reglas en Clips para el Sistema Experto.....	140
Tabla 7. 1 Herramientas para desarrollo de sistemas expertos.....	142
Tabla 8. 1 Caso prueba Gestión de proyectos.....	178
Tabla 8. 2 Valores de caso de prueba especificación de requerimientos funcionales.....	181
Tabla 8. 3 Reglas de inferencia para requerimientos funcionales.....	183
Tabla 8. 4 Resultados obtenidos de las inferencias para el sistema SFE.....	188
Tabla 8. 5 Valores de caso de prueba especificación de requerimientos.....	190
Tabla 8. 6 Reglas de inferencia para requerimientos no funcionales.....	191
Tabla 8. 7 Resultados obtenidos de las inferencias del sistema experto.....	200

ÍNDICE DE ECUACIONES

Ecuación 3. 1 Cálculo del valor de una Dimensión.....	39
Ecuación 3. 2 Viabilidad total.....	40

RESUMEN

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 describir de un lenguaje natural a una especificación, en la que el requerimiento tenga aplicación en el dominio, esta descripción 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 trabajo nos hemos centrado 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
Especificación de requerimientos, Marcos de problema, Sistemas Expertos.

ABSTRACT

The complexity of software projects as well as the multidisciplinary nature of requirements engineering requires developers to carefully select the techniques and practices of requirements specification for the software development process. One of the biggest problems in the requirements specification is to describe a natural language to a specification, in which the requirement has application in the domain, this description is commonly affected by poor understanding of the application domain, the more work for the software engineer is to explore on unknown areas and work on a language in which both the user and the engineer, maintain a different level of abstraction. In this paper we have focused on the problems of specifying requirements assisted and / or supported by an expert system: indicating what to say. One of the techniques used for requirements specification is to determine the types of problems frameworks in order to obtain the definition of requirements. We have implemented an expert system which allows a requirements specification document based on the IEEE 830, understandable and legible for the initial phases of development of a software product document.

Key Words - Requirements Engineering, Problem analysis, Requirements Specification, Problem Frames, Expert Systems.

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA

Para cumplir con la construcción de un Sistema Software, se debe comprender múltiples elementos, desde el escenario en donde este se desarrolla, los riesgos que lo involucran y los recursos, siendo estos los que se deben gestionar hábilmente cuando se decide implementar un proyecto de desarrollo de software.

Una de las mayores deficiencias en la práctica de construcción de software es la poca atención que se presta a la discusión del problema. En general los desarrolladores se centran en la solución dejando el problema inexplorado. El problema a resolver debe ser deducido a partir de su solución.

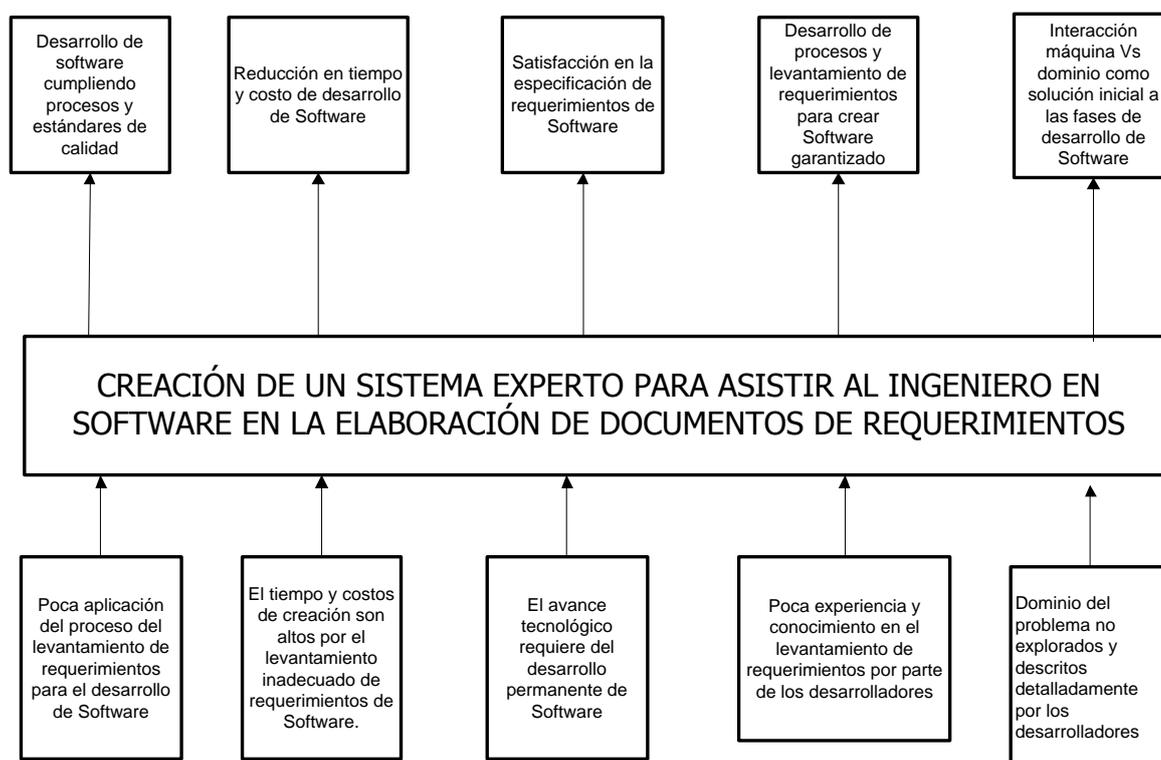
Esta aproximación, orientada a la solución, puede funcionar en campos donde todos los problemas sean bien conocidos, clasificados e investigados, en donde la innovación genere la detección de nuevas soluciones a problemas.

En el desarrollo de software tales características se cumplen parcialmente. La versatilidad de la tecnología y su rápida evolución hace que exista un repertorio de problemas en constante cambio, cuya solución software sea de enorme importancia.

Los proyectos de software han fallado porque sus requerimientos no han tenido una correcta descripción o fueron inadecuadamente explorados lo que origina el aumento de tiempo de desarrollo y de los costos iniciales del mismo.

Los requerimientos se ubican en el dominio de la aplicación donde está el problema, por lo que es necesario definirlo de una forma precisa y explícita

Lo detallado crea un efecto en la institución u organización, provocando el incremento de tiempos y costos que podrían haberse evitado elaborando una correcta documentación de los requerimientos de software. Es de ahí que nace la inquietud de desarrollar un sistema experto que promueva el levantamiento adecuado de la documentación de requerimientos de software.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 1. 1. Diagrama Causa Efecto de creación de un Sistema Experto

Existen pocas herramientas de soporte tecnológico, lo que da lugar a demora en tiempo e incremento de costos al elaborar la documentación de requerimientos de software.

1.2 FORMULACIÓN DEL PROBLEMA

La ESPE Extensión Latacunga oferta el Programa de IV Nivel de Ingeniería de Software, el mismo que tiene como propósito analizar la creación y desarrollo de software a nivel nacional e internacional bajo el cumplimiento de estándares, el problema se formula tomando en consideración las siguientes interrogantes.

¿Cuáles serán las herramientas que permitan construir un sistema experto que sirva de guía para la elaboración de la documentación de requerimientos de software?

¿Cuáles serán los recursos necesarios para desarrollar un software que valide un sistema experto de requerimientos para desarrollo de documentación?

¿Cuáles son las metodologías basadas en el conocimiento que permiten desarrollar un sistema experto?

¿Cómo detecta, en la etapa de documentación de requerimientos de software, contradicciones y ambigüedades, que influyan en el ciclo de vida de un producto de software?

¿Cuáles son las situaciones en que es escaso el conocimiento sobre el dominio de la aplicación?

¿Cómo se puede ahorrar tiempo y costos al elaborar el documento de requerimientos de software?

¿Cuáles son los beneficios que tendrá el desarrollo de un sistema experto para la identificación y análisis de requerimientos de un producto software?

El bajo nivel de conocimientos en el proceso de levantamiento de requerimientos de software, sumado al tiempo y costos de desarrollar requerimientos de software así como el creciente avance tecnológico en contraparte con la poca experiencia y conocimiento de desarrolladores, fundamenta que se realice la investigación debidamente justificada para la creación de un sistema experto para asistir al ingeniero en software en la elaboración de documentos de requerimientos.

1.3 OBJETIVOS

1.3.1 GENERAL

Crear un sistema experto que permita asistir al Ingeniero en Software usando metodologías basadas en el conocimiento para especificar requerimientos, cumpliendo los estándares de calidad referentes a esta área.

1.3.2 ESPECÍFICOS

- Determinar el estado del arte de los sistemas expertos y las diferentes metodologías que se aplican en su desarrollo.
- Adquirir los conocimientos de los expertos en desarrollo de software para obtener de una manera simple, sencilla y clara los requerimientos de software.
- Elevar el conocimiento sobre el dominio de aplicación en el que actúa un producto software.
- Identificar las formas para establecer el dominio de la aplicación para la solución por módulos.
- Crear el Sistema Experto que ayude a especificar correctamente a los requerimientos de software.

1.4 JUSTIFICACIÓN E IMPORTANCIA

La investigación se justifica técnicamente porque proporcionará una herramienta de apoyo para documentar los requerimientos de software constituyéndose en una importante ayuda para los desarrolladores.

Esta herramienta tendrá la posibilidad de guiar el proceso de levantamiento de requerimientos (antecedentes de los proyectos de las consultoras de software) y realizar un gran número de operaciones (es decir, tomar decisiones para la documentación de requerimientos de software) en poco tiempo de manera que se obtengan conclusiones y decisiones que sean un aporte al proceso de levantamiento requerimientos.

Económicamente permitirá al desarrollador de software reducir el tiempo para elaborar la documentación de requerimientos de software, evitando en algunos casos volver a especificarlos, lo que influirá en una reducción significativa de costos.

Socialmente se orienta al aporte que genera en la comunidad informática (desarrolladores de software) para elaborar una documentación de requerimientos de software oportuna y confiable. A la vez se beneficiará indirectamente a los clientes y/o usuarios del software, a través del apoyo del sistema experto.

1.5. PLANTEAMIENTO DE HIPÓTESIS

1.5.1 HIPÓTESIS GENERAL

¿Permitirá la creación e implementación de un sistema experto, asistir al Ingeniero en Software en el uso de metodologías basadas en el conocimiento, especificar requerimientos, al disponer de una herramienta que cumpla los estándares de calidad?

1.5.2 HIPÓTESIS ESPECÍFICAS

¿Las metodologías basadas en el conocimiento permiten desarrollar un sistema experto?

¿En la etapa de documentación de requerimientos de software; contradicciones y ambigüedades, generan la posibilidad de desalinear el proceso de creación y el ciclo de vida de un producto de software?

¿La cultura del desarrollo de software en nuestro país omite fases del levantamiento de requerimientos que influye en costos y calidad del producto?

¿Es indispensable que se cumpla con el proceso de levantamiento de requerimientos de software para conseguir un producto de calidad que satisfaga las necesidades del usuario?

1.6 VARIABLES DE INVESTIGACIÓN

En función del objetivo general: Crear un sistema experto que permita asistir al Ingeniero en Software usando metodologías basadas en el conocimiento para especificar requerimientos, reduciendo tiempos y costos cumpliendo los estándares de calidad referentes a esta área se plantean las variables de investigación.

1.6.1 VARIABLE INDEPENDIENTE

Sistema experto para asistir al Ingeniero en Software.

1.6.2 VARIABLE DEPENDIENTE

Documentos de requerimientos, que cumplen estándares de calidad.

1.7 OPERACIONALIZACIÓN DE VARIABLES

1.7.1. VARIABLE INDEPENDIENTE.

Sistema experto para asistir al Ingeniero en Software.

Tabla 1. 1 Operacionalización de la variable independiente.

CONCEPTO	CATEGORÍA	INDICADOR	PREGUNTAS
Sistema que realiza funciones en base al conocimiento de un experto.	<ul style="list-style-type: none"> • Tecnológica • Científica. 	Tipo de metodología a utilizar = 1.	¿Qué metodología es la más utilizada para desarrollar sistemas expertos?
		Sistema experto a diseñar =1	Que sistemas expertos son los más usados por el ingeniero de software
		Expertos a ser consultados para validación de información ≥ 3	¿Qué perfil debe reunir el experto para el levantamiento de requerimientos?

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

1.7.2. VARIABLE DEPENDIENTE

Documentos de requerimientos, que cumplen estándares de calidad.

Tabla 1. 2 Operacionalización de la variable dependiente.

CONCEPTO	CATEGORÍA	INDICADOR	PREGUNTAS
Especificación formal de requerimientos de software que satisfacen las necesidades del cliente.	Tecnológica Investigativa.	Tiempo utilizado para desarrollar el levantamiento de documentos de requerimientos ≥ 30 días.	¿De qué depende el tiempo para desarrollar el levantamiento de documento de requerimientos?
		Cumplimiento del estándar de especificación de requerimientos en un porcentaje $\geq 70\%$.	¿Qué parámetros del estándar de especificación de requerimientos son de cumplimiento obligatorio?
		Tiempo utilizado para desarrollar un software ≥ 30 días.	¿De qué depende el tiempo para desarrollar un levantamiento de software?

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

CAPÍTULO II

MARCO TEÓRICO

2.1 INTRODUCCIÓN

El reto para 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 en el proceso de desarrollo de Software.

En la Ingeniería de Software se ha introducido una serie de estándares para medir y certificar la calidad, tanto del sistema a desarrollar, como del proceso en sí, sin embargo es evidente que todavía existe una alta incidencia de fallos en proyectos de software causados por factores que inciden directamente en la calidad del proyecto como son planificaciones mal estructuradas, requerimientos incompletos, inconsistentes o mal establecidos y en general procesos del desarrollo de software que no cumplen estándares mínimos de calidad Díaz (2005).

La Ingeniería de Requerimientos provee en las fases tempranas del desarrollo de software la gestión y administración de requerimientos utilizando métodos, técnicas y herramientas que generen especificaciones correctas, claras y sin ambigüedades tal gestión es de vital importancia ya que guía y determina el alcance del proyecto de software.

La razón para desarrollar esta investigación se fundamenta en la gran cantidad de proyectos de software que no llegan a cumplir sus objetivos.

En nuestro país somos partícipes de este problema a diario, ya que se ha vuelto común la compra de sistemas extranjeros, para luego "personalizarlos" supuestamente a la medida de las empresas.

El uso de herramientas automatizadas en la gestión de requerimientos ayuda a los Ingenieros de Software a reducir el trabajo en la documentación, estas herramientas de acuerdo al estudio realizado por (Rosique Contreras,

Jiménez Buendía et al. 2010) se basan en su gran mayoría 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.

El uso de Sistemas Basados en Conocimiento a diferencia de los sistemas tradicionales permite analizar, guiar y tomar decisiones sobre algún tipo de problemas que puede ser resuelto computacionalmente. El proyecto de Investigación da como alternativa a los Ingenieros de Software que a través de un Sistema experto pueda servir de guía en el proceso de especificación y documentación de los requerimientos en el indique funcionalidad, comportamiento rendimiento e interconexión.

2.2. LOS REQUERIMIENTOS

De las definiciones que existen para requerimiento, a continuación se presentan las que aparecen en el glosario de la IEEE 610.12-1990.

- a) Una condición o capacidad requerida por un usuario para resolver un problema o alcanzar un objetivo.
- b) Una condición o capacidad que debe ser poseída por un sistema o componente del sistema para satisfacer un contrato, estándar, especificación, u otro documento formalmente impuesto.
- c) Una representación documentada de una condición o capacidad como las descritas en los incisos a) o b).

Lawrence (2002), define: “Un requerimiento es una característica del sistema o una descripción de algo que el sistema es capaz de hacer con el objeto de satisfacer el propósito del sistema”.

Silva (2000) manifiesta que: “Requisito: Algo que no sucede actualmente en el dominio pero que se desea que suceda en el futuro. Que ocurra este hecho en el futuro es una de las razones por las que se debe construir el sistema”.

Los requerimientos son necesidades de los clientes, la dificultad de convertir un dominio desconocido en especificaciones técnicas es uno de los problemas más comunes que presentan los analistas de requerimientos o Ingenieros de Software al momento de levantar un documento de requerimientos, por esta razón la propuesta del trabajo se basa en realizar una aproximación del dominio del problema que ayude a identificar lo funcional del futuro Software, de tal manera que los analistas de requerimientos reduzcan cambios significativos en versiones futuras del documento y que a su vez los desarrolladores puedan obtener un documento legible y fiable que les permita iniciar su fase de desarrollo.

2.3. INGENIERÍA DE REQUERIMIENTOS

La Ingeniería de Requerimientos provee el mecanismo para la comprensión de las necesidades de los usuarios utilizando métodos, técnicas y herramientas que permita analizar, documentar y especificar requerimientos de software.

A continuación se presentan algunas definiciones:

Jackson (1995) menciona "Ingeniería de Requerimientos trata de los fenómenos que ocurren en el mundo y no trata de los fenómenos que ocurren dentro de la máquina".

Boehm. (1979). Define: "Ingeniería de Requerimientos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema."

Guide (1987). Establece que: "Ingeniería de Requerimientos es el proceso por el cual se transforman los requerimientos declarados por los clientes, ya sean hablados o escritos, a especificaciones precisas, no ambiguas, consistentes y completas del comportamiento del sistema, incluyendo funciones, interfaces, rendimiento y limitaciones"

Leite (1987). Menciona: "Es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requerimientos"

"Ingeniería de requerimientos es un enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto" definido por Rational Software.

La Ingeniería de Requerimientos es definida también, según Ortas (1997), como: "Un conjunto de actividades en las cuales, utilizando técnicas y herramientas, se analiza un problema y se concluye con la especificación de una solución (a veces más de una)."

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

Establecer requerimientos técnicos detallados en niveles que incluyan todas las interfaces con gente y recursos técnicos es una de las partes más difíciles de construir un sistema manifiesta Brooks (1987).

Jackson (1995) cita: Que el primer paso para el Ingeniero del Sistema es el de capturar las necesidades del cliente. Estos requerimientos no se deben buscar en una descripción de la máquina a ser construida han de ser buscados en una descripción de los efectos que el cliente quiere que la máquina lleve a cabo.

La extracción del conocimiento se basa en el descubrimiento de los requerimientos del sistema. Aquí, los analistas de requerimientos deben trabajar junto al cliente para descubrir el problema que el sistema debe resolver, los diferentes servicios que el sistema debe prestar, las restricciones que se pueden presentar.

La extracción de los requerimientos involucra tener un conocimiento del dominio de la aplicación.

Una de las dificultades mayores en el proceso de extracción y de análisis es la mala comunicación que pueda existir entre el analista y el cliente ya que los dos involucrados pueden definir la misma necesidad con diferentes terminologías y perspectivas provocando ambigüedades e inconsistencias en los requerimientos.

Esta mala interpretación se da por que los clientes no siempre son buenos para describir con exactitud lo que desean y los desarrolladores no son buenos para comprender los intereses del negocio.

La etapa de análisis inicia luego de producirse un bosquejo general realizado en la extracción de los requerimientos, según Lawrence (2002) La manera de realizar un análisis del problema consiste en identificar las personas, procesos, recursos involucrados, y después documentar las relaciones que ocurren entre ellos.

Cada uno de los requerimientos de un sistema trata de los objetos o entidades, los estados en que éstos pueden estar y las funciones que se realiza para cambiar los estados o las características de los objetos.

Una forma muy útil para Lawrence (2002) es separar los requerimientos en tres categorías:

- 1.- Requerimientos que deben ser absolutamente satisfechos.
- 2.- Requerimientos que son muy deseables pero no indispensables.
- 3.- Requerimientos que son posibles, pero que podrían eliminarse.

Para poder separar estos requerimientos el Ingeniero de Software debe tener un conocimiento previo del dominio del problema que le permita visualizar el ámbito de la aplicación, establecer métodos de análisis que relacionen principios fundamentales es importante como cita Presman(1993) en los que se debería considerar:

1. Representar y comprender el ámbito de información del problema
2. Desarrollar los modelos que representan la información, funciones y el comportamiento del sistema.
3. Subdividir los modelos (y el problema) de forma que se descubran los detalles de una manera progresiva (o jerárquica).
4. El proceso de análisis debe ir de la información esencial hacia el detalle de la implementación.

2.4 DIFICULTADES PARA DEFINIR LOS REQUERIMIENTOS

Las mayores dificultades en las que puede encontrarse el analista a la hora de especificar los requerimientos son:

- Los requerimientos provienen de muchas fuentes y con distintos niveles de detalle
- Los requerimientos son difíciles de comprender por ser expresados en un lenguaje natural.
- Los requerimientos en algunas ocasiones son considerados como obvios produciendo pérdidas de información que puede ser relevante en el desarrollo del software.
- Los requerimientos nunca son iguales cada uno debe ser considerados con un cierto grado de prioridad.
- Los requerimientos no son declarados con trazabilidad que identifiquen el origen a través de documentos elicitados
- Cada requerimiento tiene propiedades únicas y abarcan áreas funcionales específicas dependiendo del proyecto.

2.5. REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

Los requerimientos describen el comportamiento de un sistema, estos pueden ser definidos como: funcionales y no funcionales.

2.5.1 REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales en un sistema describen la funcionalidad o los servicios que se espera que este provea. Estos dependen del tipo de software, del sistema que se desarrolle y también de los posibles usuarios. Cuando se expresan como requerimientos del usuario, habitualmente se describen de forma general mientras que los requerimientos funcionales del sistema detallan la función del mismo, sus entradas, salidas y excepciones.

Un requerimiento funcional describe una interacción entre el sistema y un ambiente Lawrence (2002). Los requerimientos funcionales son independientes de la implementación de una solución para el problema del cliente.

Los requerimientos funcionales definen el comportamiento específico del software, la especificación de estos requerimientos deben ser completos, consistentes y sin ambigüedades como lo describe en la IEEE std 830 sin embargo estas declaraciones son inicialmente informales que pueden causar inconsistencias entre lo entendido por los programadores y lo deseado por los clientes, la inconsistencia sucede cuando se rompe una regla Silva (2000). Los requerimientos funcionales ocupan la mayor parte del documento y en mucho de los casos los problemas son visualizados en fases posteriores del ciclo de vida del software.

2.5.2. REQUERIMIENTOS NO FUNCIONALES

Son aquellos requerimientos que no se refieren directamente a las funciones específicas que entrega el sistema, sino a características como la fiabilidad, la respuesta en el tiempo y la capacidad de almacenamiento Valdenegro (2011).

Los requerimientos no funcionales no son menos importantes que los funcionales ya que la falla de uno de estos inutiliza el sistema, estos requerimientos imponen restricciones en el diseño o en la implementación, establecen propiedades que le hacen al software usable, confiable y rápido.

Los requerimientos no funcionales surgen de la necesidad del usuario, debido a las restricciones en el presupuesto, a las políticas de la organización, necesidad de interoperabilidad con otros sistemas de software, hardware o a factores externos como los reglamentos de seguridad, las políticas de privacidad.

Tanto los requerimientos funcionales como no funcionales son obtenidos de los clientes de manera formal y cuidadosa Lawrence (2002). En un documento de especificación suele separarse los requerimientos funcionales de los no funcionales sin embargo separar estos dos tipos de requerimientos dificulta algunas veces ver la relación existentes entre ellos.

2.6. EL DOCUMENTO DE REQUERIMIENTOS DEL SOFTWARE

El documento de requerimientos es una manera de comunicar y guardar los requerimientos, este documento proporciona una representación del software que pueda ser revisada y aprobada por el cliente.

Una buena práctica es llevar los documentos de requerimientos a dos niveles como se detallan a continuación.

DRU: Documento de Requerimientos de Usuario el mismo que no requiere de mayor detalle ni de conocimiento técnico. Este documento incluye la descripción del problema y las metas que se espera lograr con la construcción del nuevo sistema, generalmente este documento es un medio legible para el cliente.

ERS: Especificación de Requerimientos de Software incluye con mayor detalle lo descrito por el DRU al que responde preguntas sobre las características que poseerá el nuevo sistema y que permitirá alcanzar los objetivos deseados.

En ocasiones un único documento puede servir para ambos propósitos que conlleve a un entendimiento entre analistas de requerimientos, clientes y

desarrolladores pero por lo general se debe llevar estos documentos por separado

La Especificación de Requerimientos de Software es la declaración oficial de lo que deben implementar los desarrolladores de software.

El documento de requerimientos describe los servicios y funciones que el sistema debe proveer, las restricciones bajo las cuales el sistema funciona, restricciones sobre las propiedades emergentes del sistema, definiciones de otros sistemas con los cuales el sistema integra información acerca del dominio de aplicación.

El estándar IEE std 830-1991 establece que un documento de requerimientos debe tener la siguiente estructura

- a. Introducción
 1. Propósito del documento de requerimientos
 2. Alcance del producto
 3. Definiciones, acrónimos y abreviaturas
 4. Referencias
 5. Descripción del resto del documento
- b. Descripción general
 1. Perspectiva del producto
 2. Funciones del producto
 3. Características de los usuarios
 4. Restricciones generales
 5. Suposiciones y dependencias.
- c. Requerimientos específicos: Incluyen los requerimientos funcionales, no funcionales y de interfaz.

2.6.1 CARACTERÍSTICAS DE UN BUEN ERS

Las características definidas por la IEEE std 830-1991 para un ERS son:

- a) Correcto;
- b) Inequívoco;

- c) Completo;
- d) Consistente;
- e) Delinear que tiene importancia y/o estabilidad;
- f) Comprobable;
- g) Modificable;
- h) Identificable.

a) Correcto

Un ERS es correcto si, y sólo si, cada requisito declarado se encuentra en el software.

b) Inequívoco

Un ERS es inequívoco si, y sólo si, cada requisito declarado tiene sólo una interpretación. Como un mínimo, se requiere que cada característica de la última versión del producto se describa usando un único término.

En casos dónde un término en un contexto particular tenga significados múltiples, el término debe ser incluido en un glosario dónde su significado es hecho más específico.

c) Completo

Un ERS está completo si, y sólo si, incluye los elementos siguientes:

- Los requisitos están relacionados a la funcionalidad, el desarrollo, las restricciones del diseño, los atributos y las interfaces externas. En particular debe reconocerse cualquier requisito externo impuesto por una especificación del sistema que debe ser tratado.
- La definición de las respuestas del software a todos los posibles datos de la entrada del sistema y a toda clase de situaciones. Una nota que es importante especificar son las contestaciones a las entradas válidas e inválidas a ciertos valores.

- Tener todas las etiquetas llenas y referencias a todas las figuras, tablas, diagramas en el ERS y definición de todas las condiciones y unidades de medida.

d) Consistente

Se refiere a la consistencia interior. Si un ERS no está de acuerdo con algún documento del superior-nivel, como una especificación de requisitos de sistema, entonces no es correcto.

e) Delinear que tiene importancia y/o estabilidad

Un ERS debe delinear la importancia y/o estabilidad si cada requisito en él tiene un identificador para indicar la importancia o estabilidad de ese requisito en particular. Típicamente, todos los requisitos que relacionan a un producto del software no son igualmente importantes. Algunos requisitos pueden ser esenciales, sobre todo para las aplicaciones de vida crítica, mientras otros pueden ser deseables.

f) Comprobable

Un ERS es comprobable si, y sólo si, cada requisito declarado es comprobable. Un requisito es comprobable si, y sólo si, allí existe algún proceso rentable finito con que una persona o la máquina puede verificar que el producto del software reúne el requisito. En general cualquier requisito ambiguo no es comprobable.

Los requisitos de No-verificable incluyen las declaraciones como "trabaja bien", "interface humana buena" y "normalmente pasará" no pueden verificarse los requisitos de esos porque es imposible de definir las condiciones "bueno," "bien" o "normalmente". La declaración que "el programa nunca entrará en una vuelta infinita" es el no-verificable porque la comprobación de esta calidad es teóricamente imposible.

g) Modificable

Un ERS es modificable si, y sólo si, su estructura y estilo son tales que puede hacerse cualquier cambio a los requisitos fácilmente, completamente y de forma consistente mientras conserva la estructura y estilo.

h) Identificable

Un ERS es identificable si el origen de cada uno de sus requisitos está claro y si facilita las referencias de cada requisito en el desarrollo futuro o documentación del mismo. Se recomiendan dos tipos de identificabilidad:

- El identificable dirigido hacia atrás (es decir, a las fases anteriores de desarrollo). Esto depende explícitamente en cada requisito de la referencias de su fuente en los documentos más antiguos.
- El identificable hacia adelante (es decir, a todos los documentos producidos posteriormente por el ERS).

Esto depende en cada requisito en el ERS que tiene un único nombre o número de la referencia. El identificable delantero del ERS es especialmente importante cuando el producto del software entra en el funcionamiento y en la fase de mantenimiento. Como el código y los documentos del plan se modifican, es esencial poder determinar el juego completo de requisitos que pueden afectarse por esas modificaciones.

2.7. DOCUMENTO DE REQUERIMIENTOS PROPUESTO PARA EL S.E

El documento de requerimientos propuesto para el Sistema Experto toma las partes funcionales de la Especificación de Requerimientos Software siguiendo el estándar IEEE 830, con los ítems que por experiencia de las empresas se ocupan como información necesaria para la documentación.

El documento de requerimientos tiene la estructura que se detalla en la tabla 2.1.

Tabla 2. 1 Documento de Requerimientos

Dominio del problema	Entidades, Atributos, relaciones Eventos Formatos Fuentes de Información
Requerimientos	Consultas Reglas de comportamiento Mapeos Operaciones en dominios creados Correspondencias entre dominios
Interfaces Externas	Sistemas Usuarios Hardware Software
Restricciones de Diseño	Las políticas de la empresa Las limitaciones del Hardware Las Interfaces a otras aplicaciones Los requisitos de lenguaje Los protocolos de comunicación La Seguridad y consideraciones de seguridad
Características/Atributos del Sistema	Disponibilidad Seguridad Capacidad
Glosario/Definiciones	Acrónimos Abreviaturas

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

2.7.1 DOMINIO DEL PROBLEMA

Realiza la descripción de entidades, relaciones existentes, eventos producidos. Para el documento de requerimientos se hace una aproximación a características de dominios ya explorados.

2.7.2 REQUERIMIENTO

Realiza la descripción del requerimiento estableciendo entradas y salidas, los requerimientos funcionales de acuerdo a la tipificación propuesta por Jackson(1995) pueden ser de cinco tipos: consultas, Reglas de

comportamiento, Mapeos, Operaciones en dominios creados, correspondencia entre dominios.

2.7.3 INTERFACES EXTERNAS

Debe incluirse en caso de existir requerimientos de interfaz con otros sistemas como hardware, software e interfaces de comunicaciones.

2.7.4 RESTRICCIONES DE DISEÑO

Se refiere a limitaciones que los desarrolladores pueden encontrar. Las restricciones definidas por la IEEE std 830-1991 son:

- a. Las políticas de la empresa
- b. Las limitaciones del Hardware
- c. Las Interfaces a otras aplicaciones
- d. El funcionamiento Paralelo
- e. Las funciones de la Auditoría
- f. Las funciones de Control
- g. Los requisitos de lenguaje
- h. Los protocolos de comunicación
- i. Los requisitos de Fiabilidad
- j. Credibilidad de la aplicación
- k. La Seguridad y consideraciones de seguridad.

2.7.5 CARACTERÍSTICAS ATRIBUTOS DEL SISTEMA

Es importante la especificación de los atributos del sistema para que su logro pueda verificarse objetivamente. Estos atributos son:

- a. Fiabilidad.
Especifica que los factores requeridos establezcan la fiabilidad requerida del sistema del software al momento de la entrega.
- b. Disponibilidad
Especifica que los factores requeridos garanticen un nivel de disponibilidad definido para el sistema, por ejemplo: puntos de recuperación.
- c. Seguridad

Especifica los factores requeridos que protegen al software del acceso accidental, malévolo, modificaciones o destrucción. Se recomienda uso de técnicas como la encriptación.

d. Mantenimiento

Especifica atributos de software que relaciona la facilidad de mantenimiento del propio software.

e. Portabilidad

Especifican atributos de software relacionados a la facilidad de poner el software a otro servidor y/o sistemas operativos.

2.7.6 GLOSARIO

Utiliza todas las terminologías usadas para el documento de requerimientos.

2.8 MARCOS DE PROBLEMA

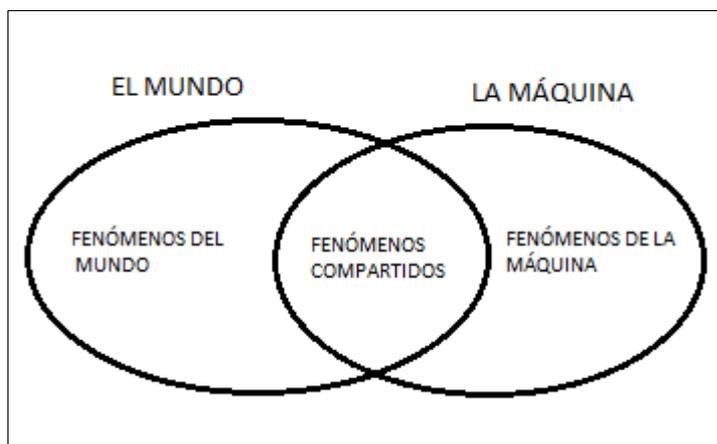
Para poder documentar los requerimientos es necesario hacer una aproximación al análisis del problema, la distinción entre requisito y entorno (dominio) resulta de vital importancia Silva (2000) “definir lo que es verdadero, independientemente o no de la existencia del sistema y aquello que no es verdadero pero que se desea que sea en el futuro esto es lo que diferencia entre requisito y dominio”.

Los marcos de problemas dan mayor énfasis a la comprensión de problemas reales, Jackson (1995) clasifica a estos marcos a través de la descomposición de sub-problemas más simples. Las relaciones existentes entre requisito, propiedades del dominio y especificaciones de la máquina caracterizan a clases de problemas.

El comportamiento externo de una máquina describe la especificación de requerimientos, estos requerimientos tienen relación con los fenómenos compartidos del dominio.

Los fenómenos compartidos son el lugar de interacción (intersección). Los fenómenos compartidos limitan la independencia de cada dominio, pero esta

limitación debe ser pequeña en comparación con el tamaño del dominio para que la partición tenga sentido como se ilustra en la Figura 2.1.



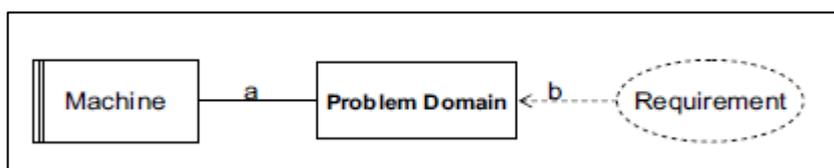
Fuente: Jackson

Figura 2. 1. Fenómenos compartidos

La Máquina interactúa con algunos dominios de la partición. Debe conectar el grafo de interacción de los dominios (un dominio que no tiene un camino de interacción hacia la máquina indica un error en el modelo).

Se debe entender cada dominio y las interacciones (fenómenos compartidos) entre estos y de estos con la máquina.

Los problemas de software tiene la siguiente forma



Fuente: Jackson

Figura 2. 2 Forma General de un Problema de software

La Figura 2.2 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 de información para llevar a cabo el requerimiento. La conexión desde el dominio del problema para la máquina está representada por fenómenos compartidos, "a" y "b". Fenómenos compartidos pueden ser datos, eventos, comandos y estados. La tarea del diseño de la solución es la construcción de la máquina.

La solución es construir la máquina de forma que asegure que el requisito tiene aplicación en el dominio.

Los requisitos deben ser escritos en términos del contexto del problema y no en términos de la máquina, ya que al mirar primero la solución y hablar en términos de la máquina estamos corriendo el riesgo de perder aspectos importantes de la especificación de requerimientos Jackson (1995), esto quiere decir que se debe tomar en cuenta aspectos implícitos y explícitos que pueden estar dentro del contexto y que no son tomados en cuenta a la hora de realizar la especificación de los mismos.

El uso de marcos de problema otorga dos ventajas importantes.

La primera es que si se posee un nutrido repertorio de clases de subproblemas conocidos en los cuales se puedan descomponer los problemas realistas, se obtendrá un proceso de descomposición guiado por una clasificación de problemas muy sistemática.

La segunda es que un marco de problema es asociado siempre con uno o más métodos para capturar el problema en detalle y desarrollar su solución.

2.8.1 MARCOS DE PROBLEMAS ELEMENTALES

En la tabla 2.2 se presentan marcos de problema que corresponde a 5 tipos de Requerimientos:

Tabla 2. 2 Tipos de Marcos de Problemas

Tipo de Requerimiento	Descripción	Marco de Problema
Consultas	Requerimiento de información sobre alguna parte del dominio del problema	Información
Reglas de comportamiento	Reglas que debe seguir el comportamiento del dominio del problema	Control
Mapeos	Mapeos sobre datos de entrada y de salida del software	Transformación
Operaciones sobre dominios creados	Operaciones que realizan los usuarios sobre objetos que existen solo dentro del software	Workpieces
Correspondencias entre dominios	Mantenimiento de dominios que no poseen fenómenos compartidos en sus estados correspondientes.	Conexión

Fuente: Jackson

Cada Marco de Problema es identificado de acuerdo al tipo de requerimiento, de esta manera el Sistema Experto propuesto determina una aproximación al dominio del problema.

a) Marcos de Información

Generalmente este tipo de marcos tienen requerimientos que son de consultas donde se muestra parte del mundo real en la que los usuarios obtienen información.

Para determinar este tipo de marco de problema los analistas realizan el análisis de la documentación de entrada y determinan si la información proporcionada responde a preguntas de consultas, pedidos de información, sin que en ningún caso se pretenda modificar el mundo real

b) Marcos de Control

Establece si se trata de un problema de control, analizando si el software controla un dispositivo haciendo que cumpla una serie de reglas de comportamiento.

c) Marco de problema de Workpiece

En un problema del tipo Workpieces, el software actúa como una herramienta para comparar objetos que existen solo dentro del mismo software.

Determinar si es un problema de tipo Workpiece significa que los usuarios requieran 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.

d) Marcos de problema de Transformación

Determinar si es un Marco de Problema de transformación significa que el software genere datos de salida mapeados contra datos de entrada de acuerdo a reglas específicas.

Para identificar este tipo de marcos se analiza a través de una documentación de entrada si el usuario realiza ciertos cálculos en un conjunto de datos de entrada que produzcan transformaciones a un conjunto de datos de salida.

e) Marcos de problemas de conexión

Los marcos de problemas de conexión son los que tienen relación con varios dominios.

2.9. SISTEMAS BASADOS EN EL CONOCIMIENTO

Los sistemas basados en el conocimiento procesan el conocimiento de la aplicación mediante representaciones cognitivas del ser humano como el aprendizaje y el razonamiento, que a diferencia de los sistemas convencionales la representación del conocimiento se lo realiza en forma algorítmica.

Según Peña (2006) una de las razones por la que se considera a los Sistemas Basados en el Conocimiento como una extensión tecnológica de

los Sistemas de Información es porque su complejidad es mayor y que dentro de sus propósitos descritos se encuentran:

- a. Aprender.
- b. Evolucionar.
- c. Adaptar.
- d. Razonar.
- e. Tomar decisiones.
- f. Contener conocimiento empírico, mundano y del lenguaje.
- g. Analizar problemas.
- h. Generar alternativas de solución.
- i. Emular al experto humano.
- j. Generar conocimiento a partir del que ya se posee.

Entre uno de los productos más significativos de los Sistemas Basados en el conocimiento se encuentran los Sistemas Expertos que son los encargados de representar el conocimiento de especialistas de una rama en la procura de su aprovechamiento para tareas específicas en el área de aplicación.

2.9.1. SISTEMAS EXPERTOS

Los Sistemas Expertos son programas diseñados para resolver problemas que normalmente son tomados del conocimiento de un experto humano en una rama específica y mediante esquemas propios del cerebro humano tales como memorización, razonamiento y aprendizaje automático para apoyar a un usuario en la toma de decisiones. En otras palabras un Sistema experta imita o emula una actividad del ser humano para resolver un problema en un área específica.

Definido por Peña (2006) “Un Sistema Experto es un sistema computacional que adquiere conocimiento especializado en un campo específico para explotarlo mediante métodos de razonamiento que emulan el desempeño del experto humano en la solución de problemas”.

Existen varias justificaciones para el diseño de un sistema experto:

- Hacer que el conocimiento sea más accesible para una mayor cantidad de usuarios.
- Que económicamente el conocimiento esté al alcance de más usuarios.
- Lograr la trascendencia del conocimiento a través del tiempo sin importar la posible desaparición del experto humano.
- Que la tasa de retorno de la inversión sea mayor a la inversión para el desarrollo del sistema.
- Que el conocimiento proporcione soluciones satisfactorias en un dominio específico

Sin embargo no todos los problemas son encaminados a la tecnología de los Sistemas Expertos solo aquellos que justifiquen y sea apropiados para un fin.

La posibilidad para realizar un sistema experto se basa en:

a) Que sea posible en cuanto a los expertos:

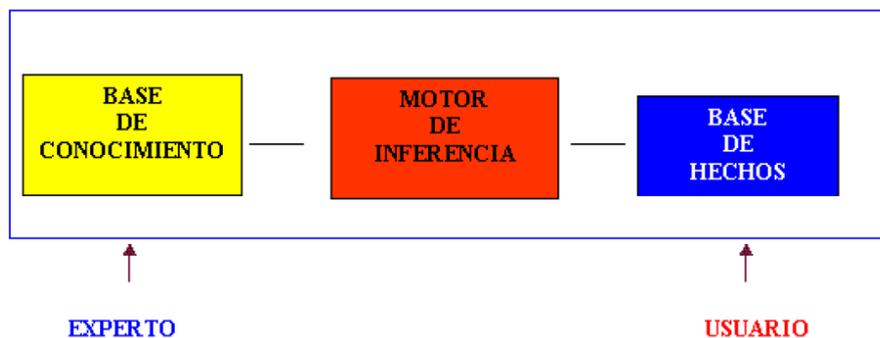
El primer requisito es que existan verdaderos expertos en el área, del problema a resolver. Si no existe una fuente de conocimiento y experiencia, el desarrollo fracasará.

b) Que sea posible en cuanto al problema:

La tarea debe pertenecer a un dominio en el cual el conocimiento existente para resolverla es preciso y está bien estructurado. Si la tarea fuese tan novedosa o no existe el conocimiento del dominio para resolver el problema, tampoco sería adecuada la utilización de la Ingeniería del Conocimiento como herramienta para resolver un Sistema Experto.

2.9.2 ARQUITECTURA

La arquitectura base para un Sistema Experto está organizada en base a tres elementos como muestra la Figura 2.3



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 2. 3. Arquitectura Básica de un Sistema Experto

a. **Base del conocimiento.**

En ella se almacena el conocimiento proporcionado por el experto humano en un área de la ciencia o el conocimiento humano. Una forma de representar este conocimiento es a través de reglas.

b. **Motor de inferencia.**

Se encarga de realizar las inferencias en la base del conocimiento de acuerdo con los parámetros definidos en la heurística del sistema y que permitirá inferir la solución de un problema o llevar a cabo la toma de decisiones es decir, el conocimiento de que dispone un sistema para gestionar un proceso de razonamiento. Dicho motor de inferencia trabaja con la información contenida en la base de conocimientos y la base de hechos para deducir nuevos hechos es considerado en un Sistema experto como el núcleo donde se encuentra la parte procedural.

c. **Base de hechos**

Contiene los hechos descubiertos en una consulta, es una memoria auxiliar en la que se almacenan los datos sobre situaciones concretas en la cual se va a realizar la aplicación (hechos iniciales que describen el enunciado del problema a resolver).

Tanto la Ingeniería de Software como la ingeniería de conocimientos convergen y son perfectamente complementarias por lo que es necesario conocer tanto las herramientas, técnicas y métodos de ambas disciplinas

para poder elaborar y desarrollar Software como lo manifiesta Acuña Silvia (2003). Por lo que para realización de nuestro trabajo nos hemos valido de la utilización de métodos y herramientas de Software convencional como de los Sistemas Basados en el Conocimiento (SBC).

2.10. METODOLOGÍA PARA LA CONSTRUCCIÓN DEL SISTEMA EXPERTO

La metodología utilizada en este trabajo es la metodología I.D.E.A.L desarrollado en la Facultad de Informática del Universidad Politécnica de Madrid, se tomó en cuenta esta metodología por la estructura gradual que toma en base al conocimiento del experto.

El desarrollo del sistema se basa en la filosofía del modelo en espiral en donde cada fase finaliza con un prototipo que conduce a la fase siguiente del ciclo. El mantenimiento perfectivo involucra la incorporación sistemática de nuevos conocimientos (adquiridos por el uso del sistema) típica cualidad de un SE. Los ejes de la base del cono representan el costo y el tiempo.

El eje de la calidad se representa de abajo hacia arriba; va de mayor diámetro o sea conocimientos menos específicos y de menor calidad, a menor diámetro, conocimientos más exactos y de mayor calidad. Al inicio del funcionamiento del sistema se obtienen grandes cantidades de conocimientos de distinta calidad pero a medida que el sistema se usa el conocimiento se refina, se obtienen menos conocimientos, pero de mayor calidad.

2.10.1 FASES Y ETAPAS DE LA METODOLOGÍA IDEAL

La Metodología IDEAL posee las siguientes fases y etapas Gómez et. al. (1997):

- a. FASE I: Requerimientos, viabilidad, especificación técnica**
 - I.1. Plan de requisitos y adquisición de conocimientos.
 - I.2. Evaluación y selección de la tarea.
 - I.3 Definición de las características de las tareas.

b. FASE II: desarrollo de los prototipos de demostración

- II.1. Concepción de la solución: Descomposición en subproblemas y determinación de analogías.
- II.2. Adquisición y conceptualización de los conocimientos.
- II.3. Formalización de los conocimientos y definición de la arquitectura.
- II.4. Selección de la herramienta e implementación.
- II.5. Validación y evaluación del prototipo.
- II.6. Definición de nuevos requisitos, especificaciones y diseño.

c. FASE III: Ejecución de la construcción del sistema integrado.

- III.1. Requisitos y diseño de la integración con otros sistemas.
- III.2. Implementación y evaluación de la integración.
- III.3. Aceptación por el usuario del sistema final.

d. FASE IV: Actuación para conseguir el mantenimiento perfecto.

- IV.1. Definir el mantenimiento del sistema global.
- IV.2. Definir el mantenimiento de las bases de conocimientos.
- IV.3. Adquisición de nuevos conocimientos y actualización del sistema.

e. FASE V: Lograr una adecuada transferencia tecnológica.

- V.1. Organizar la transferencia tecnológica.
- V.2. Completar la documentación del sistema basado en conocimientos construido.

A continuación se presenta una descripción breve de cada una de las fases según lo descrito por la Metodología IDEAL Gómez et. al. (1997).

a. Fase I: Requerimientos, viabilidad, especificación técnica

Incluye la definición de los objetivos de la aplicación y, en base a ellos, determinar si la tarea a realizar es susceptible de ser tratada con la tecnología de la INCO. Al completar la fase se logran definir las funcionalidades, el rendimiento y las interfaces que se espera que tenga

el sistema. Se define el objetivo y el alcance para el Sistema Basado en Conocimiento.

I.1 Plan de Requisitos y Adquisición de Conocimientos:

Comprende la identificación de las necesidades del cliente describiendo los objetivos del sistema. Es necesario que el Ingeniero en Conocimientos comience con una breve Adquisición de Conocimientos.

I.2 Evaluación y selección de la tarea: Conformar el llamado Estudio de Viabilidad. Se lleva a cabo realizando una evaluación de la tarea y su posterior cuantificación con el fin de determinar el grado de complejidad de la tarea.

I.3 Definición de las características de la tarea: Se configura la especificación del sistema, completando la especificación de conocimientos iniciales que se tienen del sistema.

b. Fase II: Desarrollo de los Prototipos de demostración, Investigación y campo operacional:

Concierne al desarrollo de los distintos prototipos que permiten ir definiendo y refinando, más rigurosamente, las especificaciones del sistema, de una forma gradual hasta conseguir las especificaciones exactas de lo que se puede hacer y cómo realizarlos. Los prototipos que se suelen desarrollar son los de: Demostración, Investigación, Campo y Operación. Esta fase se divide en las siguientes etapas:

II.1 Concepción de la solución (Descomposición en subproblemas y determinación de analogías): Esta etapa tiene como objetivo producir un diseño general del sistema prototipo. Engloba las actividades del desarrollo del Diagrama de Flujo de Datos y el diseño arquitectónico.

II.2 Adquisición y Conceptualización de los Conocimientos: Si bien la adquisición de conocimientos se realiza de forma constante, en ésta etapa es donde adquiere mayor importancia. Aparte de adquirir conocimientos, es también necesario conceptualizarlos para modelar el comportamiento del experto.

II.3 Formalización de los Conocimientos y definición de la Arquitectura: Las actividades que posee son: seleccionar los formalismos para representar en la máquina los conocimientos conceptualizados, y realizar el diseño detallado del Sistema Experto.

II.4 Selección de la herramienta e implementación: Seleccionar la herramienta de desarrollo adecuada al problema para luego realizar implementación.

II.5 Validación y Evaluación del Prototipo: En esta etapa se valida el funcionamiento del prototipo. Se suelen realizar Casos de Prueba y Ensayos en Paralelo con el Experto.

II.6 Definición de nuevos Requisitos: Se corresponde con la definición de los requisitos, especificaciones y diseño del siguiente prototipo.

c. Fase III. Ejecución de la Construcción del Sistema Integrado

Permite integrar el Sistema Experto a otros sistemas computacionales. Esta fase se divide en las siguientes etapas:

III.1 Requisitos y diseño de la integración: Consiste en el estudio y diseño de interfaces y puentes con otros Sistemas Hardware y Software.

III.2 Implementación y Evaluación del Sistema Integrado: Su fin es desarrollar, usando técnicas de Ingeniería en Software, los requisitos de la etapa anterior. Implementa la integración del Sistema Experto con los otros Sistemas Hardware y Software.

III.3 Aceptación del Sistema por el cliente: Es la prueba última de aceptación por los expertos y usuarios finales, que debe satisfacer la expectativa de fiabilidad y eficiencia.

d. Fase IV Actuación Para Conseguir el Mantenimiento Perfectivo.

Trata del mantenimiento perfectivo del sistema. Este mantenimiento viene reflejado en el ciclo de vida en la tercera dimensión de la espiral cónica. Esta fase se divide en las siguientes etapas:

IV.1 Definir el mantenimiento del Sistema global: Esta etapa emplea las técnicas de Ingeniería en Software, definiendo el mantenimiento de la base de conocimientos del Sistema Experto.

IV.2 Definir el Mantenimiento de las Bases de Conocimientos: Se encarga del mantenimiento de la Base de Conocimientos.

IV.3 Adquisición de nuevos Conocimientos y actualización del Sistema: Se encarga de diseñar nuevos protocolos para adquirir nuevos conocimientos y de incorporar al Sistema los nuevos conocimientos adquiridos.

e. Fase V. Lograr una adecuada transferencia tecnológica.

Para la correcta implantación del sistema, es necesaria una correcta transferencia tecnológica de manejo desde la parte que lo desarrolló hacia los usuarios finales. Esta fase se divide en las siguientes etapas:

V.1 Organizar la transferencia tecnológica: Se suelen usar entrenamientos en sesiones de tutoría entre los diseñadores y los usuarios para explicar el manejo del sistema y el entendimiento de la documentación.

V.2 Completar la Documentación del Sistema Basados en Conocimientos construido: Se debe crear el manual de usuario, el cual deberá estar orientado al usuario final.

El trabajo se enfocará a las dos primeras fases ya que se realizará un primer prototipo para la demostración del Sistema Experto.

CAPÍTULO III

ESTUDIO DE VIABILIDAD

3.1 INTRODUCCIÓN

La parte inicial para la construcción de un Sistema experto es probar la viabilidad del mismo para lo cual debemos definir la existencia real del experto para resolver el problema y si la tarea se encuentra dentro de un dominio que se pueda resolver en el campo de la Ingeniería del Conocimiento (INCO). Se evalúan los aspectos de Plausibilidad, Adecuación, Justificación y Éxito que caracterizan al problema utilizando el Test de Viabilidad propuesto por la Metodología IDEAL. A continuación se definen cada uno de estos aspectos Gómez et. al (1997).

Plausibilidad: Determina si se cuenta con los medios necesarios para poder abordar un problema a través de la Ingeniería del Conocimiento (INCO). Se analizan aspectos sobre la existencia del Experto y las tareas que lleva a cabo.

Justificación: Determina si se justifica el abordaje de la Ingeniería del Conocimiento INCO para resolver el problema. Se analizan aspectos sobre la necesidad de la experiencia y la inversión a realizar.

Adecuación: Determina si el problema es adecuado para ser resuelto por la INCO. Esto quiere decir que no todos los problemas pueden ser abordados por esta tecnología siendo únicamente necesario resolverlos por sistemas convencionales o algorítmicos que tienen un sentido común.

Éxito: Determina si existe la alta colaboración y predisposición por parte del Experto para el desarrollo del sistema y alta convicción de los directivos para afrontar los costos de este tipo de ingeniería.

3.2 TEST DE VIABILIDAD

Para llevar a cabo el test de evaluación que determine la viabilidad del proyecto es necesario emplear una métrica, la metodología IDEAL utiliza un método que consiste en asignar valores a las características agrupadas en cuatro dimensiones (plausibilidad, justificación, adecuación y éxito), la metodología permite incorporar la manipulación de valores lingüísticos asignándoles intervalos difusos que permitan realizar cálculos básicos.

Los valores considerados para las características son de tres tipos: Lingüísticos, Numéricos, y Booleanos. Según los valores de cada característica del problema se estima si el proyecto es viable o no.

Los valores lingüísticos pueden ser tomados en un conjunto de valores entre: Muy poco o nada, Poco, Regular, Mucho y Todo. Cuanto más verdadera parece la característica, mayor valor se le asigna, es decir, "mucho" o "todo", "poco" o "nada". Finalmente, el valor "regular" es para los casos en los que no se sabe muy bien. Estos valores se pueden ver como cuantificadores de las características.

Los valores lingüísticos deben ser convertidos a valores difusos correspondiente en el intervalo [0,10] como muestra la tabla 3.1.

Tabla 3. 1 Valores lingüísticos – Puntos angulares

Nada	0	0	1.2	2.2
Poco	1.2	2.2	3.4	3.4
Regular	3.4	3.4	5.6	6.6
Mucho	5.6	6.6	7.8	8.8
Todo	7.8	8.8	10	10

Fuente: Gómez. A

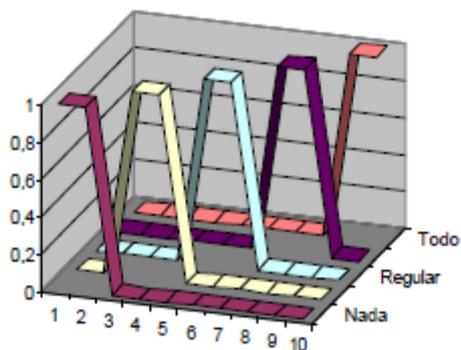


Figura 3. 1. Valores Lingüísticos

Fuente: Gómez. A

Un valor lingüístico se define por su función de pertenencia del intervalo $[0, 10]$ en el intervalo $[0, 1]$; que indica en qué grado se ajusta a dicho valor lingüístico, sabiendo que cuanto más se acerca la función a 1, más cierto es el valor lingüístico.

Los valores Numéricos pueden ser rangos $[X-Y]$ por ejemplo de $[0-9]$ y finalmente Booleanos que pueden definirse con valores de si, no.

Tabla 3. 2 Puntos angulares para los valores booleano

Sí	10	10	10	10
No	0	0	0	0

Fuente: Gómez. A

La tabla 3.3 muestra un análisis del Test de Viabilidad para determinar si un Sistema Experto es viable o no.

Tabla 3. 3 Análisis del Test de Viabilidad según la Metodología Ideal

Característica	Categoría	Dimensión	Peso	Naturaleza	Tipo	Umbral	Valor
Existen expertos, están disponibles y son cooperativos	Experto	P1	+10	Esencial	B	Si	
.....	Experto y Tarea	... P5	Esencial y Deseable	B, L, N	
Resuelve una tarea útil y necesaria	Tarea	J1	+8	Deseable	N	No	
.....	Experto, Tarea, Directivo, y Usuario	... J7	Esencial y Deseable	B, L, N	
La transferencia de experiencia entre humanos es factible	Tarea	A1	+7	Deseable	L	No	
.....	Experto, y Tarea	... 15	Esencial y Deseable	B, L	
Existe una ubicación idónea para el SE	Directivo, Usuario	E1	+7	Deseable	L	No	
.....	Experto, Directivo, Usuario y Tarea	... E23	Esencial y Deseable	B, L, N	

A continuación se da una explicación breve de cada uno de los campos ocupados en la matriz.

Las características se agruparán según su naturaleza y llamaremos a cada grupo “Dimensión”, existiendo cuatro dimensiones a saber: Plausibilidad, Éxito, Adecuación y Justificación Gómez. et al. (1997).

Categoría: Muestra a quien se referirá la característica: Directivos, Usuarios o a los Expertos.

Peso: Permite dar una importancia relativa a cada característica en la globalidad del test. El peso tiene dos componentes, uno de carácter numérico, que puede tomar un valor entero en el intervalo [1..10]. La otra es de carácter binario, la cual toma el valor + si la importancia favorece la construcción del SE y – si disminuye el grado de interés en el desarrollo del sistema. Estos valores son dados por la INCO

Naturaleza del valor asociado a la característica: Puede ser booleano, numérico o lingüístico.

Tipo: Una característica puede ser deseable o esencial y muestra su importancia.

Umbral: Es una referencia para características esenciales, este valor es fijo, pero es necesariamente igual en todas las características, y es de la misma naturaleza que el valor de las características.

Valor: Para cada proyecto hay que asignarle un valor a cada característica. Este valor es asignado por quien está realizando la evaluación.

3.2.1 CÁLCULO DEL VALOR

El cálculo del valor se realiza a través de los valores asignados a cada una de las características por Dimensión, utilizando para éste cálculo la ecuación 3.1

$$VC_i = \frac{1}{2} \frac{\sum_{k=1}^{r_i} P_{ik}}{\sum_{k=1}^{r_i} \frac{P_{ik}}{V_{ik}}} + \frac{1}{2} \frac{\sum_{k=1}^{r_i} P_{ik} V_{ik}}{\sum_{k=1}^{r_i} P_{ik}}$$

Ecuación 3. 1 Cálculo del valor de una Dimensión

Dónde:

VC_i : Valor global de la aplicación en una dimensión dada.

V_{ik} : Valor de la característica k en la dimensión i .

P_{ik} : Peso de la característica k en la dimensión i .

r_i : Número de la característica en la dimensión i .

Esta fórmula representa la viabilidad para una dimensión dada.

Valor por dimensión analizada

Cada dimensión está compuesta por 4 coordenadas, una por cada dimensión. Si el promedio de las 4 coordenadas pasa el umbral propuesto quiere decir que esa dimensión es cubierta favorablemente según lo establecido por la Metodología IDEAL.

En el caso de la característica de justificación se toma el valor máximo de los valores asignados a las características y luego se calcula la aproximación numérica de los intervalos difusos y se toma el máximo.

Para calcular la evaluación de viabilidad del proyecto, considerando todas las dimensiones se utiliza la media aritmética ponderada de los valores obtenidos para cada dimensión con los siguientes pesos:

- Plausibilidad y adecuación 8
- Justificación 3
- Éxito 5

Calculando con la ecuación 3.2.

$$V_f = \frac{\sum_{i=1}^4 P_i V_i}{\sum_{i=1}^4 P_i}$$

Ecuación 3. 2 Viabilidad total

Dónde:

V_f = Valor Final

Si es mayor que 6 el S.E es viable.

3.3 ANÁLISIS DE VIABILIDAD PARA EL PROYECTO

A continuación se presenta el análisis de cada una de las dimensiones para evaluar el Sistema Experto.

3.3.1 DIMENSIÓN DE PLAUSIBILIDAD

Característica P1: *Existen expertos, están disponibles y son cooperativos.*

Análisis: Existen expertos en el área de requerimientos. Se consideró como experto al Msc. Edison Espinosa el mismo que cursa un Doctorado en Ingeniería de Software y ha mostrado apertura y disponibilidad para brindar información. Se consideró además los Ing. Juan Peñaherrera y al Ing. Diego Velasco por la experiencia que tienen en desarrollo de Software y ser parte de la empresa Babel

Valor: Si

Característica P2: *El experto es capaz de estructurar sus métodos y procedimientos de trabajo.*

Análisis: Los expertos conocen de métodos empleados e identifican bien sus procedimientos en el trabajo a esto se suma su experiencia en elicitación y especificación de requerimientos.

Valor: Mucho

Característica P3: *La tarea está bien estructurada y se entiende.*

Análisis: La tarea se entiende por la existencia de información. Pero al momento de poner a la práctica la especificación de requerimientos se presentan ciertas dificultades con personas involucradas con poca experiencia en el proceso. Esta es una de las razones que toman los expertos para la existencia del sistema experto.

Valor: Todo

Característica P4: *Existen suficientes casos de pruebas y sus soluciones asociadas.*

Análisis: Existe información de cómo documentar requerimientos basándose en estándares que sirven de guías para realizar una especificación de requerimientos, además de las experiencias de la empresa con varios proyectos ejecutados.

Valor: 8

Característica P5: *La tarea solo depende de los conocimientos y no sólo del sentido común.*

Análisis: Es necesario partir de conocimientos previos e ir profundizando en la temática específica, esto debe garantizar un correcto abordaje, análisis y solución del problema.

Valor: 8

3.3.2 DIMENSIÓN DE JUSTIFICACIÓN

Característica J1: *Resuelve una tarea útil y necesaria.*

Análisis: El dotar de métodos, herramientas, sistemas que den soporte a las personas involucradas en el desarrollo software es fundamental más aún en etapas tempranas de desarrollo.

Valor: Mucho

Característica J2: *Se espera una alta tasa de recuperación de la inversión.*

Análisis: Se espera que la productividad sea mayor desde el momento de la elicitación de requerimientos para asegurar la calidad de lo que el usuario necesita desde etapas tempranas y no producir pérdidas económicas posteriores.

Valor: 7

Característica J3: *Hay escasez de experiencia humana.*

Análisis: Es importante, la experiencia constituye un factor a tomar en cuenta que garantiza el éxito de un proyecto software, de hecho son pocos los expertos para la especificación de requerimientos.

Valor: Regular.

Característica J4: *Hay necesidad de tomar decisiones en situaciones críticas o ambientes hostiles, penosos y/ o pocos gratificantes.*

Análisis: No hay situaciones críticas en ambientes hostiles ya que el trabajo se desarrolla en ambientes propios para el desarrollo de software.

Valor: Poco.

Característica J5: *Hay necesidad de distribuir los conocimientos.*

Análisis: Es necesaria la distribución de conocimientos en toda la comunidad que se dedica a procesos de desarrollo de software.

Valor: Todo

Característica J6: *Los conocimientos pueden perderse de no realizarse el sistema.*

Análisis: Al ser los requisitos información heurística y que depende de personal encargado específicamente de estas funciones, los conocimientos si pueden perderse.

Valor: Mucho

Característica J7: *No existen soluciones alternativas.*

Análisis: Si bien es cierto existen Herramientas para asistir en el proceso de desarrollo de software, pocas son las herramientas creadas en la toma de decisiones al elaborar un documento de requerimientos

Valor: Si

3.3.3. DIMENSIÓN DE ADECUACIÓN

Característica A1: *La transferencia entre humanos es viable.*

Análisis: La existencia de documentación disponible hace factible que dicha experiencia pueda transmitirse a otras personas mediante el guiado del proceso.

Valor: Mucho

Característica A2: *La tarea requiere “experiencia”.*

Análisis: La experiencia es indispensable para garantizar el éxito en los proyectos

Valor: Mucho

Característica A3: *Los efectos de la introducción del Sistema Experto no pueden preverse.*

Análisis: la introducción del Sistema Experto se espera que no cause efecto secundario ya que el software es para asistir al Ingeniero en el documento de requerimientos.

Valor: Regular

Característica A4: *La tarea requiere razonamiento simbólico.*

Análisis: Existe razonamiento simbólico ya que se formaliza las necesidades del usuario sirviendo como entrada del sistema.

Valor: Mucho

Característica A5: *La tarea requiere el uso de heurísticas para acotar el espacio de búsqueda.*

Análisis: Si se requiere de heurística en el momento de hacer un análisis del dominio del problema que permita acotar espacios de búsqueda para aconsejar al Ingeniero de Software en los pasos que debe seguir.

Valor: Mucho

Característica A6: *La tarea es de carácter público y más táctica que estratégica.*

Análisis: La tarea es de carácter táctica ya que toma de referencias estándares para la especificación de requerimientos los mismos que no afectan a la organización

Valor: Si

Característica A7: *Se espera que la tarea continúe sin cambios significativos durante un largo período de tiempo.*

Análisis: Se espera que no existan cambios significativos ya que los estándares así lo han demostrado y donde no han sufrido versiones cambiantes constantemente.

Valor: Mucho

Característica A8: *Se necesitan varios niveles de abstracción en la resolución de la tarea.*

Análisis: No se necesita niveles altos de abstracción ya que las tareas a resolver sobre técnicas de elicitación no son muy complejas.

Valor: Poco

Característica A9: *El problema es relativamente simple o puede descomponerse en subproblemas.*

Análisis: La descomposición en subproblemas se da ya que en el proceso de elicitación la información tomada como fuente debe ser categorizada para un posterior análisis.

Valor: Mucho

Característica A10: *El experto no sigue un proceso determinista en la resolución del problema.*

Análisis: Cada proyecto posee características propias por lo que requiere de análisis específico para cada uno de ellos.

Valor: Si

Característica A11: *La tarea acepta la técnica de prototipado gradual.*

Análisis: El refinamiento de requisitos en cada etapa requiere de un prototipado gradual ya que estos inicialmente no son muy claros.

Valor: Si

Característica A12: *El experto resuelve el problema a veces con información incompleta o incierta*

Análisis: Frecuentemente se tiene desconocimiento del dominio del problema a resolver, no se emplean técnicas adecuadas en la elicitación de requerimientos, no se analiza bien la información lo que lleva a tener información incompleta e incierta al momento de realizar un documento de especificación de requerimientos.

Valor: Mucho

Característica A13: *Es conveniente justificar las soluciones adoptadas.*

Análisis: Siempre se debe justificar, de hecho se deben establecer procesos formales para explicar las razones por las que se adoptó cierta(s) soluciones.

Valor: Mucho

Característica A14: *La tarea requiere investigación básica.*

Análisis: No requiere de investigación básica

Valor: No

Característica A15: *El sistema funcionará en tiempo real con otros programas o dispositivos.*

Análisis: El sistema es considerado como una herramienta de soporte y toma de decisiones propias para cada proyecto de Software en la que no requiere la interacción en tiempo real con otros programas o dispositivos.

Valor: No

3.3.4 DIMENSIÓN DE ÉXITO

Característica E1: *Existe una ubicación idónea para el SE.*

Análisis: Dado que el SE será utilizado por equipos de Ingeniería en requerimientos en coordinación con la persona encargada del Proyecto en lugares destinados para la ejecución. El SE no requiere una ubicación específica.

Valor: Regular

Característica E2: *Problemas similares se han resuelto con la Ingeniería del Conocimiento (INCO).*

Análisis: En una revisión bibliográfica se han encontrado problemas similares resueltos con la INCO en análisis de requerimientos.

Valor: Si

Característica E3: *El problema es similar a otros en los que resultó imposible aplicar esta tecnología.*

Análisis: No existen referencias de tales casos

Valor: No

Característica E4: *La comunidad del proyecto está influenciada por vaivenes políticos.*

Análisis: No hay influencias en este campo

Valor: Nada

Característica E5: *La inserción del sistema se efectúa sin traumas, es decir, apenas se interfiere en la rutina cotidiana.*

Análisis: El SE es un asistente de una tarea resuelta por los Ingenieros de requerimientos. Sin embargo podría generar cierta resistencia por cambios producidos.

Valor: Poco

Característica E6: *Se dispone de experiencia en INCO.*

Análisis: Se dispone de la experiencia de los expertos, a más de buenas referencias bibliográficas.

Valor: Regular.

Característica E7: *Se dispone de los recursos humanos, hardware y software necesarios para el desarrollo e implementación del sistema.*

Análisis: Se dispone tanto de recursos humanos como de hardware necesario para el desarrollo e implementación, ya que el recurso tecnológico no requiere ni de equipo ni software especializado.

Valor: Todo

Característica E8: *El experto resuelve el problema en la actualidad.*

Análisis: El experto lo resuelve siempre, siendo este un paso fundamental para el desarrollo de software

Valor: todo

Característica E9: *La solución del problema es prioridad para la institución.*

Análisis: Es prioritario en cuanto a que una buena especificación de requerimientos determinará la calidad del software al ser entregado.

Valor: Mucho

Característica E10: *Las soluciones son explicables.*

Análisis: Se debe dar soluciones explicables debido a que es una guía para el proceso inicial de desarrollo de software y los requerimientos deben tener trazabilidad y seguimiento entre ellos.

Valor: Mucho

Característica E11: *Los objetivos del sistema son claros y evaluables.*

Análisis: Los objetivos del sistema si son claros ya que se aplican estándares claros y evaluables por que se pueden comparar con lo elicitado e ir visualizando cambios para su respectiva depuración.

Valor: Mucho

Característica E12: *Los conocimientos están repartidos entre un conjunto de individuos.*

Análisis: Los conocimientos son de ámbito general para los Ingenieros de Software, por lo que no es necesaria la presencia de equipos simultáneos a la vez.

Valor: Poco

Característica E13: *Los directivos, usuarios, experto e Ingenieros del Conocimiento están de acuerdo en las funciones del SE.*

Análisis: Tanto los expertos como usuarios del sistema si están de acuerdo con las funciones del SE, ya que éste será aplicado a personas que requieran del beneficio del sistema.

Valor: Mucho

Característica E14: *La actitud de los expertos ante el desarrollo del sistema es positiva y no se sienten amenazados por el proyecto.*

Análisis: Se tiene la apertura y optimismo por parte de los Expertos sobre el SE

Valor: Mucho

Característica E15: *Los expertos convergen en sus soluciones y métodos.*

Análisis: Generalmente los expertos involucrados llegan a una misma solución.

Valor: Mucho

Característica E16: *Se acepta la planificación del proyecto propuesta por el IC.*

Análisis: Fue aceptada

Valor: Si

Característica E17: *Existen limitaciones estrictas de tiempo en la realización del sistema.*

Análisis: No existen limitaciones en tiempos, pero es deseable realizarlo en tiempos razonables.

Valor: Regular

Característica E18: *La dirección y usuarios apoyan los objetivos y directrices del proyecto.*

Análisis: Se aceptan las directrices del proyecto por ser parte del proceso.

Valor: Mucho

Característica E19: *El nivel de formación requerido por los usuarios del sistema es elevado.*

Análisis: Se aplica a Ingenieros de Software con conocimiento en requerimientos técnicos de Software

Valor: Mucho

Característica E20: *Las relaciones IC – Expertos son fluidas.*

Análisis: Son fluidas considerando los medios de comunicación existentes, pero limita un poco en cuanto a la distancia de algunos expertos.

Valor: Regular

Característica E21: *El proyecto forma parte de un camino crítico con otros sistemas.*

Análisis: No forma parte de un camino crítico.

Valor: No

Característica E22: *Se efectuará una adecuada transferencia tecnológica.*

Análisis: Se ha previsto la elaboración de un plan de capacitación a los usuarios, como también la confección de los manuales necesarios para el uso del SE.

Valor: Mucho

Característica E23: *Lo que cuenta en la solución es la calidad de la respuesta.*

Análisis: La calidad de la respuesta es importante ya que de esta depende para una buena especificación de requerimientos evitando tiempo y costos en el sistema.

Valor: Sí

3.4 CÁLCULO DE VIABILIDAD DEL PROYECTO

Para el cálculo de la Viabilidad se realiza los siguientes pasos.

- De acuerdo a las justificaciones de las características descritas se determina el valor correspondiente a cada Dimensión.

- Se calcula el valor para cada dimensión. Presentada en cada tabla para el cálculo correspondiente.
- Se calcula el valor final que consiste en el cálculo de la media aritmética ponderada del resultado obtenido para cada dimensión

3.4.1 DETERMINACIÓN DEL VALOR POR DIMENSIÓN

Tabla 3. 4 Dimensión de Plausibilidad

Denominación de la Característica	Categoría	Dimensión	Peso	Tipo	Naturaleza	Umbral	Valor
Existen expertos, están disponibles y son cooperativos.	Experto	P1	10	Esencial	Booleana	Sí (sí)	Si
El experto es capaz de estructurar sus métodos y procedimientos de trabajo.	Experto	P2	7	Deseable	Difusa	No	Mucho
La tarea está bien estructurada y se entiende.	Tarea	P3	8	Deseable	Difusa	No	Todo
Existen suficientes casos de prueba y sus soluciones asociadas.	Tarea	P4	10	Esencial	Numérica	Sí (8)	8
La tarea sólo depende de los conocimientos y no del sentido común.	Tarea	P5	9	Deseable	Numérica	No	8

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 3. 5 Dimensión de justificación

Denominación de la Característica	Categoría	Dimensión	Peso	Tipo	Naturaleza	Umbral	Valor
Resuelve una tarea útil y necesaria.	Tarea	J1	8	Deseable	Difusa	No	Mucho
Se espera una alta tasa de recuperación de la inversión	Directivos / Usuarios	J2	7	Deseable	Numérica	No	7
Hay escasez de experiencia humana.	Experto	J3	6	Deseable	Difusa	No	Regular
Hay necesidad de tomar decisiones en situaciones críticas o ambientes hostiles, penosos y/o pocos gratificantes.	Tarea	J4	10	Deseable	Difusa	No	Poco
Hay necesidad de distribuir	Tarea	J5	10	Deseable	Difusa	No	Todo

los conocimientos.							
Los conocimientos pueden perderse de no realizarse el sistema.	Experto	J6	10	Deseable	Difusa	No	Mucho
No existen soluciones alternativas.	Tarea	J7	8	Esencial	Booleana	Si(si)	Si

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 3. 6 Dimensión de adecuación

Denominación de la Característica	Categoría	Dimensión	Peso	Tipo	Naturaleza	Umbral	Valor
La transferencia de conocimientos entre humanos es factible.	Tarea	A1	7	Deseable	Difusa	No	Mucho
La tarea requiere "experiencia".	Tarea	A2	10	Deseable	Difusa	No	Mucho
Los efectos de la introducción del SE no pueden preverse.	Tarea	A3	-2	Deseable	Difusa	No	Regular
La tarea requiere razonamiento simbólico.	Tarea	A4	5	Deseable	Difusa	No	Mucho
La tarea requiere el uso de heurística para acotar el espacio de búsqueda.	Tarea	A5	7	Deseable	Difusa	No	Mucho
La tarea es de carácter práctico y más táctica que estratégica.	Tarea	A6	8	Deseable	Booleana	No	Si
Se espera que la tarea continúe sin cambios significativos durante un largo período de tiempo.	Tarea	A7	8	Esencial	Difusa	Si (Mucho)	Mucho
Se necesitan varios niveles de abstracción en la resolución de la tarea.	Tarea	A8	8	Deseable	Difusa	No	Poco
El problema es relativamente simple o puede descomponerse en subproblemas.	Tarea	A9	6	Deseable	Difusa	No	Mucho
El experto no sigue un proceso determinista en la resolución del problema.	Experto	A10	3	Deseable	Booleana	No	Si
La tarea acepta la técnica de prototipado gradual.	Tarea	A11	8	Deseable	Booleana	No	Si
El experto resuelve el problema a veces con información incompleta o incierta.	Experto	A12	3	Deseable	Difusa	No	Mucho
Es conveniente justificar las	Tarea	A13	3	Deseable	Difusa	No	Mucho

soluciones adoptadas.							
La tarea requiere investigación básica.	Tarea	A14	-10	Esencial	Booleana	Si (No)	No
El sistema funcionará en tiempo real con otros programas o dispositivos.	Tarea	A15	-6	Deseable	Difusa	No	No

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 3. 7 Dimensión de éxito

Denominación de la Característica	Categoría	Dimensión	Peso	Tipo	Naturaleza	Umbral	Valor
Existe una ubicación idónea para el SE.	Directivos / Usuarios	E1	7	Deseable	Difusa	No	Regular
Problemas similares se han resuelto con INCO.	Tarea	E2	8	Deseable	Booleana	No	Si
El problema es similar a otros en lo que resultó imposible aplicar esta tecnología.	Tarea	E3	-5	Deseable	Booleana	No	No
La continuidad del proyecto está influenciada por vaivenes políticos.	Directivos / Usuarios	E4	-9	Esencial	Difusa	Si (Poco)	Nada
La inserción del sistema se efectúa sin traumas, es decir, apenas se interfiere en la rutina cotidiana.	Directivos / Usuarios	E5	8	Deseable	Difusa	No	Poco
Se dispone de experiencia en INCO.	Tarea	E6	7	Deseable	Difusa	No	Regular
Se dispone de los recursos humanos, hardware y software necesarios para el desarrollo e implementación del sistema.	Tarea	E7	4	Deseable	Difusa	No	Todo
El experto resuelve el problema en la actualidad.	Experto	E8	4	Deseable	Difusa	No	Todo
La solución del problema es prioridad para la institución.	Directivos / Usuarios	E9	8	Esencial	Difusa	Si (Mucho)	Mucho
Las soluciones son explicables.	Tarea	E10	5	Deseable	Difusa	No	Mucho
Los objetivos del sistema son claros y evaluables	Tarea	E11	6	Deseable	Difusa	No	Mucho
Los conocimientos están repartidos entre un conjunto de individuos.	Experto	E12	-7	Deseable	Difusa	No	Poco
Los directivos, usuarios, experto e IC están de	Directivos /	E13	4	Esencial	Difusa	Si (mucho)	Mucho

acuerdo en las funcionalidades del SE.	Usuarios						
La actitud de los expertos ante el desarrollo del sistema es positiva y no se sienten amenazados por el proyecto.	Experto	E14	8	Deseable	Difusa	No	Mucho
Los expertos convergen en sus soluciones y métodos.	Experto	E15	5	Deseable	Difusa	No	Mucho
Se acepta la planificación del proyecto propuesta por el IC.	Directivos / Usuarios	E16	8	Esencial	Booleana	Si (si)	Si
Existen limitaciones estrictas de tiempo en la realización del sistema.	Tarea	E17	-6	Deseable	Difusa	No	Regular
La dirección y usuarios apoyan los objetivos y directrices del proyecto.	Directivos / Usuarios	E18	7	Esencial	Difusa	Si (Mucho)	Mucho
El nivel de formación requerido por los usuarios del sistema es elevado	Directivos / Usuarios	E19	-2	Deseable	Difusa	No	Mucho
Las relaciones IC - Experto son fluidas.	Experto	E20	4	Deseable	Difusa	No	Regular
El proyecto forma parte de un camino crítico con otros sistemas.	Tarea	E21	-6	Deseable	Booleana	No	No
Se efectuará una adecuada transferencia tecnológica.	Directivos / Usuarios	E22	8	Esencial	Difusa	Si (Mucho)	Mucho
Lo que cuenta en la solución es la calidad de la respuesta.	Experto	E23	5	Deseable	Booleana	No	Si

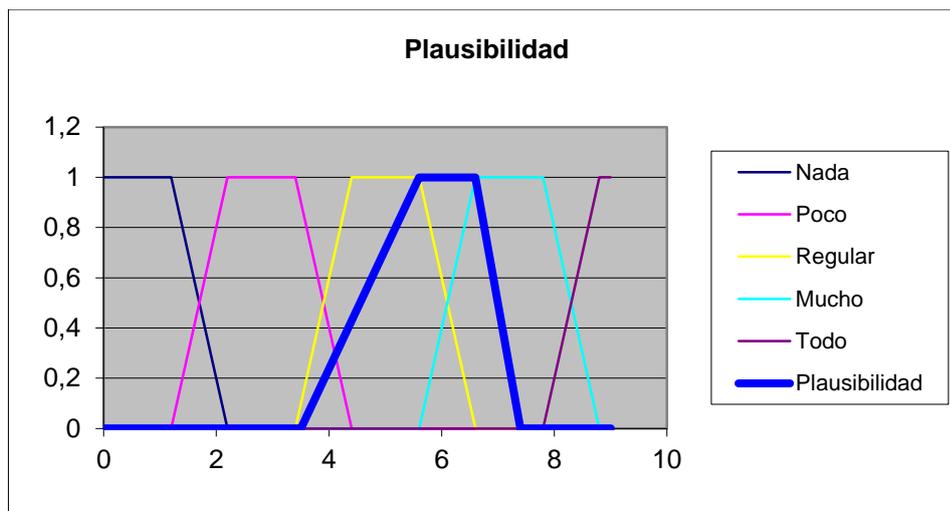
Elaborado por: Corral, María Alexandra, Carrillo José Luis.

3.4.2 CÁLCULO DEL VALOR DE CADA DIMENSIÓN

Tabla 3. 8. Cálculo de la Dimensión Plausibilidad

Característica	Peso	Valor	Intervalo Difuso				Peso*Valor				Peso/Valor			
P1	10	Si	10	10	10	10	100	100	100	100	1,00	1,00	1,00	1,00
P2	7	Mucho	5,6	6,6	7,8	8,8	39,2	46,2	54,6	61,6	1,25	1,06	0,90	0,80
P3	8	Todo	7,8	8,8	10	10	62,4	70,4	80	80	1,03	0,91	0,80	0,80
P4	10	8	8	8	8	8	80	80	80	80	1,25	1,25	1,25	1,25
P5	9	8	8	8	8	8	72	72	72	72	1,13	1,13	1,13	1,13
44							354	369	387	394	5,65	5,34	5,07	4,97
Resultado:							7,91	8,3	8,73	8,9				

Elaborado por: Corral, María Alexandra, Carrillo José Luis.



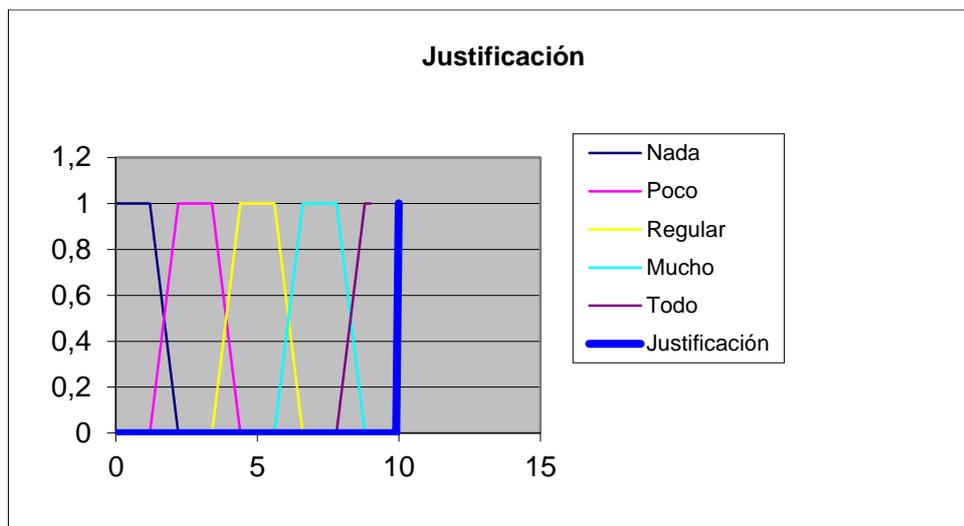
Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 3. 2 Intervalos de Dimensión de Plausibilidad

Tabla 3. 9 Cálculo de la Dimensión Justificación

Característica	Peso	Valor	Intervalo Difuso				Peso*Valor				Aprox. Numérica	
J1	8	Mucho	5,6	6,6	7,8	8,8	44,8	52,8	62,4	70,4	57,6	
J2	7	7	7	7	7	7	49	49	49	49	49	
J3	6	Regular	3,4	4,4	5,6	6,6	20,4	26,4	33,6	39,6	30	
J4	10	Poco	1,2	2,2	3,4	4,4	12	22	34	44	28	
J5	10	Todo	7,8	8,8	10	10	78	88	100	100	91,5	
J6	10	Mucho	5,6	6,6	7,8	8,8	56	66	78	88	72	
J7	8	Sí	10	10	10	10	80	80	80	80	80	
59										Max:	91,5	
Resultado:							7,8	8,8	10	10		

Elaborado por: Corral, María Alexandra, Carrillo José Luis.



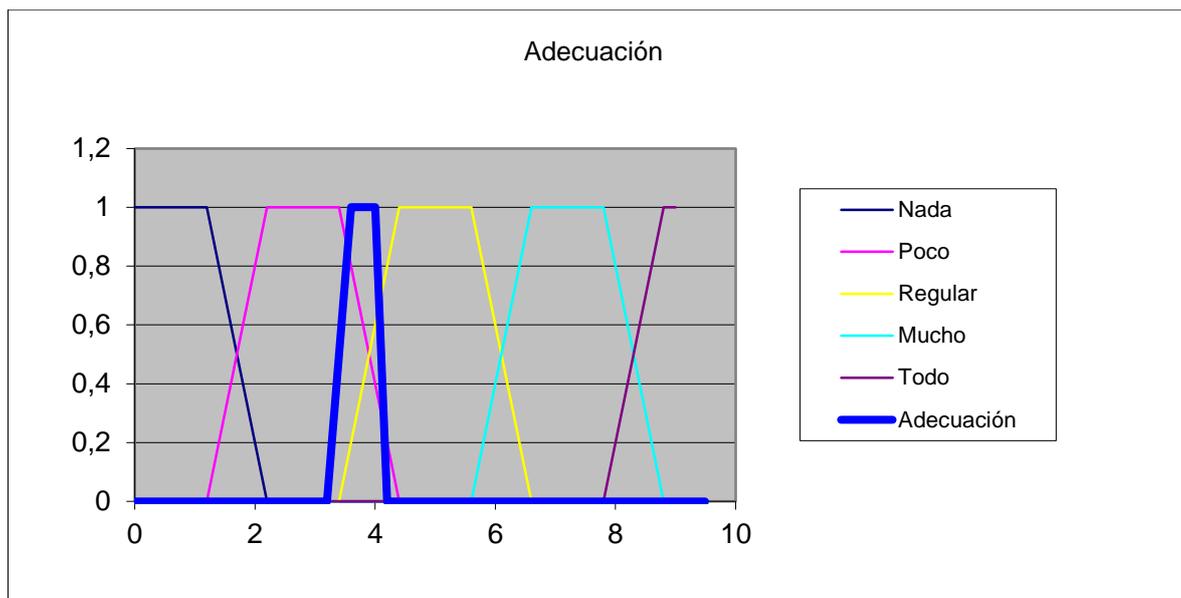
Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 3. 3. Intervalos de Dimensión de Justificación

Tabla 3. 10 Cálculo de la Dimensión de Adecuación

Característica	Peso	Valor	Intervalo Difuso				Peso*Valor				Peso/Valor				
A1	7	Mucho	5,6	6,6	7,8	8,8	39,2	46,2	54,6	61,6	1,25	1,06	0,90	0,80	
A2	10	Mucho	5,6	6,6	7,8	8,8	56	66	78	88	1,79	1,52	1,28	1,14	
A3	2	Regular	3,4	4,4	5,6	6,6	6,8	8,8	11,2	13,2	0,59	0,45	0,36	0,30	
A4	5	Mucho	5,6	6,6	7,8	8,8	28	33	39	44	0,89	0,76	0,64	0,57	
A5	7	Mucho	5,6	6,6	7,8	8,8	39,2	46,2	54,6	61,6	1,25	1,06	0,90	0,80	
A6	8	Sí	10	10	10	10	80	80	80	80	0,80	0,80	0,80	0,80	
A7	8	Mucho	5,6	6,6	7,8	8,8	44,8	52,8	62,4	70,4	1,43	1,21	1,03	0,91	
A8	8	Poco	1,2	2,2	3,4	4,4	9,6	17,6	27,2	35,2	6,67	3,64	2,35	1,82	
A9	6	Mucho	5,6	6,6	7,8	8,8	33,6	39,6	46,8	52,8	1,07	0,91	0,77	0,68	
A10	3	Sí	10	10	10	10	30	30	30	30	0,30	0,30	0,30	0,30	
A11	8	Sí	10	10	10	10	80	80	80	80	0,80	0,80	0,80	0,80	
A12	3	Mucho	5,6	6,6	7,8	8,8	16,8	19,8	23,4	26,4	0,54	0,45	0,38	0,34	
A13	3	Mucho	5,6	6,6	7,8	8,8	16,8	19,8	23,4	26,4	0,54	0,45	0,38	0,34	
A14	10	No	0	0	0	0	0	0	0	0	0,00	0,00	0,00	0,00	
A15	6	No	0	0	0	0	0	0	0	0	0,00	0,00	0,00	0,00	
94							480,80	539,80	610,60	669,60	17,90	13,42	10,89	9,59	
							Resultado:	5,18	6,37	7,56	8,46				

Elaborado por: Corral, María Alexandra, Carrillo José Luis.



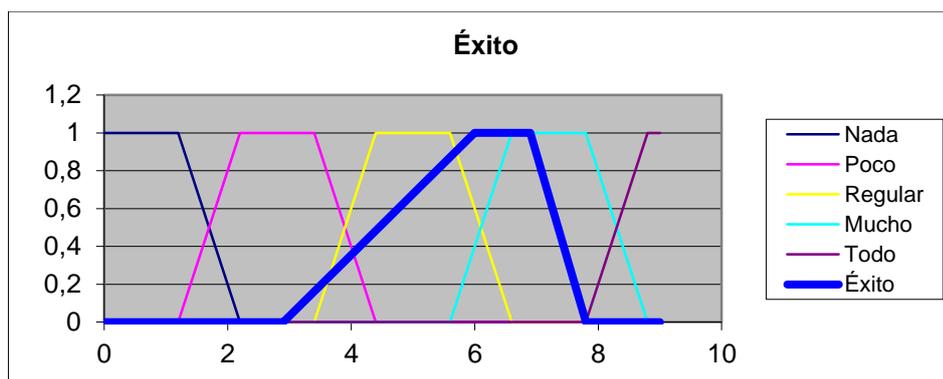
Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 3. 4. Intervalos de Dimensión de Justificación

Tabla 3. 11 Cálculo de la Dimensión de Éxito

Característica	a	Peso	Valor	Intervalo Difuso				Peso*Valor				Peso/Valor			
				3,4	4,4	5,6	6,6	23,80	30,80	39,20	46,20	2,06	1,59	1,25	1,06
E1	7	Regular	3,4	4,4	5,6	6,6	23,80	30,80	39,20	46,20	2,06	1,59	1,25	1,06	
E2	8	Sí	10	10	10	10	80,00	80,00	80,00	80,00	0,80	0,80	0,80	0,80	
E3	5	No	0,01	0,01	0,01	0,01	0,05	0,05	0,05	0,05	500,00	500,00	500,00	500,00	
E4	9	Nada	0,01	1,2	2,2	3,4	0,09	10,80	19,80	30,60	900,00	7,50	4,09	2,65	
E5	8	Poco	1,2	2,2	3,4	4,4	9,60	17,60	27,20	35,20	6,67	3,64	2,35	1,82	
E6	7	Regular	3,4	4,4	5,6	6,6	23,80	30,80	39,20	46,20	2,06	1,59	1,25	1,06	
E7	4	Todo	7,8	8,8	10	10	31,20	35,20	40,00	40,00	0,51	0,45	0,40	0,40	
E8	4	Todo	7,8	8,8	10	10	31,20	35,20	40,00	40,00	0,51	0,45	0,40	0,40	
E9	8	Mucho	5,6	6,6	7,8	8,8	44,80	52,80	62,40	70,40	1,43	1,21	1,03	0,91	
E10	5	Mucho	5,6	6,6	7,8	8,8	28,00	33,00	39,00	44,00	0,89	0,76	0,64	0,57	
E11	6	Mucho	5,6	6,6	7,8	8,8	33,60	39,60	46,80	52,80	1,07	0,91	0,77	0,68	
E12	7	Poco	1,2	2,2	3,4	4,4	8,40	15,40	23,80	30,80	5,83	3,18	2,06	1,59	
E13	4	Mucho	5,6	6,6	7,8	8,8	22,40	26,40	31,20	35,20	0,71	0,61	0,51	0,45	
E14	8	Mucho	5,6	6,6	7,8	8,8	44,80	52,80	62,40	70,40	1,43	1,21	1,03	0,91	
E15	5	Mucho	5,6	6,6	7,8	8,8	28,00	33,00	39,00	44,00	0,89	0,76	0,64	0,57	
E16	8	Sí	10	10	10	10	80,00	80,00	80,00	80,00	0,80	0,80	0,80	0,80	
E17	6	Regular	3,4	4,4	5,6	6,6	20,40	26,40	33,60	39,60	1,76	1,36	1,07	0,91	
E18	7	Mucho	5,6	6,6	7,8	8,8	39,20	46,20	54,60	61,60	1,25	1,06	0,90	0,80	
E19	2	Mucho	5,6	6,6	7,8	8,8	11,20	13,20	15,60	17,60	0,36	0,30	0,26	0,23	
E20	4	Regular	3,4	4,4	5,6	6,6	13,60	17,60	22,40	26,40	1,18	0,91	0,71	0,61	
E21	6	No	0,01	0,01	0,01	0,01	0,06	0,06	0,06	0,06	600,00	600,00	600,00	600,00	
E22	8	Mucho	5,6	6,6	7,8	8,8	44,80	52,80	62,40	70,40	1,43	1,21	1,03	0,91	
E23	5	Sí	10	10	10	10	50,00	50,00	50,00	50,00	0,50	0,50	0,50	0,50	
							669,00	779,71	908,71	1011,51	2032,15	1130,81	1122,48	1118,62	
Resultado:							2,407	2,827	3,285	3,64994					

Elaborado por: Corral, María Alexandra, Carrillo José Luis.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 3. 5. Intervalos de Dimensión de Éxito

3.4.3 CÁLCULO DEL VALOR FINAL

Tabla 3. 12 Cálculo Valor Final

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:	6,9
-------------------------	------------

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

El Sistema Experto es **viable**, la media general obtenida es mayor a 6, desde la perspectiva de la INCO

CAPÍTULO 4

ADQUISICIÓN DEL CONOCIMIENTO

4.1 INTRODUCCIÓN

La adquisición de conocimientos es el proceso en el que se extraen todos los conocimientos del experto para comprender el Dominio, la información puede ser extraída de varias fuentes útiles para dicho fin. El proceso de adquisición se divide en dos etapas de acuerdo a la metodología IDEAL:

- La extracción del conocimiento, es la fase en la que la información es obtenida de la revisión de manuales, libros u otra forma escrita, la técnica utilizada para este trabajo es el análisis estructural de textos, que tiene por objetivo buscar palabras, términos, conceptos relacionados con el tema a tratar, los términos buscados son dependientes del dominio que pueden ser identificados por varias fuentes de información.
- Educación del conocimiento es la fase en la que el Ingeniero del Conocimiento interactúa con el experto para analizar y profundizar todos los conocimientos necesarios para la resolución del Sistema Experto. La técnica utilizada es través de entrevistas abiertas o estructuradas.

El proceso utilizado para la adquisición del conocimiento se basa en reuniones realizadas a Directivos y a expertos, la planificación de las reuniones aplicadas tiene el siguiente esquema:

Primera reunión con el experto. Para obtener conocimientos generales del dominio del problema y la familiarización de la terminología utilizada.

Segunda reunión. Con el objetivo de obtener respuestas a las preguntas del test de viabilidad, estas reuniones generalmente se las realizan con los Directivos de la empresa, siendo el proyecto de investigación una aplicación

de uso general para Ingenieros de Software, se consideró a dos expertos que conocen sobre el tema y el entorno de la aplicación.

Tercera reunión. Con la finalidad de profundizar conocimientos en el dominio de la aplicación y en las tareas que desempeñará el Sistema Experto, para éstas reuniones se ha considerado a expertos que desempeñan funciones específicas en el área de especificaciones técnicas de requerimientos. En esta etapa se realiza las reuniones que sean necesarias hasta obtener el conocimiento requerido por I.C que le permita estructurar claramente el dominio de la aplicación.

Revisión de documentación existente.

El proceso de extracción se basa en adquirir la documentación existente que le permita obtener conocimiento del Dominio y afianzarse con la terminología (Análisis Estructural). La documentación utilizada para el presente trabajo son Documentos de especificación de Requerimientos aplicados por los expertos en proyectos de Software ejecutados. Además se realizaron revisiones bibliográficas para tratar temas relacionados a Marcos de problemas.

La Metodología IDEAL plantea que para cada reunión se utilice un ciclo de sesiones con el esquema que se presenta a continuación.

1. Preparación de la sesión.

- Información a tratar.
- Amplitud, profundidad, etc.
- Técnica adecuada.
- Preparación de preguntas.

2. Sesión.

- Repaso del análisis de sesiones anteriores
- Explicación al experto de los objetivos.
- Educción.
- Resumen y comentarios del experto.

3. Transcripción.

4. Evaluación y análisis de la Sesión.

- Lectura para obtener una visión general.
- Extracción de conocimientos concretos.
- Lectura para recuperar detalles olvidados.
- Críticas para mejorar por parte del IC.

4.2 Educción del Conocimiento

En esta etapa el conocimiento es adquirido a través de entrevistas realizadas a los expertos.

4.2.1 PRIMERA REUNIÓN CON EL EXPERTO

Preparación de la Sesión

Información a tratar: Un acercamiento al Dominio del problema y a la terminología utilizada.

Amplitud, profundidad: Se trata de dar una visión general del Sistema Experto sin profundizar procesos ni tareas que se ejecutan en la especificación de requerimientos.

Técnica adecuada: Entrevista no estructurada

Preguntas:

1. Defina lo que es requerimiento
2. ¿El término requerimiento lo define igual a requisito?
3. ¿Qué tarea principal debería desempeñar el Sistema Experto?
4. ¿Qué técnica es más utilizada para e licitación de requisitos?
5. Defina lo que es una especificación.
6. ¿A qué tipo de usuario estará dirigido: Usuario común, ingeniero de software?

Transcripción de la primera reunión

Fecha: 28 Diciembre de 2012

Experto: Doctor Edison Espinosa

Lugar: Internet

IC: Ing. José Luis Carrillo, Alexandra Corral

Objetivo: Tener una visión general del Dominio y la terminología utilizada

1.- Defina lo que es requerimiento

Requerimiento de software es una necesidad funcional a ser implementada como parte de un grupo de funcionalidades de un sistema software

2.- El término requerimiento lo define igual a requisito.

Dentro de la ingeniería del software se consideran similares.

3.- Que tarea principal debería desempeñar el Sistema Experto

Un sistema experto debería servir de soporte a la toma de decisiones de un área específica.

4.- Que técnica es más utilizada para e licitación de requisitos.

Encuesta

5.- Defina lo que es una especificación.

Especificación es una(s) característica(s) de un sistema software que debe ser considerada (s) para su implementación.

6.- A qué tipo de usuario estará dirigido: Usuario común, ingeniero de software

Es el usuario que responde a un perfil previamente definido, perfil que permite el uso de un grupo de funcionalidades de un sistema software.

Evaluación y análisis de la primera sesión

Se logró identificar terminología base del Dominio del problema como:

Requerimiento: Requerimiento software es una necesidad funcional a ser implementada como parte de un grupo de funcionalidades de un sistema software.

Se identificó que la terminología requerimiento y requisito pueden ser considerados iguales para lo que en el transcurso del desarrollo del proyecto nos referiremos más a la terminología “Requerimiento”.

Especificación: Especificación es una(s) característica(s) de un sistema software que debe ser considerada (s) para su implementación.

El Sistema Experto debe ser considerado como un soporte para la toma de decisiones en el momento de documentar los requerimientos.

Se consideró un Segundo experto para abordar sobre los mismos temas y definir en forma general las funciones del S.E.

4.2.2 SEGUNDA REUNIÓN CON EL EXPERTO

Preparación de la Sesión

Información a tratar: Un acercamiento al Dominio del problema y a la terminología utilizada.

Amplitud, profundidad: Se trata de dar una visión general del Sistema Experto sin profundizar procesos ni tareas que se ejecutan en la especificación de requerimientos.

Técnica adecuada: Entrevista no estructurada

Preguntas:

1. Defina lo que es requerimiento
2. ¿El término requerimiento lo define igual a requisito?
3. ¿Qué tarea principal debería desempeñar el Sistema Experto?
4. ¿Qué técnica es más utilizada para elicitación de requisitos?
5. Defina lo que es una especificación.
6. A qué tipo de usuario estará dirigido: Usuario común, ingeniero de software.

Transcripción de la segunda reunión

Fecha: 7 de Enero de 2013

Experto: Ing. Juan Peñaherrera

Lugar: Empresa Babel

IC: Ing. Alexandra Corral, José Luis Carrillo

Objetivo: Tener una visión general de la terminología y de la tarea principal del SE

1.- Defina lo que es requerimiento

Es una necesidad básicamente de automatizar algo de nuestro campo

2.- El término requerimiento lo define igual a requisito

La terminología es igual.

3.- Que tarea principal debería desempeñar el Sistema Experto

Tratar de agrupar los requerimientos de acuerdo a su conceptualización hacia donde van dirigidos en el sistema, tratar de agruparlos por procesos y a que procesos están afectando, esa agrupación de requerimientos es una de las partes más difíciles, generar una lista de requerimientos que luego se puedan dividir en módulos.

Hay una parte de la base de datos que dice que dentro de una oración o dentro de un requisito, los sustantivos son entidades y los verbos son los procesos, esta parte es interesante porque le ayuda a describir los requerimientos, y a visualizar las bases de datos y los procesos a nivel de software.

4.- Que técnica es más utilizada para e licitación de requisitos.

Entrevistas

5.- Defina lo que es una especificación

Es delimitar, dividir y también ir desglosando que es lo que yo necesito

Evaluación y análisis de la Sesión

La terminología utilizada coincide con la reunión aplicada al primer experto, teniendo una concordancia en definiciones concretas.

El experto determina que dentro de las funciones que debería desempeñar el S.E está la agrupación de requerimientos por procesos.

4.2.3 TERCERA REUNIÓN CON EL EXPERTO

Preparación de la Sesión

Información a tratar: Determinar si el estudio para la realización del S.E es viable.

Amplitud, profundidad: Se trata de determinar si el Sistema Experto propuesto es viable de acuerdo a la INCO

Técnica adecuada: Entrevista no estructurada

Preguntas:

- 1.- ¿Conoce de proyectos que han fracasado por la mala elicitación de requerimientos en la actualidad?
- 2.- El análisis de un problema se lo resuelve por sentido común o se basan en conocimientos previos.
- 3.- ¿Según su criterio se justifica la creación de un sistema experto para documentar requerimientos?
- 4.- ¿Cuáles son los problemas más frecuentes en la elicitación de requerimientos?
- 5.- ¿Se requiere de experiencia elevada para documentar requerimientos?

Transcripción de la tercera reunión

Fecha: 11 de Enero del 2013

Experto: Doctor Edison Espinosa

Lugar: Internet

IC: Ing. Alexandra Corral, José Luis Carrillo

Objetivo: Determinar características del SE para realizar el estudio de viabilidad

1. ¿Conoce de proyectos que han fracasado por la mala e licitación de requerimientos en la actualidad?

Son pocas las empresas que muestran sus estadísticas de proyectos fracasados. El hecho es que la actividad de elicitación de requisitos es básica e imprescindible para realizar un sistema software.

2. El análisis de un problema se lo resuelve por sentido común o se basan en conocimientos previos.

Es necesario partir de conocimientos previos e ir profundizando en la temática específica, esto debe garantizar un correcto abordaje, análisis y solución del problema.

3. ¿Según su criterio se justifica la creación de un sistema experto para documentar requerimientos?

El dotar de métodos, herramientas, sistemas que den soporte a las personas involucradas en el desarrollo software es fundamental.

4.- ¿Cuáles son los problemas más frecuentes en la elicitación de requerimientos?

El desconocimiento del dominio del problema a resolver.

Las técnicas a emplear para la elicitación

El análisis de la información elicitada

Documentar los resultados (ERS)

Realizar el seguimiento del proceso de elicitación.

5.- Se requiere de experiencia elevada para documentar requerimientos.

Considero que es importante, la experiencia es un factor a tomar en cuenta y que garantiza el éxito de un proyecto software.

6.- Es conveniente justificar las soluciones adoptadas.

Siempre se debe justificar, de hecho se deben establecer procesos formales para explicar las razones por las que se adoptó cierta(s) soluciones.

7.- ¿El tiempo esperado para dar una respuesta es limitado?

Es limitado en tanto y cuanto la característica del sistema lo requiera. Siempre considerando el cronograma de ejecución del proyecto software.

8.- ¿El sistema debe buscar la solución óptima?

El sistema debe resolver el problema, y por ende debe ser óptima.

9.- ¿El sistema a qué tipo de usuarios debería ser dirigido? ¿A personal con poca experiencia en el campo de Ingeniería de Requerimientos o a personal con experiencia en este campo?

Es importante que las personas tengan un alto grado de experiencia

Evaluación y análisis de la sesión 3

La información tomada sirvió para realizar el estudio de viabilidad descrito en el capítulo anterior el mismo que dio como resultado que la realización del sistema es viable de acuerdo a la INCO.

2.2.4 CUARTA REUNIÓN CON EL EXPERTO

Preparación de la Sesión

Información a tratar: Dar un acercamiento más profundo del dominio del problema.

Amplitud, profundidad: Se trata de analizar más detalladamente el dominio del problema para conocer el manejo por parte de los expertos en la elicitación de requerimientos.

Técnica adecuada: Entrevista no estructurada

Preguntas:

1. ¿Podría definir qué son los requerimientos?
2. ¿Podría definir que es una especificación de requerimientos?
3. ¿Qué tipo de información utiliza para la especificación de requerimientos?
4. ¿Utiliza estándares a la hora de especificar los requerimientos?
5. Describa algunas características que debe tener los requerimientos.
6. ¿Qué problemas más comunes ha enfrentado en el momento de especificar los requerimientos?
7. ¿Asigna prioridades a los requerimientos como por ejemplo esencial, deseable?
8. ¿Cuáles son los criterios para evaluar dicha prioridad?

Transcripción de la cuarta reunión

Fecha: 21 de Enero del 2013

Experto: Doctor Edison Espinosa

Lugar: Internet

IC: Ing. Alexandra Corral, José Luis Carrillo

Objetivo: Profundizar conocimientos del dominio de la aplicación

1. ¿Podría definir qué son los requerimientos?

Requerimiento de software es una necesidad funcional a ser implementada como parte de un grupo de funcionalidades de un sistema software

2. ¿Podría definir que es una especificación de requerimientos?

Es un conjunto de características necesarias establecidas en las etapas iniciales del desarrollo software para la implementación de un sistema software

3. ¿Qué tipo de información utiliza para la especificación de requerimientos?

Fuentes primarias (Profesional, libros, procedimientos, normas, análisis de casos, observación).

4. ¿Utiliza estándares a la hora de especificar los requerimientos?

Se debe seguir un modelo específico, es importante seguir un estándar ya que esto va a permitir dar seguimiento al proyecto.

5. Describa algunas características que debe tener los requerimientos.

Completo, no ambiguo, aplicable, probable, trazable.

6. ¿Qué problemas más comunes ha enfrentado en el momento de especificar los requerimientos?

El desconocimiento del dominio del problema.

7. ¿Asigna prioridades a los requerimientos?

Se debe especificar en el estándar que se va a seguir, y directamente va a estar relacionado con el ciclo de vida a adoptar para el proyecto.

8. ¿Cuáles son los criterios para evaluar dicha prioridad?

El grado de impacto que va a tener el requisito sobre la funcionalidad total del sistema.

Evaluación y Análisis de la sesión 3

Se profundizó en conceptos adicionales tales como:

Requerimientos: Requerimiento software es una necesidad funcional a ser implementada como parte de un grupo de funcionalidades de un sistema software.

Especificación de Requerimientos Es un conjunto de características necesarias establecidas en las etapas iniciales del desarrollo software para la implementación de un sistema software.

Se determinó que para la especificación de requerimientos es necesario obtener información de fuentes tales como la de un profesional, libros, procedimientos, normas, análisis de casos, observación. Además de la importancia de seguir un estándar que dé seguimiento al proyecto.

Los requerimientos deben ser asignados por prioridades que dependen del nivel de impacto del sistema.

Conceptos identificados

Requerimiento

Especificación.

Dominio del problema

4.2.5 QUINTA REUNIÓN CON EL EXPERTO

Preparación de la Sesión

Información a tratar: Conocer el proceso de levantamiento de requerimientos

Amplitud, profundidad: Se trata de analizar más detalladamente el dominio de la aplicación para conocer el manejo por parte de los expertos en la elicitación de requerimientos.

Técnica adecuada: Entrevista no estructurada

Preguntas:

1. ¿Qué información se toma de entrada para los datos elicitados?
2. ¿Cómo caracteriza al dominio para aplicar restricciones o limitaciones al sistema?
3. ¿Qué metodología utilizan la empresa para el proceso de levantamiento de requerimientos?
4. ¿Podrías dar un ejemplo de cómo conceptualizas un requerimiento para luego agruparlos por procesos?

Transcripción de la quinta reunión

Fecha: 04 de Febrero del 2013

Experto: Ing. Juan Peñaherrera

Lugar: Oficina Babel

IC: Ing. Alexandra Corral, José Luis Carrillo

Objetivo: Profundizar conocimientos del dominio de la aplicación

1. ¿Qué información se toma de entrada para los datos elicitados?

Para el levantamiento de requerimientos se realizan varias entrevistas, y conforme se va expandiendo estas entrevistas se va prestando mayor atención en los insumos de entrada para los requisitos, la primera entrevista es para conceptualizar de forma general el ámbito y el dominio del problema, luego se realiza un análisis sobre las partes en las que se divide el problema, y se va determinando los sustantivos y verbos más importantes que ayuden a visualizar de mejor manera el dominio de la aplicación.

La tarea empieza con documentos elicitados sean estos en Excel, archivos físicos o impresos que indican acciones de salida o de cálculos que se deban realizar automáticamente o manualmente, es una forma de recopilar la información, o puede darse casos que sean sistemas de virtualización, por ejemplo una gasolinera en donde la información es obtenida mediante elementos físicos, pero surgen las interrogantes en cuanto de lo que se tiene y de lo que no se tiene, por ejemplo contar con la parte física pero no con las interfaces entre el dispositivo y la computadora, el usuario es quien va explicando el proceso como requiere que funcione, un ejemplo del proceso puede ser que el dispositivo imprima una lista requerida, y que ésta sea guardada en una base de datos.

Otro ejemplo puede ser que se requiera unirse a sistemas externos, en este sentido es importante saber qué tipo de información se envía y la que

se recibe externamente de otros sistemas. Toda esta información siempre es obtenida mediante las entrevistas, de todas maneras los insumos pueden ser de diversas formas como se mencionaba anteriormente, incluso existen aspectos que los usuarios lo hacen de una manera natural, es decir en forma verbal y en el momento que se realiza la elicitación hay que determinar cómo registrar o cómo recoger esos insumos.

2. ¿Cómo caracteriza al dominio para aplicar restricciones o limitaciones al sistema?

En sistemas contratados para entidades públicas es necesario definir en primer lugar aspectos técnicos es decir, si el sistema va a interactuar con dispositivos físico o no, incluir en un contrato estos aspectos es importante para evitar problemas posteriores entre las partes. Las limitaciones en este sentido son técnicas

Por ejemplo, el control de personal que se realiza a través de un sistema biométrico, uno de los requerimientos puede ser: “El sistema deberá verificar la asistencia del personal”, este es un requerimiento funcional en el que intervienen aspectos técnicos, por lo que es necesario entonces ir delimitando y definiendo tanto la limitaciones técnicas como funcionales.

En el desarrollo de software a la medida el cliente es el que define qué es lo que utiliza, pero en ocasiones es necesario hacerle entender el rol que desempeña el Ingeniero de Software, porque en muchas ocasiones piensan que el Ingeniero de Software o de Sistemas es el experto en el proceso que ellos realizan y no es así, el Ingeniero de Sistemas es el experto que automatiza lo que los clientes como expertos entienden del proceso. Es necesario entonces ir determinando en base a entrevistas con los usuarios las restricciones o limitaciones que el sistema va a tener.

Por ejemplo, si nos enfocamos a un dominio del problema de tipo bancario o escolástico, no podemos decirle directamente al sistema con cuantos usuarios va a trabajar, es decir si el sistema es solo para 1000 usuarios en el caso bancario o le va a permitir ingresar 1000 alumnos en el caso

escolástico, en este sentido es importante definir y delimitar si ésta es una restricción o es más bien un requerimiento no funcional pero si técnico.

Dentro del desarrollo a la medida, es muy poco probable que se haga un sistema completamente desde cero en el sentido que nada está automatizado, en este caso la institución puede contar con bases de datos o lenguajes de programación ya establecidos, lo que es una restricción de desarrollo ya que la empresa es quien impone las herramientas. Otra restricción puede ser que el presupuesto no alcance, podemos además determinar en qué plataforma desea hacerse el desarrollo, y puede surgir las restricciones que no se cuente con el framework en especial, o que en la base de datos no se pueda hacer un espejo, es necesario ir desglosando cada una de estas restricciones e ir las categorizando. En conclusión las restricciones pueden ser técnicas, de funcionalidad, de almacenamiento, de rapidez o se las puede dividir en restricciones de pre-desarrollo, o de desarrollo.

3. ¿Qué metodología utilizan la empresa para el proceso de levantamiento de requerimientos?

Dentro de la metodología, básicamente se utiliza RUP con UML, en este caso si sale del estándar el requerimiento según la metodología debe ser algo pequeño que defina el alcance en una forma no tan global, es decir no tan detallado de lo que se quiere. Cuando se levanta un requerimiento se lo describe en un párrafo o dos párrafos y luego se hace el caso de uso extendido, los diagramas de secuencia basándose en este caso en UML.

La metodología de Babel para levantar los requisitos utiliza plantillas diseñadas en Word ya sea para las entrevistas, para el diseño de especificación de software, para el control de cambios sobre los requerimientos. Este último se lo realiza en base a minutas. Los procesos que tiene la metodología se basan en la conceptualización, planeación, desarrollo, control y cierre del proyecto; estos son los pasos básicos que se sigue y en los que hay que ir paso a paso cumpliendo de acuerdo a un cronograma.

Dentro de la conceptualización se realizan las primeras entrevistas SOW que es la Especificación del Diseño de Software, donde se define la arquitectura, la programación, las bases de datos, las interfaces externa e internas. Luego viene la etapa de planeación, pero entre lo que es la conceptualización y planeación se desarrolla el levantamiento de requerimientos el cual tiene que estar controlado por el Project Manager y el grupo que realiza este levantamiento, el control se basa en las horas utilizadas, cantidad de trabajo realizado, y si el requerimiento está bien especificado o no.

Dentro del levantamiento de los requerimientos se debe realizar las pruebas de validación que en buena teoría deberían hacerse a través de los casos de uso extendido, diagramas de secuencias que exige el RUP y UML lo cual no siempre es así y debemos obviar estos diagramas, lo que generalmente se hace para validar el requerimiento es utilizar prototipos que son básicamente pantallas que describen el proceso, este proceso se lo explica en forma verbal al usuario como va a quedar inicialmente, de haber cambios se generar una orden de trabajo y se va validando el requerimiento levantado, si no satisface hay que ir refinando nuevamente con los diagramas mencionados anteriormente, esto en cuanto a la parte de conceptualización, con esta base se entra a la planeación, el control y la especificación del requerimientos, luego viene un cronograma ya más asentado, la metodología también indica cómo va hacer el cronograma ya que no son solo los desarrolladores los que intervienen en el proyecto también son los testers, el arquitecto las horas del Project Manager, y las reuniones planificadas.

Sin embargo no todos los proyectos deben adaptarse a la metodología utilizada por la empresa sino a la que el cliente necesita.

4. ¿Podría dar un ejemplo de cómo conceptualizas un requerimiento para luego agruparlos por procesos?

Hay que hacer un levantamiento no solo del requerimiento sino también del proceso. Por ejemplo un requerimiento dice “El sistema deberá crear una lista de clientes, de habitaciones, de proveedores y de empleados” para el caso de un sistema hotelero, este es un requerimiento en el que hay que ir plasmando no solo su definición como tal, sino también el proceso. En los procesos se analizan todos los sustantivos que se encuentran y la relación que existe entre ellos, la única forma de agrupar es luego de haber levantado algunos requerimientos, la agrupación de estos requerimientos van a ser los procesos que luego se constituirán en módulos ya del sistema.

Luego para cada subdivisión se va refinando, porque puede darse el caso que un requisito se encuentre duplicado en varios módulos, por ejemplo, en el requerimientos mencionado anteriormente, se detecte que la misma pantalla para clientes se encuentre en más de un módulo, en todo caso, si se va agrupando desde el inicio a los requerimientos, sería más conveniente ir analizándolos conjuntamente con el experto sobre los temas que tienen similitud y dependiendo de esos temas ir definiendo el dominio del problema para a partir de ahí realizar las agrupaciones.

En base de datos se debe partir de una conceptualización general estableciendo las relaciones entre entidades, entonces en las primeras entrevistas deberíamos llegar a esa conceptualización para ver qué procesos pertenecen a cada agrupación y dentro del proceso ir viendo que requerimientos se van a utilizar.

Evaluación y análisis de la sesión 4.

Se analizó sobre las entradas más utilizadas al momento de realizar el levantamiento de requerimientos que son por lo general en base a entrevistas, y a la documentación proporcionada por los usuarios que dependiendo del tipo de sistema pueden ser: Archivos físicos o impresos,

Sistemas que recojan información a través de medios externos o información tomada del sistema informático ya existente.

Con este análisis podemos establecer una de las entradas del sistema experto que a través de los primeros insumos elicitados por los usuarios se pueda dar una aproximación al análisis del dominio.

Las restricciones deben ser identificadas conjuntamente con el usuario y personas encargadas de levantar los requerimientos, estas pueden ser categorizadas como restricciones de pre_desarrollo y de desarrollo. Las restricciones de pre-desarrollo pueden ser por presupuesto y alcance. Las restricciones de desarrollo pueden ser plataformas, lenguajes de programación, bases de datos, etc.

Cuando se realiza la conceptualización, se establece la Especificación del Diseño de Software en donde se define la arquitectura en la que va a estar, la programación, las bases de datos, interfaces externas e internas en este caso utilizadas para la especificación de requerimientos técnicos.

Para el análisis de sistemas externos es importante determinar el trato de la información que se envía y la información que se reciba.

La priorización de los requerimientos es determinada por las necesidades más relevantes del problema.

Los procesos que utiliza la empresa son: Conceptualización, planeación, desarrollo, control y cierre. El análisis se limita solo a la conceptualización. Esta entrevista permitió afianzar más los conocimientos del dominio y se pudo obtener elementos claves los cuales se utilizarán para el Sistema Experto.

Conceptos identificados:

Dominio del problema.

Requerimiento funcional.

Restricciones
Arquitectura
Interfaces externas

4.2.6 SEXTA REUNIÓN CON EL EXPERTO

Preparación de la Sesión

Información a tratar: Profundizar conocimientos de las tareas realizadas por el experto para la aplicación.

Amplitud, profundidad: Se trata de analizar más detalladamente el dominio de la aplicación para conocer el proceso que realiza el experto para la especificación de requerimientos

Técnica adecuada: Entrevista no estructurada

Preguntas:

- 1.- ¿Que documentos de entrada aplica para la especificación de los requerimientos?
- 2.- ¿Utiliza estándares para documentar los requerimientos?
- 3.- ¿Qué proceso realiza a la hora de especificar los requerimientos?
- 4.- ¿Asigna prioridades a los requerimientos y de que depende esta priorización?
- 5.- ¿Podría dar un ejemplo de cómo identificar a un dominio de la aplicación utilizando marcos de problemas?
- 6.- ¿Podría definir qué parámetros son considerados en una plantilla de requerimientos funcionales en el momento de documentar?
- 7.- ¿Cómo documenta a los requerimientos no funcionales y en base a que los determina en un sistema?

Transcripción de la sexta reunión

Fecha: 01 de Marzo del 2013

Experto: Ing. Juan Carlos Márquez

Lugar: Oficina del experto

IC: Ing. Alexandra Corral

Objetivo: Profundizar conocimientos del dominio de la aplicación con un segundo experto

1. ¿Qué documentos de entrada aplica para la especificación de los requerimientos?

Entrevistas con los usuarios finales.

Especificaciones funcionales del sistema.

Especificaciones técnicas del sistema.

2.- ¿Utiliza estándares para documentar los requerimientos?

El estándar más utilizado es el IEEE830-1998

3.- ¿Qué proceso realiza a la hora de especificar los requerimientos?

Para la generación de requerimientos es recomendable seguir los siguientes pasos:

Establecer un acuerdo con el cliente sobre los requerimientos dentro del proyecto de software esto nos permite estimar, planear, ejecutar y monitorear el proyecto de desarrollo de software a través de todo su ciclo de vida.

Armar un primer manual de usuario, esto nos ayuda a que el cliente visualice la funcionalidad del requerimiento dentro del sistema logrando que lo entienda de mejor manera y con esto proveer de un escenario de entendimiento mucho más amplio para realizar cambios sobre la marcha.

Utilización de una metodología detallada de definición de los requerimientos de usuario.

Organizar reuniones destinadas a obtener la máxima información posible acerca de los requerimientos de los usuarios.

Traducir las necesidades en requerimientos del proyecto.

Comprender el perfil y entorno del usuario.

4.- ¿Asigna prioridades a los requerimientos y de que depende esta priorización?

Si, la priorización depende del tipo de requerimientos que se reconoce después de haberlo documentado, en sí esta clasificación se da de la siguiente manera:

Esenciales.- Son los primeros en desarrollarse ya que forman parte del núcleo de nuestro sistema.

Condicionales.- Son los últimos en aplicarse porque están sujetos a variables ya sea del negocio o del funcionamiento en sí.

Opcionales.- Se los contempla al final y su implementación o no implementación no afecta en nada al producto final.

5.- ¿Podría dar un ejemplo de cómo identificar a un dominio de la aplicación utilizando marcos de problemas?

Para la identificación de los dominios y encajar éste en un marco de problema se debe realizar preguntas al usuario con lenguaje natural de las necesidades a cumplir en el requerimiento. Con esto podemos compararlo con cada marco de problema y cotejar si la necesidad se identifica con uno u otro marco. Por ejemplo se puede hacer preguntas sobre el dominio como:

¿El sistema requiere controlar dispositivos externos?

Ejemplos

Pistolas de códigos de barras, Microcontroladores, PLCS

¿El sistema tendrá la funcionalidad de crear documentación a partir de la información de entrada?

¿El sistema requiere la funcionalidad de operar con datos de entrada y en la salida obtener la respuesta de estos cálculos?

6.- ¿Podría definir qué parámetros son considerados en una plantilla de requerimientos funcionales en el momento de documentar?

Los parámetros a considerar dentro de la documentación de requerimientos para obtener una información lo más apegada a las necesidades del usuario pueden ser:

Descripción Breve del requerimiento

Versión y fecha del sistema

Precondiciones: Se especifican los requerimientos que se tienen que cumplir para llegar a este.

Condición de éxito: Se especifica que sucede si la ejecución del requerimiento fue exitosa

Condición de fracaso: Se especifica que sucede si el requerimiento no se realiza.

Actores: Quien está a cargo del requerimiento.

Evento Disparador evento que se debe cumplir para la realización del evento

Secuencia ordinaria: Especifica los pasos del requerimiento.

Post Condiciones: Especifica los eventos que se desencadenan una vez cumplido el presente.

Frecuencia: Se especifica la frecuencia con la que se realiza el requerimiento.

Costo del requerimiento: Costo estimado para el requerimiento

Comentarios: Algún comentario adicional para el requerimiento.

7.- ¿Cómo documenta a los requerimientos no funcionales y en base a que los determina en un sistema?

Los requerimientos no funcionales se ocupan del cómo realizar el sistema, para su documentación se toman en cuenta los siguientes parámetros:

Restricciones de hardware y software

Interconexión con otros sistemas.

Disponibilidad.

Políticas de adquisición de software.

Requisitos de seguridad.

Portabilidad

Fiabilidad

Mantenimiento.

Evaluación y análisis de la sesión 5.

Esta sesión fue muy fructífera ya que se pudo determinar gran parte del conocimiento del experto para realizar el Sistema. Se identificó lo siguiente:

El análisis del dominio se lo puede realizar a través de preguntas claves al usuario por ejemplo.

¿El sistema controla algún tipo de dispositivo externo?.

¿El sistema tendrá la funcionalidad de crear documentación a partir de la información de entrada?

¿El sistema requiere la funcionalidad de operar con datos de entrada y en la salida obtener la respuesta de estos cálculos?

Con este análisis podemos definir ya una entrada al sistema para determinar el tipo de Marco de problema. El análisis de la especificación del requerimiento funcional está dado en base a varios parámetros que son especificados en el documento.

Se determinó que la priorización de los requerimientos es en base al impacto que tiene en el sistema, se los puede clasificar en esencial, condicional y opcional dependiendo de la importancia dada al requerimiento al momento de la aplicación.

Se definió características que el software debe tener a la hora de documentar así como los requerimientos no funcionales aplicados en un documento de requerimientos.

Esta base permitirá establecer parámetros aplicados al documento de especificación de requerimientos.

Conceptos identificados

Actores

Restricciones de hardware y software

Portabilidad del sistema.

Interconexión con otros sistemas.

Disponibilidad.

Requisitos de seguridad.

Portabilidad

Fiabilidad

Mantenimiento.

4.3 EXTRACCIÓN DEL CONOCIMIENTO

En el proceso de extracción del conocimiento nos hemos valido de textos bibliográficos, artículos, papers, tesis de Postgrados así como de documentación utilizada en algunas empresas de desarrollo de software para obtener mayor información del dominio y poder establecer las primeras sesiones con el experto.

En este tratado se ha podido reafirmar conocimientos sobre Ingeniería de Requerimientos, se ha aprendido teorías sobre Marcos de problemas, se ha conocido formas de documentar requerimientos de acuerdo a plantillas utilizadas por empresas de desarrollo de software siguiendo el estándar IEEE std 830. Estos conocimientos han permitido poder conceptualizar el dominio del problema.

La técnica utilizada para la extracción del conocimiento es a través del Análisis estructural de textos.

4.3.1 ANÁLISIS ESTRUCTURAL DE TEXTOS

La información obtenida para el análisis estructural de textos son plantillas diseñadas en Word por la empresa Babel utilizadas para la documentación

de los ERS (Especificación de Requerimientos de Software), la información utilizada es extraída a partir del estándar IEEE std 830, los cuales permitieron obtener datos para analizar los parámetros empleados en la documentación y terminología utilizada.

Se obtuvo información de revisiones bibliográficas sobre temas relacionados a Marcos de Problemas propuesto por el autor Jackson que permite dar una aproximación a tipos de dominios de acuerdo a las entradas de los requerimientos.

De esta revisión se obtuvo información relevante para el S.E de acuerdo a lo mencionado por Jackson en la que determina que para realizar una aproximación al análisis del Dominio se puede hacer a través de cinco tipos de Marcos de problemas Elementales como son:

Marco de Problema de Información.

En la que se analiza las entradas especificadas por los usuarios y se determina si su salida es solo de consulta o de información requerida por el usuario, entonces podemos deducir que es un Marco de Problema de Información.

Marcos de Control

Si el software controla algún tipo de dispositivo haciendo que cumpla una serie de reglas, entonces es un Marco de control.

Marco de problema de Workspiece

Cuando los usuarios requieran 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, entonces deducimos que es un tipo de Marco de problema de control.

Marcos de problema de Transformación

Si se determina que las entradas producen cambios o transformaciones en sus salidas a través de cálculos entonces podemos decir que es un Marco de problema de transformación.

Marcos de problemas de conexión

Los marcos de problemas de conexión son los que tienen relación con varios dominios.

Conceptos

Marcos de problema

Dominio

Marco de Problema de Información.

Marco de Problema de control.

Marco de Problema de Workspiece.

Marco de Problema de transformación.

Marco de Problema de Conexión.

Análisis de Dominio

Todos los sistemas Informáticos parten de un análisis, que constituye la fase inicial para todo proceso de desarrollo de Software, entender y representar un Dominio como base para la especificación de requerimientos constituye la etapa más importante en el proceso de desarrollo de Software.

La propuesta está inicialmente encaminada al estudio del análisis del dominio del problema a través de un modelo conceptual en las que determine características comunes. La conceptualización es representada por conceptos del dominio de la aplicación. Una vez identificados los conceptos o entidades se establece los atributos y las relaciones, los mismos que serán parte de procesos comunes que por lo general son parte de los sistema de información, como son los ABC (Altas, bajas y cambios).

El análisis de dominios es recomendable según Díez Cebollero(2009) como una herramienta útil para el desarrollo de sistemas en dominios estables, en

los cuales la información del contexto puede ser fácilmente deducible a partir de expertos de dominio.

El análisis del dominio para el presente trabajo se lo da de una manera genérica, cuyo objetivo es dar solución no solo a problemas específicos sino aquellos con características comunes.

4.3.2 EVALUACIÓN DEL ANÁLISIS ESTRUCTURAL DE TEXTOS

El análisis estructural de textos, ha permitido al Ingeniero de conocimientos a través de la documentación utilizada e investigada realizada llegar a un consenso con el experto el mismo que abalizado la información extraída dando su aceptación.

4.4 EVALUACIÓN DE LA ADQUISICIÓN DE CONOCIMIENTOS

El razonamiento de los expertos comienza con el análisis de la información extraída en el proceso de elicitación para la obtención de los requerimientos que por lo general son expresados en lenguaje natural. El análisis de un problema comienza por la conceptualización, que es la representación gráfica del dominio de la aplicación con sus entidades, relaciones y atributos.

Luego realiza una análisis de las restricciones del sistema sean éstas técnicas o de desarrollo, analiza la existencia de sistemas externos o sistemas existentes en los que determina la información enviada y recibida, ésta información que analiza el experto determina que si son sistemas externos la información puede ser por tramas, archivos planos, entre otros.

Establece los tipos de usuarios que manejaran el sistema y las tareas que realizaran los usuarios en el sistema.

Con este análisis tanto de la delimitación de restricciones como el análisis de conexión con otros sistemas y de usuarios, empieza la descripción de los requerimientos no funcional que constituyen ser las características propias de cada sistema.

En conclusión, las reuniones realizadas con los expertos sirvieron para obtener conocimientos sobre las tareas que realizan en la especificación de los requerimientos. Las reuniones se las realizó con varios expertos independientes de una sola institución, de las cuales se analizó información relevante que determinen el funcionamiento del Sistema Experto. Esta información ha sido evaluada por los expertos dando su aceptación a los conocimientos adquiridos.

La información extraída en esta fase es utilizada para la conceptualización del Conocimiento aplicada en etapas posteriores.

CAPÍTULO 5

CONCEPTUALIZACIÓN

5.1 INTRODUCCIÓN

En la etapa de Conceptualización el Ingeniero del Conocimiento (I. C.) tiene un entendimiento del Dominio a través de la Adquisición del Conocimiento realizada en la fase anterior, sin embargo esta etapa no es secuencial de una fase anterior ya que en el proceso de conceptualización se sigue Adquiriendo conocimientos del experto quien va refinando y organizando la información, en esta etapa el I.C. estructura y modela los conocimientos que le permitan determinar el comportamiento del Sistema Experto (S.E). Un SE, es un software que imita el comportamiento de un experto humano en la solución de un problema.

La fase de conceptualización permite a demás tomar la terminología adecuada que será utilizada a lo largo del proyecto. Para la Conceptualización se utilizan formalismos de representación del conocimiento de la metodología IDEAL Gómez et. al. (1997). Esta etapa conlleva un proceso de Análisis de la información y de los conocimientos y de una Síntesis de los mismos.

El objetivo del análisis es la identificación de los conocimientos estratégicos, tácticos y fácticos, para lo cual se aplican las siguientes técnicas:

- Conocimientos estratégicos: Árbol de descomposición funcional del problema
- Conocimientos tácticos: Fórmulas, árboles de decisión, tablas de decisión, seudoreglas.
- Conocimientos fácticos: Glosario de términos, Diccionario de conceptos, Tabla de Conceptos-atributo-valor, Modelo relacional entre conceptos.

El objetivo de la síntesis es establecer cómo dichos conocimientos del análisis son transformados en el Modelo Estático y Dinámico que conforman el Modelo Conceptual.

5.2 ANÁLISIS DE LA INFORMACIÓN Y DE LOS CONOCIMIENTOS.

En esta etapa se toma la información de acuerdo a tres tipos de conocimientos categorizados como estratégicos, tácticos y fácticos.

5.2.1 Conocimientos fácticos. Especifica lo que es, o se cree que es verdad acerca del mundo en general y acerca del caso particular para el cual se está ejecutando la tarea. En esta etapa se utiliza: **Glosario términos** para identificar la terminología utilizada a lo largo del proyecto de tal manera que no exista ambigüedades y se pueda establecer un lenguaje común entre el I.C y el Experto, **Diccionario de Conceptos** identificando los conceptos de más alto nivel, en el que se especifique para cada concepto su utilidad o función, sinónimo, acrónimo y valores de los atributos que lo definen, en el diccionario de conceptos se establece la terminología utilizada para el proyecto. **Tabla Concepto- atributo-valor** parte del diccionario de Conceptos y es en donde se definen las características de los conceptos a lo largo del proyecto. Una vez identificados los conceptos se establece la relación entre ellos a través del **Modelo Relacional de conceptos**.

A continuación se presenta el desarrollo para cada una de las técnicas utilizadas en la etapa de análisis.

5.2.1.1 GLOSARIO DE TÉRMINOS

La tabla 5.1 describe la terminología utilizada a lo largo del proyecto.

Tabla 5. 1 Glosario de términos

Término	Significado
Requerimiento	<p>Condición o necesidades de los usuarios para resolver un problema.</p> <p>Condición o capacidad que debe poseer un sistema para satisfacer una especificación impuesta (Committee 1990)</p>
Dominio de aplicación	Parte del contexto del problema que será ejecutado en una máquina real. (Kim, Park et al. 2008)
Dominio del problema	Definición del contexto general del problema en el que los usuarios describen sus necesidades
Marco de problema	<p>Los marcos de problema caracterizan clases de problema que comúnmente ocurren como subproblemas de otros problemas más grandes y reales.</p> <p>Un marco de problema hace que sea posible hablar útilmente acerca de los métodos de solución para los problemas que se ajustan al marco. Jackson(1999)</p>
Marco de Problema de Información	Se refiere a requerimientos de consulta, donde se muestra parte del mundo real en la que los usuarios obtienen información. Jackson (2010)
Marco de Problema de Control	Se define a problemas en donde el software define cierto dominio que se comporte de una manera determinada haciendo que cumpla una serie de reglas de comportamiento. Jackson (2010)
Marco de problema de Transformación	Significa que el software genere datos de salida mapeados contra datos de entrada de acuerdo a reglas específicas. Jackson (2010)

Marco de problema Workpieces	Se define sobre un problema donde un usuario puede crear y editar artefactos intangibles que existen dentro del propio software que lo resuelve, tal como texto, gráficos o estructuras similares. Jackson (2010)
Marco de Problema de Conexión	Cuando se crean varios dominios en el que compartan información a través de fenómenos compartidos. Jackson (2010)
Información de Entrada	Son todos los insumos proporcionados por los usuarios que sirvan de entradas al software estos pueden ser listados, informes, reportes, documentación de entrevistas a usuarios, encuestas, cuestionarios a usuarios, documentos de elicitación.
Especificación	Es un documento que define, los requisitos en una forma completa y verificable. Define además el diseño, el comportamiento u otras características de un sistema o componente de un sistema
Documento de Requerimientos	El documento que describe los servicios y funciones que el sistema debe proveer, las restricciones bajo las cuales el sistema funciona, restricciones sobre las propiedades emergentes del sistema, definiciones de otros sistemas con los cuales el sistema integra información acerca del dominio de aplicación.
Requerimiento Funcional	Los requerimientos funcionales definen el comportamiento específico del software.(Committee and Board 1998)
Requerimiento No funcional	Define las características del Sistema como la fiabilidad, la respuesta en el tiempo, la capacidad de almacenamiento, seguridad.
Restricciones	Se refiere a limitaciones que se impone

	sobre el desarrollo del producto
Interfaces Externas	<p>Describen los requerimientos de interfaz con otros sistemas (hardware y software) e interfaces de comunicaciones.</p> <p>Se determina los archivos que se usan para transmitir información a otro sistema.</p>
Seguridad	Especifica los factores que podrían proteger el sistema de accidentes y accesos maliciosos, ejemplo: uso de técnicas de criptografía.
Conectividad	Es la interacción del Sistema con otros Sistemas externos o internos.
Actores	Son las personas que van a participar en el desarrollo del Software.
Entidades	Son conceptos que son parte del dominio de la aplicación.
Atributos	Son características propias de cada entidad.

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

5.2.1.2 DICCIONARIO DE CONCEPTOS

La tabla 5.2 describe los conceptos de más alto nivel.

Tabla 5. 2 Diccionario de conceptos

CONCEPTO	FUNCIÓN	SINÓNIMO	ATRIBUTO	DERIVADO DE
Proyecto	Es el que contiene la información necesaria para la cual se busca solucionar un problema	Sistema	<p>Nombre</p> <p>Nombre del sistema</p> <p>Objetivo</p> <p>Descripción</p> <p>Def. Acro. Abrev</p>	De las peticiones de la empresa

Marco de Problema	Permiten analizar dominios que caractericen clases de problema que comúnmente ocurren como subproblemas de otros problemas más grandes y reales.	MP (Marco de Problema)	Descripción Tipos	Parte de los requerimientos y de la descripción del Dominio
Entidades	Permite determinar los conceptos de más alto nivel que son partes del dominio de la aplicación.	Conceptos	Nombre. Descripción	Parte del dominio de la aplicación.
Requerimiento funcional	Lo que se desea que se produzca en la máquina una vez especificado.	Requisito funcional	Descripción tipo	De las necesidades del usuario.
Restricciones de diseño	Determina las limitaciones de diseño	Limitaciones	Lenguajes Bases de datos Alcance	De lo que la empresa requiere o dispone para el desarrollo del sistema
Interfaces externas	Ve la conectividad de sistemas externos o internos para el tratado de la información	Conexión con sistemas externos	Tipo de sistemas Tipo de Información	De los sistemas con los que transfiere información
Requerimiento no funcional	Determinar las características del sistema	Requisito no funcional	Alcance Bases de datos Capacidad Disponibilidad Seguridad Sensibilidad	De las características del software. De las necesidades del sistema, y de los

				requerimientos
Documento de requerimientos	Es la salida del Sistema Experto en el que contiene el detalle técnico y los requerimientos Funcionales y no funcionales	Especificación de Requisitos	Nombre de archivo Versión Usuario que género	

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

5.2.1.3 Tabla Concepto/Atributo/valor

Parte de la tabla del Diccionario de Conceptos, adicionando las características propias por concepto y su valor. Como lo describe la tabla 5.3

Tabla 5. 3 Tabla Concepto/Atributo/valor

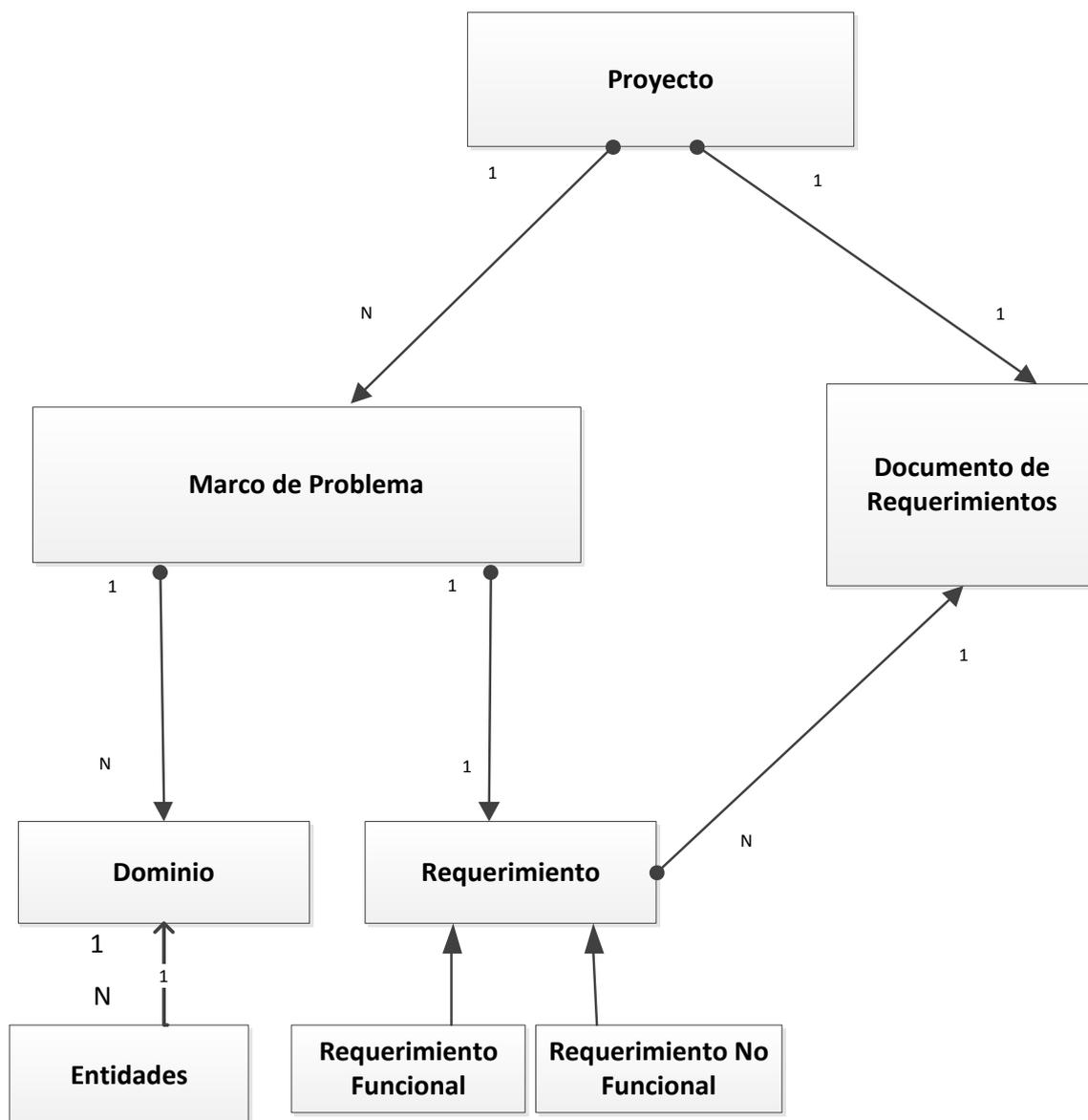
Concepto	Atributo	Valor
Proyecto	Nombre	Texto Nombre corto del proyecto
	Objetivo	Texto Describe el objetivo del proyecto
	Nombre del Sistema	Texto completo del proyecto
	Descripción	Texto Descripción detallada del proyecto
Marco de problema	Descripción	Describe el ámbito del marco de problema
	Tipos	Describe el tipo de problema Problemas de Información, Problemas de Conexión, Problemas de transformación, Problemas Workpieces, Problemas de control.
Dominio	Entidades	Texto que describe a los conceptos que son parte del dominio.

	Atributos	Texto que describen las características de las entidades
Requerimiento funcional	Descripción	Texto Descripción del Requerimiento
	Tipo de Requerimiento	Requerimiento de consulta Requerimiento de reglas de comportamiento Mapeo Operación sobre dominios creados Correspondencia entre dominios Altas Bajas Cambios
Restricciones de diseño	Lenguaje	Información para el diseño de la solución sobre políticas de adquisición <ul style="list-style-type: none"> • Software libre • Software propietario • Ambas
	Bases de datos	<ul style="list-style-type: none"> • Si • No
Alcance del sistema	Alcance de acceso hacia el sistema	<ul style="list-style-type: none"> • Utiliza solo dentro de la empresa • Utiliza desde Internet con computador de escritorio • Utiliza desde Internet utilizando dispositivos como tablets, celulares etc.
Requerimiento no funcional	Interfaces Externas	Tipos de Sistemas <ul style="list-style-type: none"> • Cliente-servidor • Distribuido • Otro
		Descripción de o los aplicativos de los cuales se tomará y se extraerá información. <ul style="list-style-type: none"> • Servicios Web • Archivos planos

		<ul style="list-style-type: none"> • Tramas • Clonando estructuras de BDD (Base De Datos) • Consultas
	Disponibilidad	<p>Parámetros para medir la disponibilidad con la cual contará el aplicativo</p> <ul style="list-style-type: none"> • 1-8 horas • 8-más horas
	Seguridad	<p>Nivel de sensibilidad del aplicativo en escala de 1 al 4</p> <ul style="list-style-type: none"> • 1 • 2 • 3 • 4
	Concurrente	<p>Cuantos usuarios pueden estar conectados simultáneamente.</p> <ul style="list-style-type: none"> • 1-20 • 20-50 • 50-más
	Capacidad	<p>Cantidad de información que maneja el sistema para un aproximado de un año.</p> <ul style="list-style-type: none"> • 1-5 GB • 5-10 GB • 10-más GB
Documento de Requerimientos	Nombre del documento Partes	<p>Nombre del documento</p> <p>Requerimientos funcionales, No Funcionales, Analisis del dominio</p>

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

El modelo relacional de Conceptos como indica la Figura 5.1 sirve para ver las relaciones existentes entre los conceptos.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 5. 1. Modelo Relacional

Si se analiza la figura 5.1 se puede determinar que un Proyecto tiene una relación de uno a varios Marcos de Problemas.

Un Marco de Problema puede tener varios Dominios y un tipo de Requerimiento. Los Requerimientos pueden ser a su vez funcionales y no funcionales.

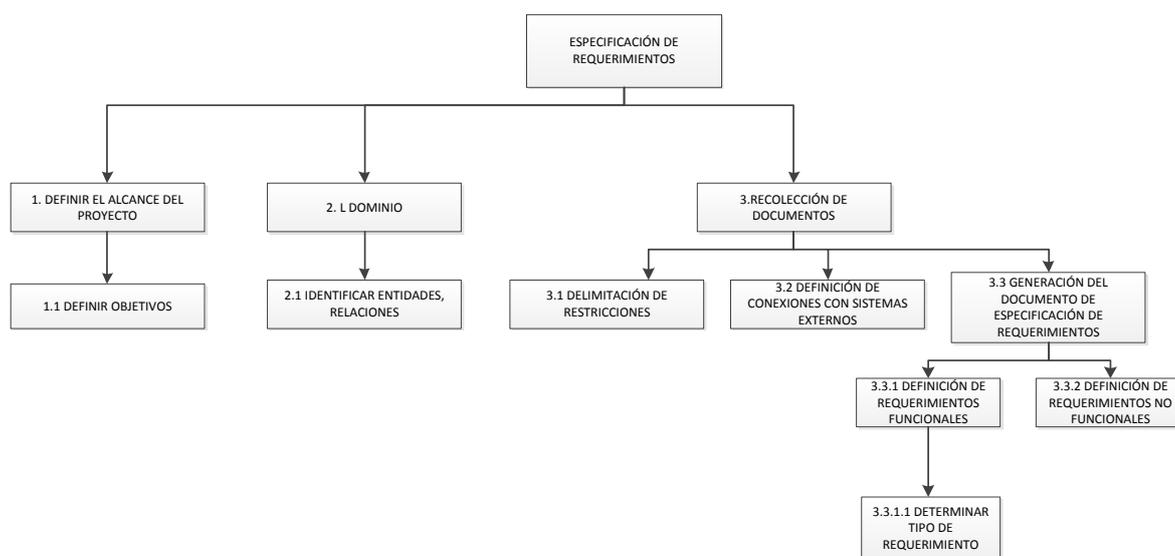
El Documento de Requerimientos contiene varios requerimientos que describen la funcionalidad del Sistema y su entorno de desarrollo.

5.2.2. CONOCIMIENTOS ESTRATÉGICOS

Los conocimientos estratégicos consisten en identificar las funciones de resolución del experto y el flujo de control que regirá el funcionamiento del sistema experto. Los pasos que se siguen para representar los conocimientos estratégicos son:

- Representación gráfica en forma de árbol de los pasos modulares que sigue el experto para completar la tarea. La Figura 5.2 describe el árbol de descomposición funcional utilizado para el presente proyecto.
- Descripción de los módulos de acuerdo a la propuesta de Gómez [Gómez, A. y otros 1997]: en la que define los pasos de Alto Nivel, subpasos de la Tarea y subpasos de bajo nivel.

Estos pasos deben ser descompuestos en varios niveles, que corresponden a la tarea que el experto realiza. Se sigue una secuencia ordenada de los pasos que debe ejecutar el sistema experto identificando la secuencia de entrada, el razonamiento y las acciones de salida.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 5. 2. Árbol de descomposición Funcional

5.2.2.1 Pasos de alto nivel

1. Definir el alcance del Proyecto
2. Análisis del dominio.
3. Recolección de Documentos

Paso 1: Definir el alcance del Proyecto.

En este paso el experto realiza un análisis de la información proporcionada para determinar el alcance y definición de las especificaciones.

Subpasos:

1.1 Definir objetivos

Entradas:

Las entradas para este paso pueden constituirse en información proporcionada por los usuarios finales, documentos, informes, reportes de sistemas externos, reportes de sistemas existente.

Razonamiento

El experto revisa toda la información proporcionada para realizar una primera aproximación a la definición y alcance del sistema a desarrollar, en este paso puede establecer un acuerdo con los clientes determinando los objetivos del sistema, usuarios, subsistemas y toda la información necesaria para la especificación.

Salida

Objetivos del sistema que se pretende desarrollar.

Paso 2: Análisis del Dominio.

Propósito:

El experto analiza la descripción del problema con el propósito de capturar e identificar información que visualicen al dominio de la aplicación a través de las entidades y relaciones.

Subpasos:

1.- Identificar entidades y relaciones

Entrada:

La entrada es la información proporcionada por el usuario en lenguaje natural a través de la descripción detallada de la aplicación.

Razonamiento:

El experto realiza un análisis del dominio para tener una comprensión del problema, la información de entrada es la descripción del sistema proporcionada por el usuario en un lenguaje natural, aquí el experto junto con el usuario identifica entidades y relaciones existentes de la descripción. Este es el primer paso de análisis de requerimientos.

Salida:

Modelo conceptual

Paso 3: Recolección de Documentos

Propósito.

En este paso el experto analiza la información elicitada sea a través de entrevistas o documentos proporcionados por los usuarios en los que determina y establece el entorno del sistema, definiendo las restricciones técnicas y de desarrollo, analiza la conexión con otros sistemas externos para luego analizar la funcionalidad y características del sistema a través de los requerimientos funcionales y no funcionales que generen el documento de especificaciones.

Subpasos:

1. Delimitación de restricciones
2. Definición de conexiones con sistemas externos

3. Generación del Documento de especificación de requerimientos

Entradas:

Documentos, entrevistas que definan la funcionalidad y características del sistema, reportes de conexiones con sistemas externos.

Salidas:

Documento de Especificación de Requerimientos

5.2.2.2. Subpasos de la tarea

1.1: Definir objetivos

Entrada

Documentos obtenidos de la elicitación

Razonamiento

Aquí el experto analiza el objetivo propuesto por el usuario y la razón principal para la construcción del sistema.

La definición de objetivos le permitirá dar un primer acercamiento al problema, el mismo que se irá profundizando en análisis posteriores. Esta definición le ayuda al experto a determinar el alcance del sistema.

Salida

Objetivos del sistema

2.2: Identificar Entidades y relaciones

Entrada

Descripción del sistema, Documentos elicitados.

Razonamiento

El experto analiza la documentación elicitada y los objetivos del sistema que servirán para identificar y desglosar el problema en los marcos de problemas elementales, a través de los requerimientos previamente elicitados.

Identificado el marco del problema el experto es capaz de emitir sugerencias a especificaciones técnicas del requerimiento.

Salida

Descripción del Marco de problema

3.1: Delimitación de Restricciones

Entrada

Documentos elicitados, entrevistas con los usuarios

Razonamiento

Aquí el experto analiza conjuntamente con el usuario las restricciones del sistema, luego clasifica en restricciones de predesarrollo y de desarrollo, con esta información el experto define el entorno de desarrollo en el que el sistema se desarrollará. Entre las restricciones que puede definir están la política de adquisición de software que la empresa realiza en las que puede decidir por software libre o propietario. También el experto define el alcance de los usuarios finales, si los usuarios usarán el sistema dentro de la empresa, vía internet a través de dispositivos móviles o por computadoras de escritorio, define además la utilización de bases de datos, además de los lenguajes de programación utilizados para el desarrollo.

Para el experto definir inicialmente estas limitaciones es importante ya que con estos datos puede conocer con lo que cuenta para las especificaciones técnicas.

Salida

Definición de restricciones y limitaciones del sistema.

3.2: Definición de conexiones con Sistema externos

Entrada

Documentación elicitada

Razonamiento

El experto analiza la conexión del futuro sistema con otros productos externos. Si el sistema es totalmente independiente también realiza una especificación. Si se está definiendo un sistema que es parte de un sistema mayor, se debe relacionar los requisitos del sistema mayor con la funcionalidad del producto y se identificarán las interfaces existentes entre estos sistemas.

El experto analiza si los sistemas requieren compartir información, esta compartición de información puede ser a través de servicios Web, archivos plano, clonando estructuras de datos, tramas, etc la información puede ser compartida entre varios sistemas sean externos o internos, si la compartición es externa hay que analizar la comunicación y también las seguridades.

Salida

Perspectivas del producto.

3.3: Definición de requerimientos funcionales

Entrada

Descripción del dominio de la aplicación

Identificación del marco de Problema

Razonamiento

Aquí el experto comienza a realizar la especificación de los requerimientos utilizando la información elicitada. La definición de los requerimientos funcionales se encuentra definido por procesos genéricos como son los ABC (Altas, Bajas y Cambios) generadas para cada entidad perteneciente a un dominio de la aplicación. Sin embargo existen otros requerimientos específicos del dominio en los que través de un análisis se los puede identificar y especificar con un grado de detalle mayor.

Salida

Descripción del requerimiento funcional

3.3.1.1 Determinar tipo de Requerimiento

Entrada

Documentación elicitada

Descripción del Marco de Problema

Razonamiento

El experto analiza también otras funcionalidades en base a las entradas proporcionadas para el requerimiento. Este análisis se basa en identificar el

tipo de requerimiento que puede ser de consulta, Mapeo, Reglas de comportamiento, operaciones sobre dominios, y correspondencia sobre dominios. Una vez Identificado el tipo de requerimiento, el experto puede recomendar sobre especificaciones técnicas e identificar el Marco de Problema en el que se encuentra el requerimiento.

3.4 Definición de requerimientos no funcionales

Entradas

Documentos elicitados

Entrevistas

Razonamiento

El experto con la documentación elicitada y las entrevistas realizadas a los clientes, empieza a definir los requerimientos no funcionales, dichos requerimientos describen características del sistema como: disponibilidad, seguridad, fiabilidad, concurrencia.

Cada uno de estas características son identificadas de acuerdo a ambientes en el que el sistema se vaya a desarrollar, por ejemplo, evalúa la seguridad de acuerdo a la sensibilidad de los sistemas, si son muy sensibles recomienda hacer auditorias y logs de encriptación. Para la disponibilidad analiza el tipo en el la aplicación se mantiene disponible. Para la concurrencia analiza por ejemplo usuarios conectados simultáneamente, la capacidad de almacenamiento, tiempo de respuesta, etc.

Salida

Descripción de Requerimientos no funcionales.

Estos conocimientos Estratégicos han sido validados por el experto mediante entrevistas realizadas, en las que ha determinado que las tareas descritas en este documento se encuentran acorde a las tareas que el experto realiza.

5.2.2.3 COMPROBACIÓN DE LOS CONOCIMIENTOS ESTRATÉGICOS

El modelo funcional ha sido validado por el experto, para ello se han realizado retroalimentaciones de las entrevistas en las cuales el experto ha evaluado la representación de conocimientos para el desempeño de las tareas, La comprobación de esta representación ha sido satisfactoria.

5.2.3 CONOCIMIENTOS TÁCTICOS

Los conocimientos tácticos determinan como el sistema puede utilizar los hechos conocidos y las hipótesis tanto en situaciones deterministas como en situaciones de incertidumbre.

El análisis de estos conocimientos producen detalladamente cada paso de razonamiento que el sistema debe hacer.

Para documentar estos conocimientos se ha utilizado para este trabajo las seudoreglas, que de acuerdo a la propuesta Gómez [Gómez, A. y otros 1997] tienen la forma descrita en la tabla 5.4.

Tabla 5. 4 Diseño de seudoreglas

Estado de la regla	Texto de la regla
Palabras del experto	
Formulación externa de la regla	
Nombre de la regla	

A partir de los conocimientos obtenidos en la Adquisición de Conocimientos y de la descripción detallada de la descomposición funcional descrita en la figura 5.2 se describen las seudoreglas.

Abreviaturas utilizadas para la reglas

Dominio-Información

Dominio-Control

Dominio- Workspiece

Req-no-func-Herramientas.

Req-no-func-Lenguaje-y-BDD
 Req-no-func-Lenguaje-y-BDD-Internet
 Req-no-func-Software-Libre-Herramientas-Internet
 Req-no-func-Lenguaje-Internet_Dispositivos –Móviles
 Req-no-func-Software-propietario-Computadores-Escritorio
 Req-no-func-Multiple-Lenguajes.
 Req-no-func-Software-propietario-BDD-Internet
 Req-no-func-Software-propietario-Computadores-Escritorio
 Req-no-func-Multiple-Lenguajes.
 Req-no-func-Conexión-Externa_Servicio_Web
 Req-no-func-Conexión-Externa_Envio-Tramas
 Req-no-func-Conexión-Externa_Consultas
 Req-no-func-Sistemas-Externos -COM+
 Req-no-func-Seguridad
 Req-no-func-Seguridad-Media
 Req-no-func-Seguridad-baja
 Req-no-func-Disponibilidad-Capacidad

A partir de la tabla 5.5 hasta la tabla 5.36 se describen las siguientes actividades:

Análisis del Dominio en el que se define el marco de problema.

Delimitación de restricciones en el que cuenta las políticas de la empresa y las restricciones de desarrollo.

Definición de requerimientos Funcionales, los parámetros que describen al requerimiento funcional son ingresados por el usuario, pero el sistema experto puede realizar un análisis del dominio de acuerdo al tipo de requerimiento registrado.

Definición de requerimientos no funcionales considera características del sistema como disponibilidad, seguridad, fiabilidad.

Conexión con sistemas externos, en los que analiza la información de envío y recepción entre los sistemas.

Tabla 5. 5 Análisis del Dominio: Marco de problema de Información

Estado de la regla	Texto de la regla
Palabras del experto	<p>El análisis de un dominio se lo realiza en base a lo que los usuarios inicialmente lo expresan verbalmente y con la documentación existente.</p> <p>El análisis empieza con decidir con que marco de problema nos encontramos, siendo complejo si se profundiza sobre dominios, de todas maneras podemos decidir sobre uno de los cinco tipos de marcos de problemas elementales, como por ejemplo:</p> <p>Si a partir de la documentación elicitada se observa que los usuarios realizan consultas, o pedidos de información sin realizar cualquier otro cambio con la información de salida entonces podemos deducir que estamos con un marco de problema de Información</p>
Formulación externa de la regla	Si el requerimiento necesita consultas de información entonces Marco de problema de Información
Nombre de la regla	Dominio-Información

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 6 Análisis del Dominio: Marco de problema de control

Estado de la regla	Texto de la regla
Palabras del experto	Si a partir de la documentación elicitada se observa que para cumplir el requerimiento utiliza dispositivos externos entonces podemos deducir que estamos frente a un marco de problema de control.

Formulación externa de la regla	Si el requerimiento necesita dispositivos externos entonces Marco de problema de control
Nombre de la regla	Dominio-Control

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 7 Análisis del Dominio: Marco de problema de Workspiece

Estado de la regla	Texto de la regla
Palabras del experto	Si a partir de la documentación elicitada se observa que el requerimiento tiene la funcionalidad de crear documentación a partir de información de entrada entonces estamos frente a un marco de problema de Workspiece
Formulación externa de la regla	Si el requerimiento tiene la funcionalidad de crear documentación a partir de información de entrada entonces Marco de problema de Workspiece
Nombre de la regla	Dominio-Workspiece

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 8 Análisis del Dominio: Marco de problema de conexión

Estado de la regla	Texto de la regla
Palabras del experto	Si a partir de la documentación elicitada se observa que el requerimiento tiene relación con varios dominios entonces nos encontramos con un marco de problema de conexión
Formulación externa de la regla	Si el requerimiento tienen relación con varios dominios entonces marco de

	problema de conexión
Nombre de la regla	Dominio-Conexión

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 9 Delimitaciones y restricciones: Herramientas de desarrollo

Estado de la regla	Texto de la regla
Palabras del experto	Las restricciones pueden ser de desarrollo, siendo los lenguajes de programación herramientas que la empresa puede limitar para su desarrollo estas pueden ser a través de Software libre o propietario
Formulación externa de la regla	Si la política es software libre y el alcance del sistema es solo dentro de la empresa entonces java con herramientas de escritorio
Nombre de la regla	Req-no-func-Herramientas

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 10 Delimitaciones y restricciones: Herramientas de desarrollo

Estado de la regla	Texto de la regla
Palabras del experto	Las empresas pueden optar por software libre en donde el manejo del sistema tiene un alcance interno a través de computadores de escritorio, y en la que por lo general toda aplicación hace uso de base de datos.
Formulación externa de la regla	Si la política es software libre y los usuarios utilizan desde computadores de escritorio y bases de datos entonces java con herramientas de escritorio y base de datos MYSQL

Nombre de la regla	Req-no-func-Lenguaje-y-BDD
--------------------	----------------------------

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 11 Delimitaciones y restricciones: Herramientas de desarrollo

Estado de la regla	Texto de la regla
Palabras del experto	Las empresas pueden optar por software libre en donde el manejo del sistema tiene un alcance externo a través de internet con computadores de escritorio, y en la que por lo general toda aplicación hace uso de base de datos
Formulación externa de la regla	Si la política es software libre y los usuarios utilizan desde Internet con computadores de escritorio y bases de datos entonces PHP con herramientas WEB y base de datos MYSQL
Nombre de la regla	Req-no-func-Lenguaje-y-BDD-Internet

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 12 Delimitaciones y restricciones: Herramientas de desarrollo

Estado de la regla	Texto de la regla
Palabras del experto	Las empresas pueden optar por software libre en donde el manejo del sistema tienen un alcance externo a través de internet con dispositivos móviles, y manejo de bases de datos.
Formulación externa de la regla	Si la política es software libre y los usuarios utilizan desde Internet con dispositivos móviles y bases de datos entonces java con herramientas WEB y base de datos MYSQL
Nombre de la regla	Req-no-func-Lenguaje-

	Internet_Dispositivos -Móviles
--	--------------------------------

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 13 Delimitaciones y restricciones: Herramientas de desarrollo

Estado de la regla	Texto de la regla
Palabras del experto	Dependiendo de las aplicaciones las empresas pueden optar por herramientas de desarrollo para software libre en las que no incluyan bases de datos entonces las mayores alternativas son java y PHP.
Formulación externa de la regla	Si la política es software libre y los usuarios utilizan computadoras de escritorio entonces java con herramientas de escritorio
Nombre de la regla	Req-no-func-Lenguaje-Herramientas

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 14 Delimitaciones y restricciones: Herramientas de desarrollo

Estado de la regla	Texto de la regla
Palabras del experto	Dependiendo de las aplicaciones las empresas pueden optar por herramientas de desarrollo para software libre en las que no incluyan bases de datos entonces las mayores alternativas son java y PHP.
Formulación externa de la regla	Si la política es software libre y los usuarios utilizan herramientas para Internet entonces PHP con herramientas Web.
Nombre de la regla	Req-no-func-Software-Libre-Herramientas-Internet

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 15 Delimitaciones y restricciones: Herramientas de desarrollo

Estado de la regla	Texto de la regla
Palabras del experto	Si las empresas optan por software propietario, de igual forma se dan los casos que puedan ser aplicaciones en donde el alcance es solo dentro de la empresa utilizando bases de datos, entonces las alternativas de herramientas de desarrollo son lenguajes propietarios que han surgido con mayor demanda en el mercado,
Formulación externa de la regla	Si la política es software propietario y los usuarios utilizan computadores de escritorio entonces C# con herramientas de escritorio y MSQl.
Nombre de la regla	Req-no-func-Software-propietario-Computadores-Escritorio

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 16 Delimitaciones y restricciones: Herramientas de desarrollo

Estado de la regla	Texto de la regla
Palabras del experto	Si las empresas optan por software propietario, el alcance puede ser a través del internet utilizando bases de datos, entonces las alternativas de herramientas de desarrollo son lenguajes propietarios que han surgido con mayor demanda en el mercado,
Formulación externa de la regla	Si la política es software propietario y los usuarios utilizan Internet entonces C# con herramientas de Web y MSQl.

Nombre de la regla	Req-no-func-Software-propietario-BDD-Internet
--------------------	---

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 17 Delimitaciones y restricciones: Herramientas de desarrollo

Estado de la regla	Texto de la regla
Palabras del experto	Las aplicaciones pueden usar herramientas en las que no incluyan bases de datos entonces se opta por lenguajes con mayor demanda.
Formulación externa de la regla	Si la política es software propietario y los usuarios utilizan computador de escritorio entonces C# con herramientas de escritorio.
Nombre de la regla	Req-no-func-Software-propietario-Computadores-Escritorio

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 18 Delimitaciones y restricciones: Herramientas de desarrollo

Estado de la regla	Texto de la regla
Palabras del experto	Las aplicaciones pueden usar herramientas en las que no incluyan bases de datos entonces se opta por lenguajes con mayor demanda y que se utilice herramientas para internet.
Formulación externa de la regla	Si la política es software propietario y los usuarios utilizan internet entonces C# con herramientas Web.
Nombre de la regla	Req-no-func-Software-propietario-Computadores-Escritorio-Internet

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 19 Delimitaciones y restricciones: Herramientas de desarrollo

Estado de la regla	Texto de la regla
Palabras del experto	Las aplicaciones pueden combinar entre software libre y software propietario, en ese caso se utiliza archivos de globalización que puedan interrelacionarse entre las herramientas de desarrollo
Formulación externa de la regla	Si utiliza múltiples lenguajes entonces archivos de globalización.
Nombre de la regla	Req-no-func-Multiple-Lenguajes

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 20 Conexión con sistemas externos: Servicios Web

Estado de la regla	Texto de la regla
Palabras del experto	Las aplicaciones pueden residir en varios lugares y la distribución o compartición de archivos pueden ser por varios métodos, por ejemplo si la transmisión de información es por servicios Web es necesario determinar parámetros del que los sistemas clientes piden, para lo que es necesario crear un servicio Web utilizando un archivo WDSL, este archivo describe la forma de comunicación de archivos, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios que se necesite.
Formulación externa de la regla	Si la gestión de ingresos hacia sistemas externos es servicios Web entonces enviar requisitos como dirección del servicio Web, archivo WDSL.

Nombre de la regla	Req-no-func-Conexión-Externa_Servicio_Web
--------------------	---

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 21 Conexión con sistemas externos: Tramas

Estado de la regla	Texto de la regla
Palabras del experto	La información recibida y enviada puede ser por tramas que es la que se transmite por la red. Las tramas son documentos ya estipulados que tienen una cabecera y un detalle, a través de la cabecera se sabe cuál es el detalle, se crea un método que interprete la trama y los requisitos de entrada salida para crear el formato de la trama de éxito y de error. Cuando la información es enviada por tramas se puede entonces decidir que para la recepción de archivos se va a consumir un archivo plano en un servicio FTP con las especificación de formato que se especifica, entonces lo que se necesita para consumir el archivo plano es una dirección FTP, la dirección y password para acceder al servidor FTP y la hora en que el archivo está listo para consumir.
Formulación externa de la regla	Si la gestión de ingresos hacia sistemas externos son por tramas. Entonces Archivo plano en un servidor FTP con las especificaciones de formato que se indiquen. Requisitos: - Dirección del FTP, Usuario y pwd del servidor FTP, formato del archivo, archivo WDSL.
Nombre de la regla	Req-no-func-Conexión-Externa_Envio-Tramas

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 22 Conexión con sistemas externos: Bases de Datos

Estado de la regla	Texto de la regla
Palabras del experto	<p>Cuando la comunicación se realiza a través de Bases de Datos, tenemos que poner una estructura idéntica de nuestra estructura de la base de datos donde vamos a extraer.</p> <p>Los requisitos que se necesitan para tener la estructura de la base de datos son: el usuario, clave del servidor y el puerto de comunicación entonces la estructura de la base debe ser la misma del sistema con la del servidor.</p>
Formulación externa de la regla	<p>Si la gestión de ingresos hacia sistemas externos es por bases de datos clonando estructuras entonces Archivo de la estructura de nuestra base de datos, - Proporcionar Usuario y clave del servidor, Proporcionar puerto de comunicación,</p>
Nombre de la regla	Req-no-func-Conexión-Externa_BDD

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 23 Conexión con sistemas externos: Bases de Datos

Estado de la regla	Texto de la regla
Palabras del experto	<p>Cuando la comunicación se realiza a través de Bases y la extracción se la realiza mediante consultas que se guardarán en tablas o vistas de nuestra base de datos, entonces crearemos tablas temporales o vistas que reciban la información extraída de los queries que se ejecuten en la base cliente</p>

Formulación externa de la regla	Si la gestión de ingresos hacia sistemas externos es por bases de datos mediante consultas que se guardaran en tablas o vistas entonces Estructuras de consulta a ejecutar, usuarios y password de la Base de datos.
Nombre de la regla	Req-no-func-Conexión-Externa_Consultas

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 24 Conexión con sistemas externos: COM+

Estado de la regla	Texto de la regla
Palabras del experto	La comunicación de procesos también son realizados por componentes que permiten la comunicación dinámica para muchos lenguajes de programación. Si requiere que la implementación de objetos sean independientes al lenguaje entonces utilizaríamos COM+
Formulación externa de la regla	Si la gestión de ingresos hacia sistemas externos COM+ entonces Proporcionar COM+ cliente al servidor, parámetros de entrada salida.
Nombre de la regla	Req-no-func-Sistemas-Externos -COM+

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 25 Definición requerimientos no Funcionales: Seguridad

Estado de la regla	Texto de la regla
Palabras del experto	La sensibilidad de los sistemas depende del tipo de Sistema que se vaya a desarrollar. Por ejemplo en sistemas bancarios existe una alta sensibilidad y

	<p>es necesario optar por una serie de políticas que permitan tener una seguridad sobre los datos.</p> <p>Existen aplicaciones en las que la sensibilidad es menor entonces las políticas tendrán un nivel menor de implementación, esto no quiere decir que el sistema no es seguro, sino que las políticas serán otras</p>
Formulación externa de la regla	Si sensibilidad alta(4) entonces manejar con autenticación de usuarios, roles y perfiles de expiración de claves y reglas mínimas de clave, métodos para el control de la auditoría y encriptaciones del canal de envío y bases de datos
Nombre de la regla	Req-no-func-Seguridad

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 26 Definición requerimientos no Funcionales: Seguridad

Estado de la regla	Texto de la regla
Palabras del experto	Si las aplicaciones tienen una sensibilidad menor entonces las políticas tendrán un nivel menor de implementación, esto no quiere decir que el sistema no es seguro, sino que las políticas serán otras
Formulación externa de la regla	Si sensibilidad media (2) entonces maneja autenticación de usuarios sin autorías ni encriptación en la comunicación.
Nombre de la regla	Req-no-func-Seguridad-Media

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 27 Definición requerimientos no Funcionales: Seguridad

Estado de la regla	Texto de la regla
Palabras del experto	Si las aplicaciones tienen una sensibilidad media entonces las políticas tendrán un nivel medio de implementación, esto no quiere decir que el sistema no es seguro, sino que las políticas serán otras
Formulación externa de la regla	Si sensibilidad media (3) entonces manejar autenticación usuarios con roles y perfiles expiración de claves y reglas mínimas de clave, métodos para el control de la auditoría y la comunicación se hará encriptado el canal de envío.
Nombre de la regla	Req-no-func-Seguridad-Media

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 28 Definición requerimientos no Funcionales: Seguridad

Estado de la regla	Texto de la regla
Palabras del experto	Si las aplicaciones tienen una sensibilidad baja entonces las políticas tendrán un nivel bajo de implementación, esto no quiere decir que el sistema no es seguro, sino que las políticas serán otras
Formulación externa de la regla	Si sensibilidad baja (1) entonces se realizará sin autenticación de usuarios ni auditorías de acceso y manipulación de datos.
Nombre de la regla	Req-no-func-Seguridad-baja

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 29 Definición requerimientos no Funcionales: Disponibilidad

Estado de la regla	Texto de la regla
Palabras del experto	Los sistemas tienen tiempo en los que generalmente pasan operando, esto determina la disponibilidad de los mismos, si los sistemas operan más de 8 horas regularmente se necesita tener consideraciones sobre los servidores para que no sufran caídas que afecten a la disponibilidad.
Formulación externa de la regla	Si La operación del sistema es de 8 horas a más entonces Se necesitan habilitar servidores replica para evitar que las caídas del sistema afecten la disponibilidad.
Nombre de la regla	Req-no-func-Disponibilidad-Alta

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 30 Definición requerimientos no Funcionales: Disponibilidad

Estado de la regla	Texto de la regla
Palabras del experto	Los sistemas tienen tiempo en los que generalmente pasan operando, esto determina la disponibilidad de los mismos, si los sistemas operan menos de 8 se realizan consideraciones sobre los servidores sacar restauraciones y respaldos por contingencia
Formulación externa de la regla	Si La operación del sistema es menor a 8 horas entonces necesitamos un servidor robusto para el alojamiento de la aplicación y programar puntos de restauración y backups.
Nombre de la regla	Req-no-func-Disponibilidad-Media

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 31 Definición requerimientos no Funcionales: Capacidad

Estado de la regla	Texto de la regla
Palabras del experto	La capacidad de los servidores dependen de la información almacenada siendo estos medidos por unidades de almacenamiento en GB o TB entonces se puede definir la capacidad del servidor con una proyección determinada esto depende además de la cantidad de usuarios que acceden simultáneamente
Formulación externa de la regla	Si capacidad 1-5 GB entonces Su software necesita un servidor moderadamente robusto con 100 GB de storage y un procesador, sin la necesidad de virtualización y sistemas de respaldos.
Nombre de la regla	Req-no-func- -Capacidad

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 32 Definición requerimientos no Funcionales: Capacidad

Estado de la regla	Texto de la regla
Palabras del experto	La capacidad de los servidores dependen de la información almacenada siendo estos medidos por unidades de almacenamiento en GB o TB entonces se puede definir la capacidad del servidor con una proyección determinada esto depende además de la cantidad de usuarios que acceden simultáneamente
Formulación externa de la regla	Si capacidad 5-10 GB entonces Su software necesita un servidor robusto

	con 500 GB de storage y dos procesadores, sin la necesidad de virtualización y sistemas de respaldos.
Nombre de la regla	Req-no-func-Disponibilidad-Capacidad

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 33 Definición requerimientos no Funcionales: Capacidad

Estado de la regla	Texto de la regla
Palabras del experto	La capacidad de los servidores depende de la información almacenada, siendo estos medidos por unidades de almacenamiento en GB o TB entonces se puede definir la capacidad del servidor con una proyección determinada esto depende además de la cantidad de usuarios que acceden simultáneamente
Formulación externa de la regla	Si capacidad 10 a más GB entonces necesita un servidor robusto con 2 TB de storage y cuatro procesadores, con virtualización y sistema de respaldos.
Nombre de la regla	Req-no-func-Disponibilidad-Capacidad Alta

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 34 Definición requerimientos Funcionales: Requerimiento de consultas

Estado de la regla	Texto de la regla
Palabras del experto	Los requerimientos funcionales describen las necesidades de los usuarios, trasladar estas necesidades a especificaciones técnicas es lo que se necesita para 138 pasar a una etapa de desarrollo. En este sentido las recomendaciones

	que se pueden dar sobre un requerimiento funcional, es de acuerdo al análisis de marcos de problemas, siendo por ejemplo si los requerimientos son de consulta entonces podemos definir formatos de salidas, y recomendaciones de bases de datos.
Formulación externa de la regla	Si requerimiento de consulta entonces reportes donde se muestre detallada la información de consulta y en varios formatos de salida ejemplo PDF, DOC, XLS. 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 las búsquedas tengan una respuesta óptima. Colocar siempre una clave principal dentro de cada tabla creada en la base de datos para mejorar el performance y la utilidad de las consultas.
Nombre de la regla	Req--func-Consultas

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 35 Definición requerimientos Funcionales: Requerimiento de Reglas de comportamiento

Estado de la regla	Texto de la regla
Palabras del experto	Si para cumplir los requerimientos necesitan controlar dispositivos externos, ejemplo: pistola de código de barras, micro controladores, plc, otros. Entonces se recomendaría acciones que permitan controlar estos dispositivos por software como manejo de driver, librerías y

	manejo de comunicación
Formulación externa de la regla	Si requerimiento necesita controlar dispositivos entonces Utilizar los drivers de instalación que viene por defecto dentro del equipo que vamos a controlar dentro de nuestro software. Procurar que la información que ingresa desde el dispositivo externo llegue en archivos planos para que su manipulación sea transparente hacia nuestro software. Contar con documentación sobre la maquinaria que se va a controlar para conocer su interfaz de comunicación y el manejo de sus funcionalidades.
Nombre de la regla	Req--func-Regla-de-comportamientos

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 5. 36 Definición requerimientos Funcionales: Operaciones sobre dominios creados

Estado de la regla	Texto de la regla
Palabras del experto	Si para cumplir los requerimientos tendrá la funcionalidad de crear documentación a partir de la información de entrada entonces se recomienda acciones sobre manejo de artefactos como documentos e imágenes.
Formulación externa de la regla	Si requerimiento necesita crear documentación entonces Utilizar librerías nativas del lenguaje de programación manejo de imágenes y de texto. Para el almacenamiento de imágenes se recomienda colocarlas en una carpeta dentro del servidor de la aplicación 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 negocio para que esto no afecte el performance de la aplicación. Que los documentos generados puedan ser impresos en varios formatos para facilidad del cliente final
Nombre de la regla	Req--func-Operaciones-Sobre-dominios-creados

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

5.2.3.1 COMPROBACIÓN DE LOS CONOCIMIENTOS TÁCTICOS

Se ha comprobado los conocimientos tácticos mediante la coincidencia del accionar del experto con las representaciones de los conocimientos tácticos, para ello el experto ha validado, completado y revisado todas las seudoreglas mismas que fueron probados en los casos que se presentan en el capítulo 8.

5.2.4 CONOCIMIENTOS FÁCTICOS

En esta etapa se determina la información que el S.E. debe conocer, se organiza la información recopilada de cada uno de los atributos que intervienen en el dominio de la aplicación.

Estos atributos son descritos en las tablas 5.37.

Tabla 5. 37 Descripción de Atributos

NOMBRE	ENTRADA	SALIDA	CALCULADO	TIPO DE VALOR	POSIBLE VALOR	DESCRIPCIÓN
Id del proyecto		✓		Numérico	Ref. única del documento	Identifica el Documento generado por un usuario
Nombre	✓			Texto	Texto libre	Describe el nombre del proyecto

Área			✓	Texto	Texto libre	Identifica el área o sub área del proyecto
Objetivos	✓			Texto	Texto libre	Texto que describe los objetivos del proyecto
Descripción	✓			Texto	Texto libre	Descripciones rápidas de lo que se espera del proyecto
Descripción del Marco de Problema		✓		Texto	Texto que identifica al tipo de marco de problema	Texto que describe a un tipo de Marco de problema elemental
Lenguaje	✓			Texto	Si No	Selecciona el tipo de lenguaje que debe desarrollarse la aplicación
Bases de datos	✓			Texto	Si No	Identifica las base de datos para el desarrollo de la aplicación
Alcance de acceso hacia el sistema	✓			Texto	Dentro de la empresa Desde Internet con computador de escritorio Desde Internet utilizando dispositivos	Determina la utilización del sistema interna o externamente
Tipos de Sistemas	✓			Texto	Cliente-Servidor Distribuido	Identifica el tipo de sistema en el que el sistema va a operar
Tipo de Información	✓			Texto	Servicios Web Archivos planos Tramas	Determina la comunicación entre sistemas para el envío y recepción de

					Clonando estructuras de BDD Consultas	archivos.
Disponibilidad		✓		Numérico	1-8 horas 8-más horas	Determina el tiempo en el que el sistema estará en funcionamiento
Seguridad		✓		Numérico	[1-4]	Determina el nivel de sensibilidad de los sistemas
Concurrente	✓			Numérico	1-20 20-50 50-más	Determina cuantos usuarios estarán conectados simultáneamente al sistema
Capacidad	✓			Numérico	1-5 GB 5-10 GB 10-más GB	Define la capacidad de almacenamiento de los servidores en un tiempo estimado
Tipo de Requerimiento		✓			Consulta Reglas de comportamiento Mapeo Operación sobre dominios creados Correspondencia entre dominios	Determina el tipo de requerimiento de acuerdo a las entradas proporcionadas por el usuario
Recomendaciones para el requerimiento		✓			Texto descriptivo	Texto que describe las inferencias realizadas al requerimiento funcional

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

La Tabla 5.38 describe el proceso de educación y conceptualización de requerimientos que realizará el Sistema a través de marcos de problema. Formulándose preguntas específicas sobre subproblemas identificados para los usuarios, que permita en la etapa de formalización inferir los marcos de problemas conocidos para luego poderlos tipificarlos en requerimientos funcionales

Tabla 5. 38 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 o Inferido	Detalle de Especificación del Requerimiento
ER 1	¿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.
ER 2	¿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.
ER 3	¿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.
ER 4	¿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.

ER 5	¿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
------	---	---	-------	------------------	---	---

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Para el caso de los requerimientos no funcionales este se basa en una serie de preguntas puntuales sobre restricciones de diseño del Sistema, la misma que se detalla en la tabla 5.39

Tabla 5. 39 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"> • Software Libre • Ambos
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"> • Si • No
ER9	Alcance de acceso de su sistema	Permite determinar el acceso de comunicación del sistema.	<ul style="list-style-type: none"> • Utiliza solo dentro de la empresa • Utiliza desde Internet con computador de escritorio • Utiliza desde Internet utilizando dispositivos como tablets, celulares etc.
ER10	El Sistema necesita tener bases de datos	Permite que el sistema almacene la información permanentemente	<ul style="list-style-type: none"> • Si • No
ER11	El sistema necesita compartir información con otros sistemas	Permite determinar si existe interconexión con sistemas externos	<ul style="list-style-type: none"> • Si • No
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"> • Servicios Web • Archivos planos • Tramas • Clonando estructuras de Base De Datos • Consultas
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"> • 1-8 horas • 8-más horas

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"> • 1 • 2 • 3 • 4
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"> • 1-5 GB • 5-10 GB • 10-más GB

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

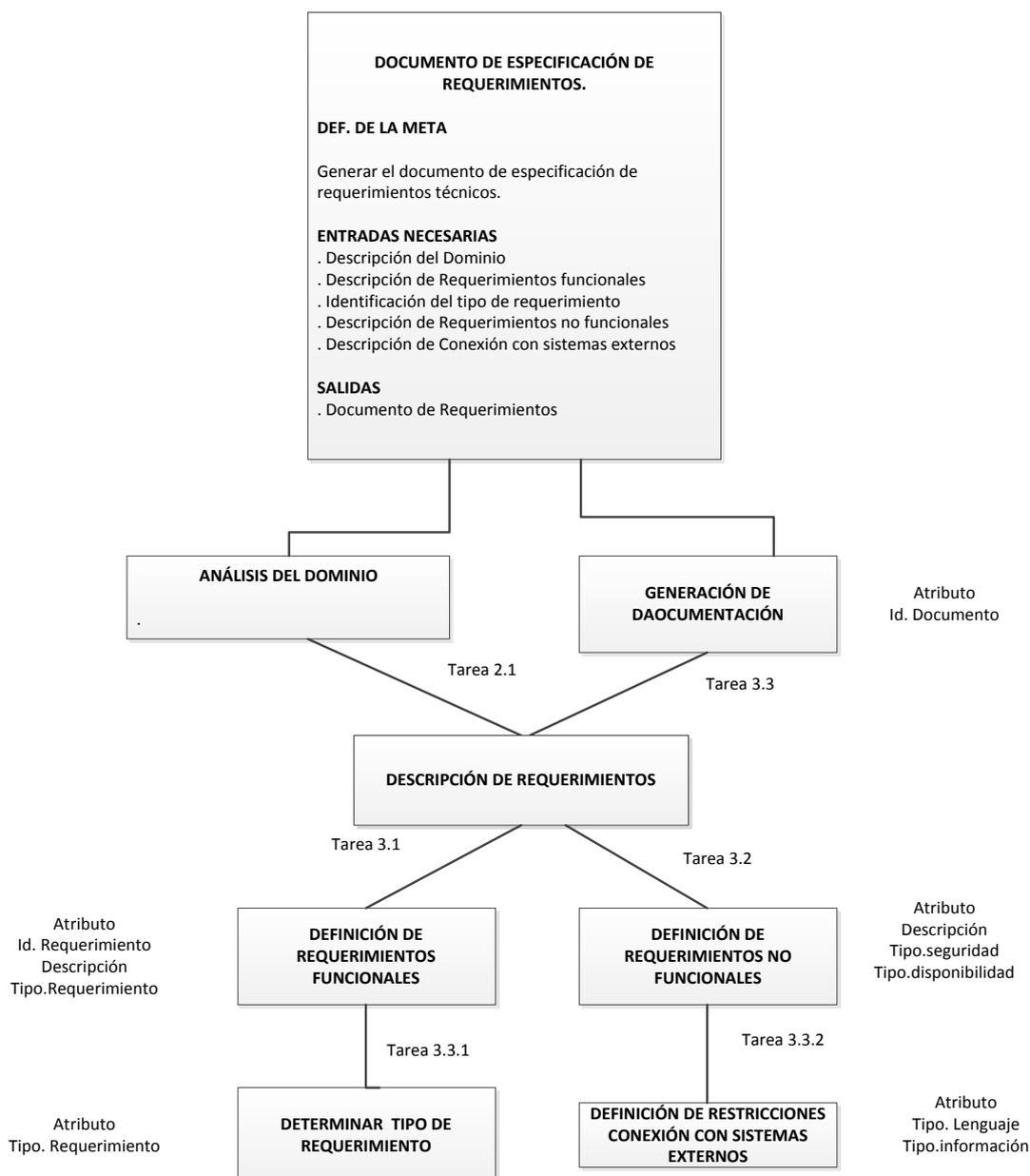
5.2.4.1 COMPROBACIÓN DE LOS CONOCIMIENTOS FÁCTICOS

Para la comprobación de los conocimientos fácticos se han realizado revisiones exhaustivas de la definición de los atributos en forma conjunta con el experto. Lo que permite completar concepto-atributos-valores y revisar las seudoreglas definidas anteriormente, considerando esta comprobación satisfactoria.

5.2.5 MODELO DINÁMICO O DE PROCESOS

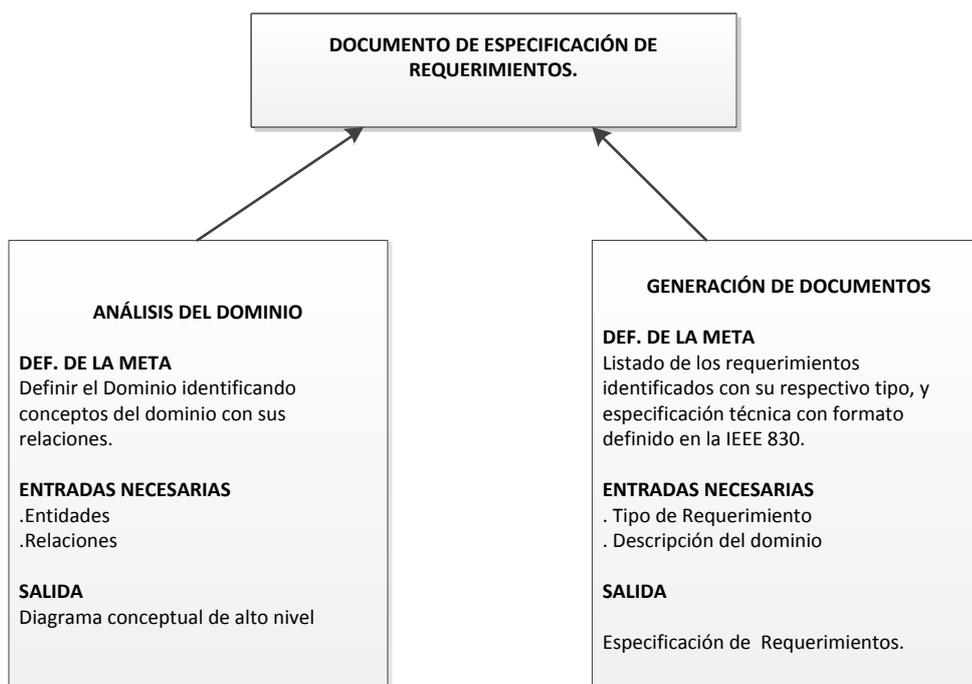
Luego de haber analizado los conocimientos estratégicos, fácticos y tácticos se realiza la síntesis. La síntesis consiste en representar los conocimientos en modelos, para este trabajo se ha tomado el modelo dinámico o de procesos representado en las figuras 5.3, 5.4 y 5.5

Para elaborar el modelo dinámico nos hemos valido del estudio de las tareas obtenidas en la etapa de análisis, en el apartado 5.2, se adquirió los conocimientos estratégicos detallando a la tarea con el razonamiento del experto, este análisis permite representar el modelo dinámico en el que se describe jerárquicamente a las tareas que realiza el experto incluyendo las metas y atributos utilizados.



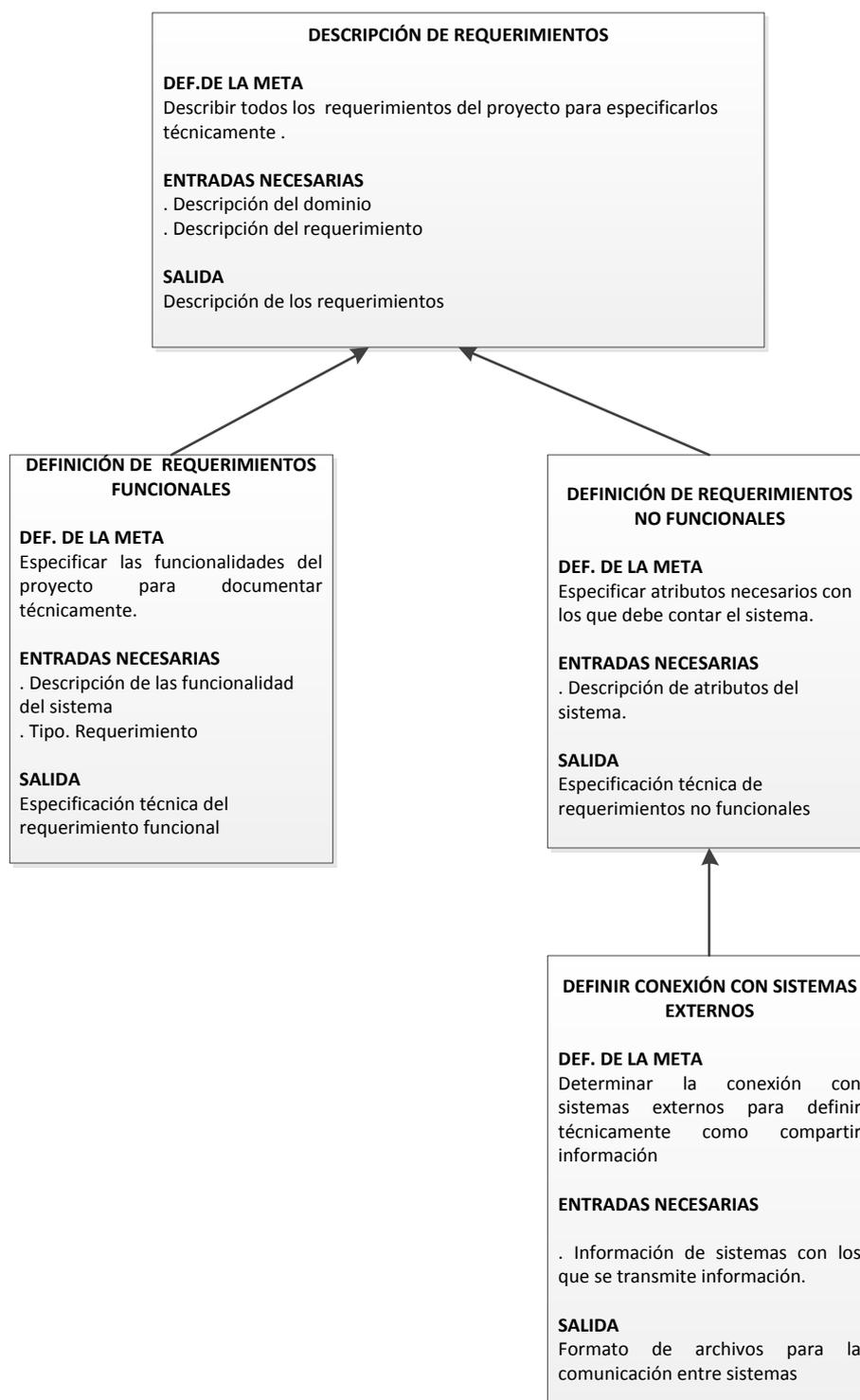
Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 5. 3. Diagrama jerárquico de tareas



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 5. 4. Jerarquía de tareas del documento de requerimientos



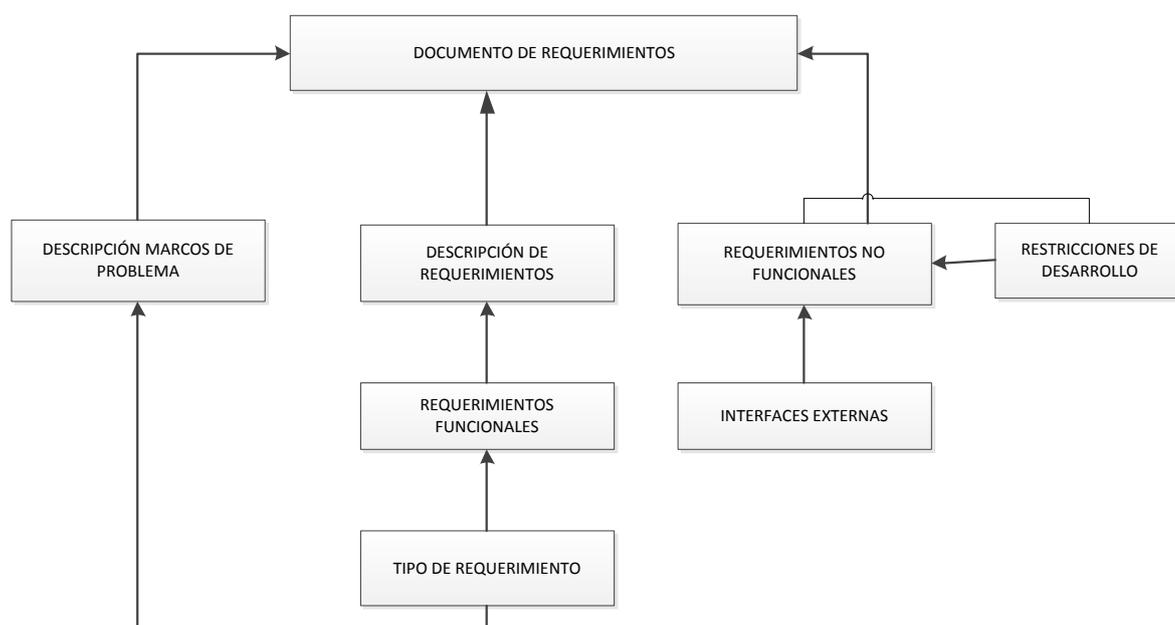
Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 5. 5. Jerarquía de tareas de Descripción de requerimientos

5.2.6 MODELO ESTÁTICO

Para representar el modelo estático se han tomado los conocimientos estratégicos en los que se ha documentado: Glosario de términos (tabla 5.1), Diccionario de Conceptos (tabla 5.2), Tabla concepto- atributo-valor (tabla 5.3) y Modelo relacional de Conceptos(figura 5.1).

La figura 5.6 describe las relaciones y los conceptos en forma esquemática.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 5. 6. Mapa Conceptual

5.3 COMPROBACIÓN DE LA CONCEPTUALIZACIÓN

Parte de la comprobación de la conceptualización es la comprobación de los distintos modelos construidos en ésta fase, ver secciones 5.2.2.3, 5.2.3.1, 5.2.4.1. Para verificar la consistencia entre los modelos se puso énfasis en la comprobación del mapa de conocimientos. La comprobación se realizó de la siguiente manera:

- Se verificó que todos los atributos sean ingresados por el usuario.
- Se verificó que los atributos inferidos no fueran subjetivos y que estuvieran contenidos en una regla.

- Se verifico que los valores de los atributos no tengan valores por omisión.
- Se comprobó que los valores de los atributos sean usados, los que no fueron usados en ésta versión del prototipo se los dejó como parte del modelo conceptual a considerar en próximas versiones del prototipo.
- Se completó y verificó que todos los atributos se encuentren en la tabla de concepto-atributo-valor.

Para validación del modelo conceptual se usaron los casos de prueba detallados en el capítulo 8 cuyos resultados fueron satisfactorios.

CAPÍTULO VI

FORMALIZACIÓN

6.1 INTRODUCCIÓN

La etapa de formalización tiene como objetivo expresar los conocimientos en estructuras que son entendidas por el computador, consiste en otras palabras, en representaciones simbólicas del conocimiento.

6.2 Selección de las Técnica de Formalización

Para la selección de la técnica, es necesario tomar la que mejor represente al conocimiento del dominio, y a su vez que sea flexible para la adquisición y representación del mundo exterior. Las técnicas utilizadas para el presente trabajo descritas por Gómez et al. (1997) se cita a continuación.

6.2.1 SISTEMAS DE PRODUCCIÓN. Basadas en seudoreglas que representan un conjunto de reglas con estructuras *Si...Entonces*, la parte del **Si** constituye los antecedentes que son acciones que se deben verificar y la parte de **Entonces** es lo consecuente, que representan las acciones a tomar si los antecedentes son ciertos, es decir una clásica regla IF-THEN.

Los Sistemas de producción cuenta con una base de conocimientos (BC) expresados en hechos y datos de cualquier información relevante para el problema a resolver. La base de conocimiento para el sistema de requerimientos son los datos elicitados por los usuarios. Los Sistemas de producción cuentan además de una estrategia de control que especifique el orden en el que las reglas se comparan con la base de conocimiento y la forma de resolver los conflictos que surjan cuando varias reglas puedan ser aplicadas a la vez Peña (2006).

6.2.2. REPRESENTACIÓN DE LOS CONOCIMIENTOS EN MARCOS.

Basados en la tabla Concepto-Atributo-Valor, se formaliza a través de marcos. Cada concepto representa un marco clase, las ranuras para cada marco clase son los atributos y los valores representan valores permitidos

que puede tomar cada atributo. La tabla Concepto-atributo-valor se ha formalizados como lo describe la tabla 6.1.

Tabla 6. 1. Tabla-Concepto-Valor

Tabla Concepto- Atributo-Valor	Marcos
Concepto	Marcos Clase (M.C)
Atributos	Ranuras del Marco clase
Valores	Valores Permitidos

A partir de la tabla 6.2 hasta la tabla 6.9 se describe la representación del conocimiento en Marcos Clase para el Sistema Experto

Tabla 6. 2. Marco Clase Proyecto

MC Proyecto Ranura	Tipo de ranura	Max/Min	Multiv	Propiedad General	Valor permitido	Valor /omisión	Si necesito	Si modifico	Si Borro
Nombre	Conjunto de caracteres	1/1	No	_____	String	-	-	-	-
Área	Conjunto de caracteres	1/1	No	_____	String	-	-	-	-
Nombre del Sistema	Númérico Conjunto de caracteres	1/1	Si	_____	Númérico	-	-	-	-
Descripción	Conjunto de caracteres	1/1	No	_____	String	-	-	-	-

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 6. 3 Marco Clase Proyecto

MC Marco de problema Ranura	Tipo de ranura	Max/Min	Multiv	Propiedad General	Valor permitido	Valor omisión	Si necesito	Si modifico	Si Borro
Descripción	Conjunto de caracteres	1/1	No	_____	String	-	-	-	-
Tipo	Conjunto de caracteres	1/5	Si	_____	String	-	-	-	-

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 6. 4 Marco Clase Dominio

MC Dominio Ranura	Tipo de ranura	Max/Min	Multiv	Propiedad General	Valor permitido	Valor omisión	Si necesito	Si modifico	Si Borro
Entidades	Conjunto de	1/n	No	_____	String	-	-	-	-

	caracteres								
Atributos	Conjunto de caracteres	1/n	Si	_____	String	-	-	-	-

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 6. 5 Marco Clase Requerimiento Funcional

MC Reque. funcional	Tipo de ranura	Max/Min	Multiv	Propiedad General	Valor permitido	Valor omisión	Si necesito	Si modifiko	Si Borro
Descripción	Conjunto de caracteres	1/1	No	_____	String	-	-	-	-
Tipo	Conjunto de caracteres	1/5	No	_____	String	-	-	-	-

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 6. 6 Marco Clase Restricciones de diseño

MC Restricción de diseño	Tipo de ranura	Max/Min	Multiv	Propiedad General	Valor permitido	Valor omisión	Si necesito	Si modifiko	Si Borro
Lenguajes	Conjunto de caracteres	1/n	Si	_____	["Software libre", "propietario", "Ambas"]	-	-	-	-
Bases de datos	Conjunto de caracteres	1/1	Si	_____	["Si", "No"]	-	-	-	-

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 6. 7 Marco Clase Alcance del Sistema

MC Alcance del Sistema	Tipo de ranura	Max/Min	Multiv	Propiedad General	Valore permitido	Valor omisión	Si necesito	Si modifiko	Si Borro
Acceso al Sistema	Conjunto de caracteres	1/n	No	_____	["Dentro de la empresa", "Internet", "Dispositivos móviles"]	-	-	-	-

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 6. 8 Marco Clase Requerimiento no funcional

MC Reque. no funcional	Tipo de ranura	Max/Min	Multiv	Propiedad General	Valor permitido	Valor omisión	Si necesito	Si modifiko	Si Borro
Interfaces Externas	Conjunto de caracteres	1/1	No	_____	["Servicios Web", "Archivos"]	-	-	-	-

					Planos", "Tramas", "Clonando estructura" , Consultas"]				
Disponibilidad	Numérico	1/1	No	_____	[1-8[-	-	-	-
Seguridad	Numérico	1/1	No	_____	[1-4[-	-	-	-
Concurrente	Numérico	1/1	No	_____	[1-50[-	-	-	-
Capacidad	Numérico	1/1	No	_____	[1- 10[-	-	-	-

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Tabla 6. 9 Marco Clase Requerimiento no funcional

MC Documento de Reque.	Tipo de ranura	Max/Min	Multiv	Propiedad General	Valor permitido	Valor omisión	Si necesito	Si modifico	Si Borro
Nombre	Conjunto de caracteres	1/1	No	_____	String	-	-	-	-
Requerimiento	Marco	1/1	No	_____	String	-	-	-	-
Análisis de Dominio	Marco	1/n	No	_____	String	-	-	-	-

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

6.3 Herramienta utilizada para el Sistema Experto

La herramienta utilizada para el desarrollo del sistemas expertos es CLIPS (Language Integrated Production System), que permiten construir bases de hechos y de conocimiento, a través de reglas de inferencias. Esta herramienta está diseñada para el desarrollo de software que requiere de conocimiento humano.

Con CLIPS se puede desarrollar software formado sólo por reglas, sólo por objetos, o mezcla de reglas y objetos (Molinero).

El modo de representar conocimiento heurístico en CLIPS es mediante reglas, las reglas tienen una estructura IF-THEN. El desarrollo del Sistema Experto realiza las inferencias con encadenamiento hacia delante, esto quiere decir que parte de hechos para llegar a resultados.

La estructura de una regla en clips está representada en el cuadro 6.10.

Tabla 6. 10 Formulación de reglas en Clips

```

(defrule <nombre-regla>

[<documentación opcional>

[(declare (saliencia <num>))]

(patrón 1)

(patrón 2)

(patrón N)

=>

(acción 1)

(acción 2)

...

(acción N)

```

Tabla 6. 11 Ejemplo de una reglas en Clips para el Sistema Experto

```

(defrule Req-no-func-Lenguaje-BDD-Internet

  (Es Herramienta ?ent1)

  (Es Computadoras ?ent2)

  (Es BaseDatos ?ent3)

  (Es Internet ?ent4)

=>

  (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))))))

```

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

En conclusión la etapa de Formalismo permite representar conocimientos del mundo exterior, para la cual se debe elegir técnicas que sean flexibles para simbolizar el esquema mental de los expertos. El Sistema Experto utiliza

conocimientos declarativos para resolver los problemas, a través de la Heurística y los metaconocimientos.

CAPÍTULO VII

SELECCIÓN DE LAS HERRAMIENTAS E IMPLEMENTACIÓN DEL SISTEMA

7.1 INTRODUCCIÓN

Para el desarrollo de Sistemas Expertos se puede utilizar cualquier lenguaje de programación, sin embargo es necesario seleccionar herramientas apropiadas que permitan representar el conocimiento. La selección de la herramienta debe proporcionar los formalismos de representación en los cuales puede codificarse la base de conocimientos, así como los mecanismos de inferencia y control para llevar a cabo el problema a resolver.

7.2 Selección de la Herramienta

Existen algunas herramientas para el desarrollo de Sistemas Expertos entre ellas se puede mencionar a las que más difusión han tenido. La tabla 7.1 describe algunas herramientas con las características más relevantes.

Tabla 7. 1 Herramientas para desarrollo de sistemas expertos

Lenguaje	Características
Lisp	Definido en 1958 por McCarthy. Este lenguaje es uno de los más utilizados. Pero a pesar de ser un lenguaje muy potente presenta ciertas deficiencias como el de no poseer un motor de inferencias propio, ni interfaces con el usuario, lo que lo hace más lento.
Prolog	Tiene su aparición en 1974 gracias a los trabajos de A. Colmerauer y P. Rousset. Utilizado para la demostración de teoremas, es un lenguaje sencillo que incorpora un motor de inferencias. Su principal aplicación es el manejo de bases de datos relacionales. Las deficiencias que presenta es el de no poder combinar programación convencional con la orientada a la lógica de predicados, razón por la cual no se lo ha incorporado en aplicaciones grandes y mucho menos en aplicaciones de control en tiempo real.
Smalltalk	Es un lenguaje orientado a objetos, no es uno de los lenguajes muy utilizados por la calidad y la potencia del lenguaje, por lo que la aplicación en desarrollos industriales y

	comerciales es baja.
C	Es uno de los lenguajes más antiguos que está siendo utilizado también para aplicaciones de inteligencia artificial, especialmente para aquellas aplicaciones que necesitan velocidad y grandes dimensiones de memoria.
Clips	Los creadores de CLIPS es la NASA, es una herramienta para la construcción de sistemas expertos, la sintaxis que utiliza es similar a la de Lisp, diseñada para el desarrollo de software que requiere de conocimiento humano, inicialmente fue creada para reglas y en versiones actuales opta por definir conocimientos mediante otros formalismos. CLIPS incorpora un completo lenguaje orientado a objetos (COOL) para la elaboración de sistemas expertos. Esta herramienta posee una amplia documentación por ser de dominio público, esto permite trabajar para aplicaciones con bajos costos. La mayor deficiencia de clips es que no permite una interface amigable para el usuario.

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

La herramienta seleccionada es Clips (C Language Integrated Production System) por que presenta características adaptables al proyecto de investigación como se describe a continuación:

- 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 obtiene 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.
- Actualmente es muy utilizado en la industria, Gobierno y en la educación.

En este contexto el Sistema Experto fue desarrollado en Clips con la implementación de otras herramientas que permitan una mejor interface con el usuario, por lo que la aplicación se encuentra embebida con un Sistema de Gestión y un Sistema Experto que realiza las inferencias a través de una base de conocimiento.

El Sistema de Gestión, utiliza una arquitectura de Tres Capas, por lo tanto para su funcionamiento es necesario tener operativo un servidor Web que soporte la ejecución de scripts PHP y un servidor de base de datos de MySQL.

La recomendación es instalar el servidor HTTP de Apache como servidor WEB y configurar la ejecución de archivos PHP con su módulo de conexión a MySQL habilitado, es conveniente por que se maneja sobre un entorno de software libre, y las herramientas ofrecen un buen soporte. Apache como servidor Web es más robusto, los archivos del servidor son PHP y Mysql es un buen gestor de bases de datos. Los 3 programas pueden encontrarse en versiones de código libre los que pueden ser descargados e instalados individualmente. A continuación se presenta un detalle de las aplicaciones y el enlace web de sus respectivos proyectos:

- **Apache HTTP server**
 - <http://httpd.apache.org/>
- **MySQL Community Edition**
 - <http://www.mysql.com/downloads/>
- **PHP**
 - <http://windows.php.net/download/>



El Sistema se vincula con el software de Sistemas Expertos CLIPS mediante un envoltorio Java. Para que esto se dé, es necesario que el equipo servidor del sistema sea capaz de ejecutar Java. Si el equipo no puede hacerlo es necesario instalar la versión más reciente de JRE (Java Runtime Environment). Se puede descargar desde el siguiente enlace:

<http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html>

La instalación de JRE es sencilla y directa. Basta con ejecutar el instalador y seguir las instrucciones.

7.3 Interfaz con el usuario e implementación

El Sistema de Gestión de Proyectos consta de los siguientes módulos.

- Gestión de Usuarios
- Mantenimiento de Catálogos
- Administración de Proyectos

El Sistema experto realiza inferencias sobre análisis de dominios, requerimientos funcionales y no funcionales.

La salida del Sistema experto es un documento en PDF con la especificación del estándar IEEE std 830, en el que se incluye inferencias sobre el análisis de dominio y tipificación de requerimientos.

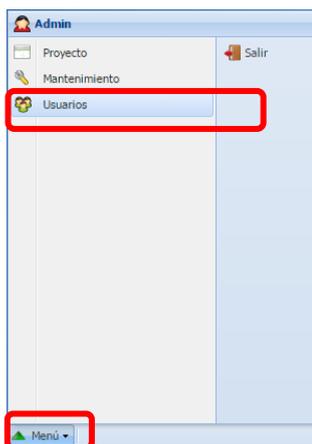
A continuación se da una guía para su utilización.

7.3.1 GESTIÓN DE USUARIOS

Este módulo permite crear, editar y eliminar usuarios del sistema. Para cada uno se define un perfil de acceso. Para acceder a este módulo es indispensable tener perfil de administrador. La figura 7.1 muestra la pantalla principal del Sistema.

La ruta de acceso al módulo es:

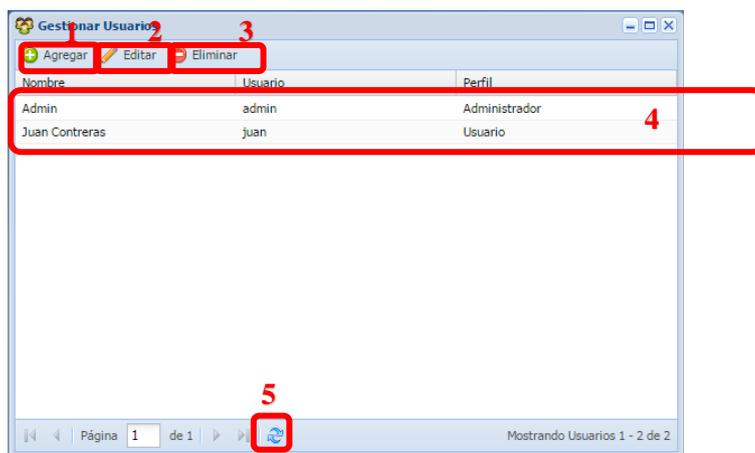
Menú -> Usuarios



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 1. Pantalla principal

La pantalla gestión de usuarios permite realizar operación sobre usuarios que podrán tener acceso al sistema como lo muestra la Figura 7.2



1. Agregar nuevos usuarios.
2. Editar usuarios existentes
3. Eliminar usuarios
4. Lista de usuarios
5. Volver a cargar usuarios.

Elaborado por: Corral, María Alexandra, Carrillo José Luis

Figura 7. 2.Pantalla Gestión de Usuarios

A partir de la figura 7.3 hasta la figura 7. 5 se indica las acciones que se pueden realizar en el módulo de gestión de usuarios. Las entradas presentadas con asteriscos representan entradas obligatorias para el Sistema de Gestión.



Editar / Crear Usuario

Usuario:* Usuario

Contraseña:* Contraseña

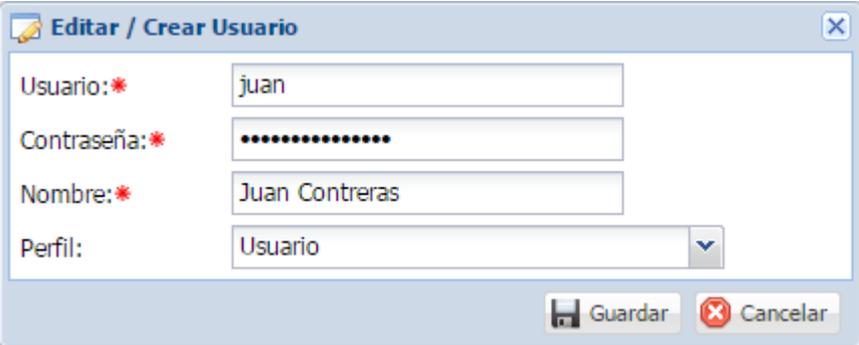
Nombre:* Nombre

Perfil: Usuario

Guardar Cancelar

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 3. Crear Usuario



Editar / Crear Usuario

Usuario:* juan

Contraseña:*

Nombre:* Juan Contreras

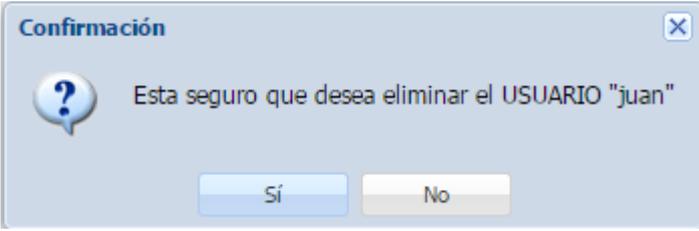
Perfil: Usuario

Guardar Cancelar

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 4. Modificar Usuario

La pantalla siguiente permite eliminar un usuario previamente seleccionado.



Confirmación

Esta seguro que desea eliminar el USUARIO "juan"

Sí No

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 5. Eliminar Usuario

Las opciones son:

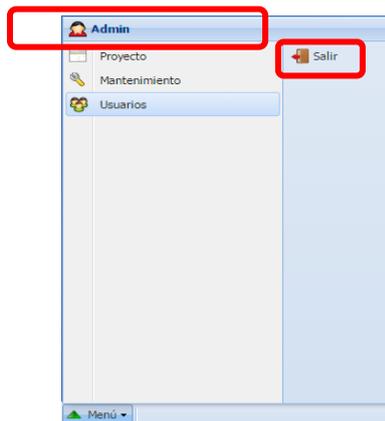
- **SI:** Elimina permanentemente el usuario de la base de datos.
- **No:** Cancela la eliminación del usuario y regresa a la ventana principal de usuarios.

7.3.2 MANTENIMIENTO DE CATÁLOGOS

Este módulo permite crear, editar y eliminar catálogos de Plataforma, Lenguaje de programación, Arquitectura, Seguridad y Topología. Para acceder a este módulo es indispensable tener perfil de administrador.

La figura 7.6 muestra la ruta de acceso al módulo.

Menú -> Mantenimiento



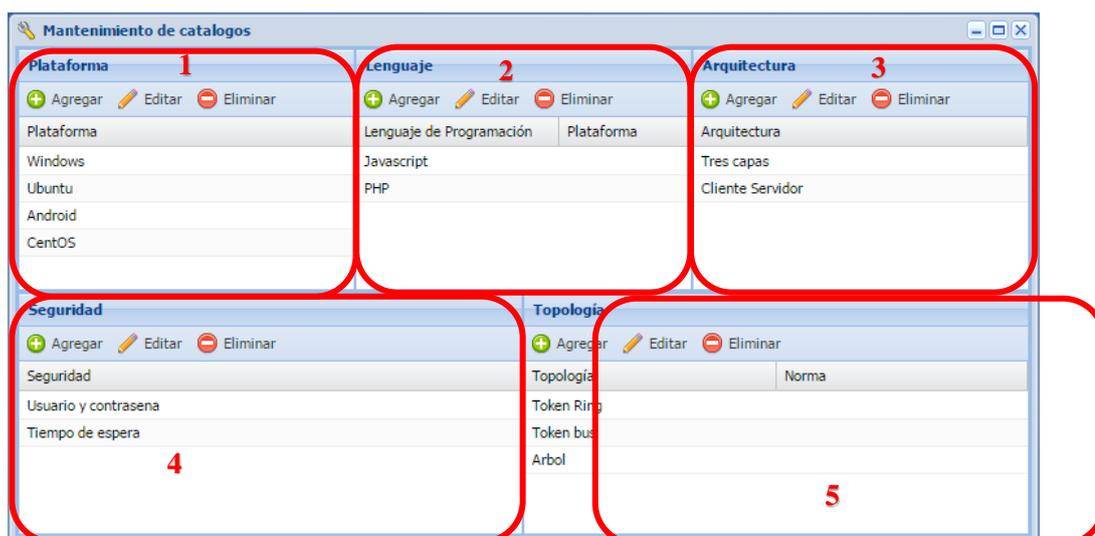
Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 6. Pantalla de acceso

El mantenimiento de catálogos se la da en base a la siguiente lista:

1. Mantenimiento de plataforma.
2. Mantenimiento de lenguaje de programación.
3. Mantenimiento de arquitectura.
4. Mantenimiento de seguridad.
5. Mantenimiento de topología.

La figura 7.7 muestra la pantalla principal del módulo de mantenimiento de catálogo



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

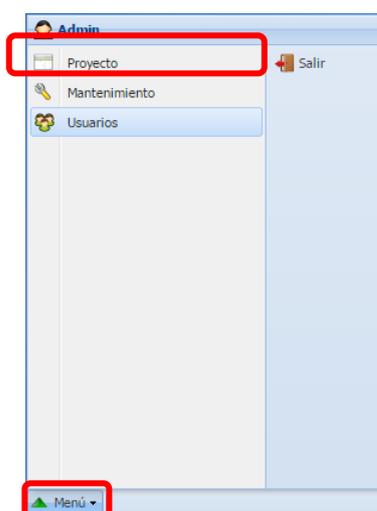
Figura 7. 7. Pantalla Principal Mantenimiento de Catálogos

7.3.4 ADMINISTRACIÓN DE PROYECTOS

Este módulo representa la parte medular de la aplicación. Permite crear, editar y eliminar los diferentes proyectos que generarán el documento de “Especificación de Requisitos de Software” del sistema. Este módulo puede ser accedido por todo tipo de usuario, tanto Administrador como Usuario general.

La ruta de acceso al módulo se muestra en la figura 7.8

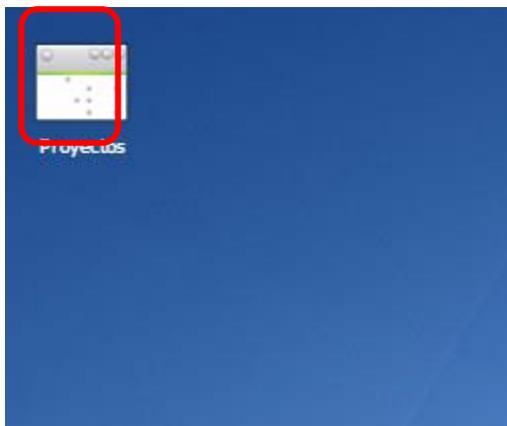
Menú -> Proyecto



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 8. Pantalla de acceso a Gestión Proyectos

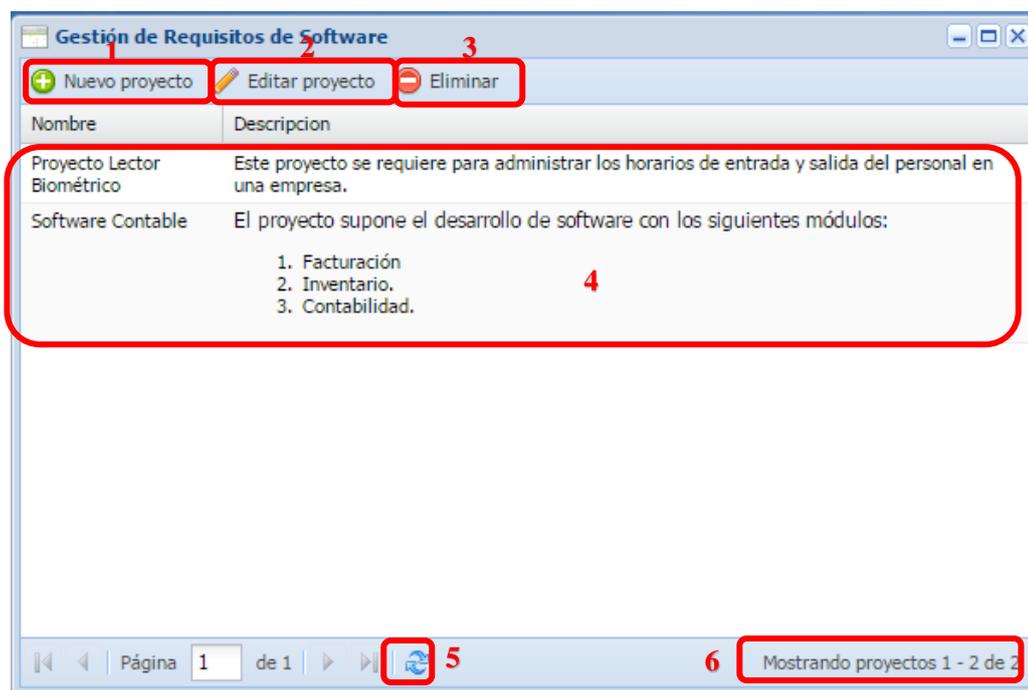
También se lo puede acceder dando un clic sobre su icono de acceso directo en el escritorio. Como lo indica la figura 7.9



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 9. Icono de acceso al escritorio

La figura 7.10 describe las acciones de gestión que se pueden realizar sobre el módulo de proyectos.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 10. Pantalla principal Gestión de Proyectos

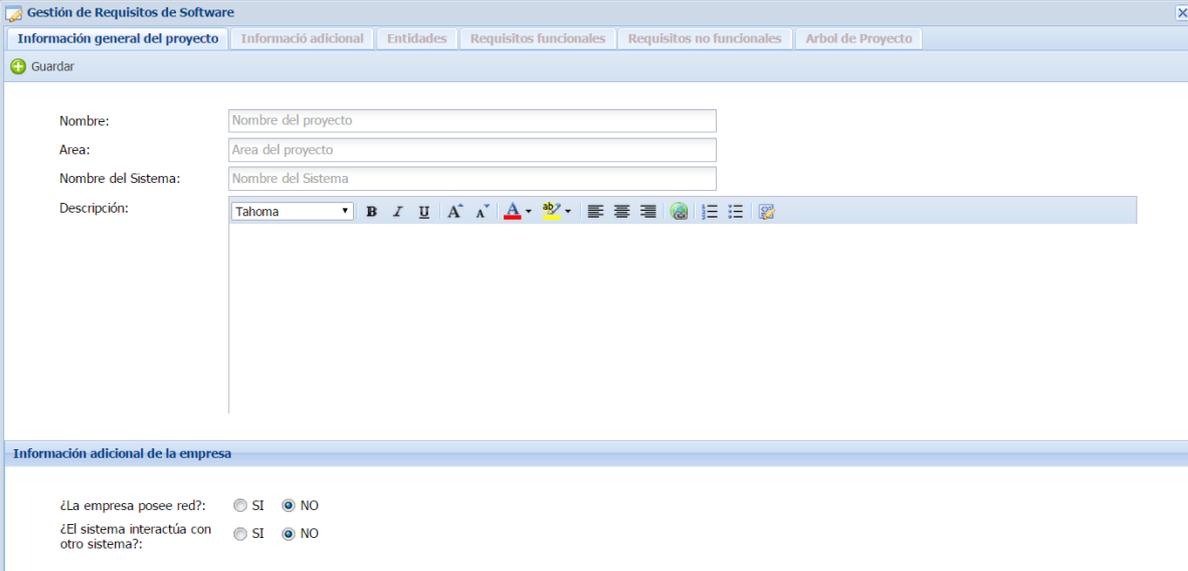
Las acciones principales para la gestión de proyectos se detallada en la siguiente lista.

1. Agregar nuevo proyecto.
2. Editar proyecto existente.
3. Eliminar proyecto.
4. Lista de proyectos.
5. Volver a cargar lista de proyectos.
6. Información sobre número de proyectos.

La descripción del proyecto es un campo creado para indicar brevemente de que se trata el mismo. Es importante recalcar que la fila del proyecto será tan grande como grande será su descripción, es por eso que se recomienda ingresar información concisa en este campo.

Tanto “Nuevo Proyecto” como “Editar Proyecto” abrirán una ventana de edición la cual ocupará toda la pantalla (incluyendo el espacio del menú principal). Por ello, si se cuenta con un perfil Administrador es importante revisar que todos los catálogos estén llenados correctamente antes de realizar la creación / edición de proyectos y así evitar tener que cerrar la ventana de edición porque faltó información en algún catálogo.

La figura 7.11 muestra información relevante sobre los campos de información que son requeridos para el sistema



The screenshot shows a web application window titled "Gestión de Requisitos de Software". The main content area is titled "Información general del proyecto" and contains a "Guardar" button. Below the button are four input fields: "Nombre:" (containing "Nombre del proyecto"), "Area:" (containing "Area del proyecto"), "Nombre del Sistema:" (containing "Nombre del Sistema"), and "Descripción:" (containing "Tahoma" and a rich text editor toolbar). At the bottom, there is a section titled "Información adicional de la empresa" with two radio button questions: "¿La empresa posee red?" (with "NO" selected) and "¿El sistema interactúa con otro sistema?" (with "NO" selected).

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 11. Información de un nuevo proyecto

La pantalla de creación / edición de nuevos proyectos consta de 6 secciones que son:

- **Información general del proyecto:** Datos generales del proyecto que incluyen Nombre del proyecto, Área, Nombre del Sistema y Descripción del proyecto.
- **Información adicional:** Datos adicionales del proyecto como Acrónimos, Abreviaturas y Definiciones.
- **Entidades:** Herramienta gráfica que permite definir un modelo Entidad -Relación con todos los elementos con los que consta el proyecto y sus respectivas relaciones.
- **Requisitos funcionales:** Información sobre necesidades operativas del proyecto.
- **Requisitos no funcionales:** Información sobre factores externos que deben considerarse en el desarrollo del proyecto.
- **Árbol del proyecto:** Denominado así porque contiene una vista de árbol con cada sección del documento de Especificación de Requisitos de Software. Desde aquí se consolida la creación del documento e incluso se edita cada parte del mismo. Además permite la creación de anexos y la exportación a PDF.

Como se presenta en la figura 7.11, al crear un nuevo proyecto todas las secciones están deshabilitadas a excepción de la primera sección. Esto se debe a que todas las secciones han de pertenecer a un proyecto con información básica, de manera que solo cuando se guarde esta información en la primera pestaña, las demás pestañas se habilitaran automáticamente.

Como ya se ha visto, en la figura 7.11 para la Gestión de Requisitos cuenta con 6 secciones, pero a diferencia de la pantalla de Creación, la de Edición automáticamente muestra todas las secciones habilitadas para edición.

Se puede navegar en cada una de ellas y la modificación se realiza directamente seleccionando el campo a editar y dando un nuevo valor. Al final de cada edición en cada sección es importante “Guardar”.

Se debe recalcar sobre el hecho de la independencia entre secciones al momento de guardar, es decir, si se ha realizado cambios a la sección “Información general del proyecto” y a la sección “Entidades”, pero solo se ha dado clic en el botón “Guardar” de “Entidades”, entonces la única información actualizada en base de datos es la de “Entidades”. Esto se debe a que el proceso de almacenamiento de cambios se hace independiente en cada pestaña y no una general por el proyecto. Esta funcionalidad permite al sistema tener un mayor rendimiento ya que evita gestionar un gran volumen de información al servidor cuando se deben dar modificaciones pequeñas (que será el escenario frecuente).

A continuación se detallaran cada una de las secciones que tiene los proyectos y la información que deberían registrarán en cada una de ellas, además de políticas de uso y restricciones en caso de aplicar.

Información General de Proyectos

Este módulo permite ingresar información general relacionada con el proyecto. La pantalla de esta sección se indica en la figura 7.12.

The screenshot shows a web application window titled "Gestión de Requisitos de Software". The main content area is titled "Información general del proyecto" and contains a "Guardar" button and four input fields: "Nombre:" (with placeholder "Nombre del proyecto"), "Area:" (with placeholder "Area del proyecto"), "Nombre del Sistema:" (with placeholder "Nombre del Sistema"), and "Descripción:" (with a rich text editor toolbar and placeholder "Tahoma"). Below this is a section titled "Información adicional de la empresa" containing two radio button questions: "¿La empresa posee red?" and "¿El sistema interactúa con otro sistema?", both with "SI" and "NO" options.

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 12. Información del Proyecto

Los campos de la sección son:

- **Nombre del proyecto:** Cadena alfanumérica que identifica al proyecto.
- **Área:** Cadena alfanumérica que define el área a la que pertenece el proyecto.
- **Nombre del Sistema:** Cadena alfanumérica que establece un nombre al sistema informático resultante del proyecto (generalmente el nombre del sistema es una abreviatura del nombre del proyecto).
- **Descripción:** Una cadena alfanumérica que describe de forma completa y breve las funciones del sistema. El editor permite crear texto enriquecido (formato de texto, numeración, viñetas, etc.).

En la parte inferior se encuentra una subsección llamada "Información adicional de la empresa" indicada en la figura 7.13. Esta sección incluye 2 preguntas relacionados con los aspectos no funcionales del proyecto pero competentes a la empresa donde se implementará. Las preguntas son:

- **¿La empresa posee red?** Si la empresa posee una red informática de cualquier tipo esta pregunta debe responder como SI. En ese caso aparecerá un nuevo campo:
 - **Seleccione la arquitectura:** Como la empresa si tiene red, se debe indicar en qué arquitectura trabaja. Las opciones son tomadas directamente del catálogo “Arquitectura”.
 -

Información adicional de la empresa

¿La empresa posee red?: SI NO

¿El sistema interactúa con otro sistema?: SI NO

Seleccione la arquitectura: Cliente/Servidor

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 13. Información adicional de la Empresa

- **¿El sistema interactúa con otros sistemas?** En caso de que el sistema a desarrollarse a partir de este proyecto deba interactuar con otros sistemas de la empresa o sistemas externos a la empresa, aquí se debe seleccionar SI.

Información Adicional

Este módulo permite ingresar información adicional sobre el proyecto. Específicamente Definiciones, Acrónimos y Abreviaturas. La pantalla principal del módulo se muestra en la figura 7.14.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

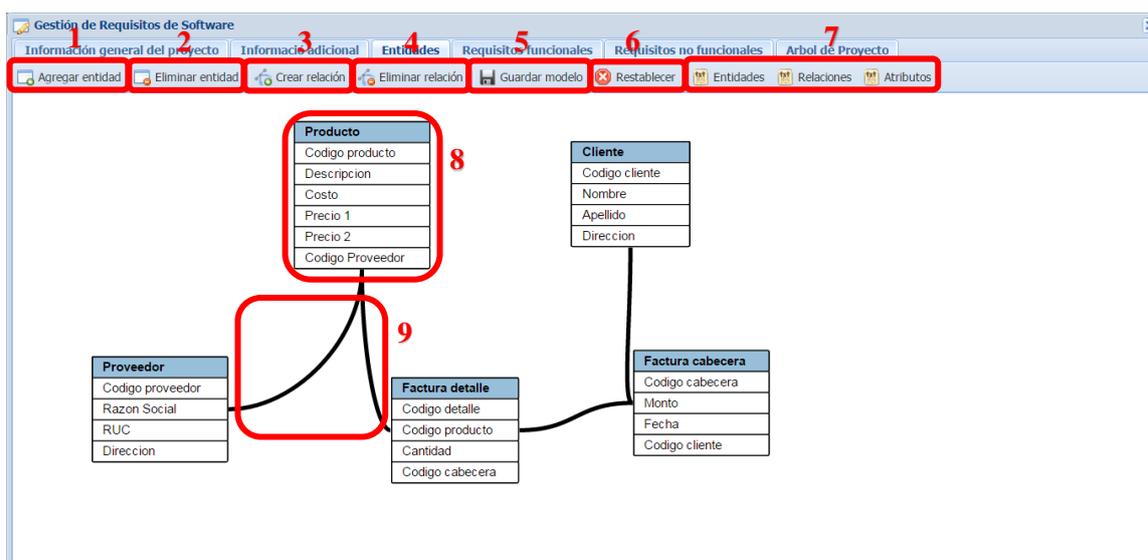
Figura 7. 14. Pantalla Información Adicional

1. Definiciones.
2. Acrónimos.
3. Abreviaturas.

Entidades

Este módulo permite interactuar con la administración de entidades de forma gráfica. Esta sección es una de las más importantes ya que aquí se describe un modelo conceptual del sistema a desarrollar, permitiendo tener una visión general del problema, y a partir del cual generará módulos de gestión por cada entidad

La pantalla principal de esta sección se muestra en la figura 7.15.

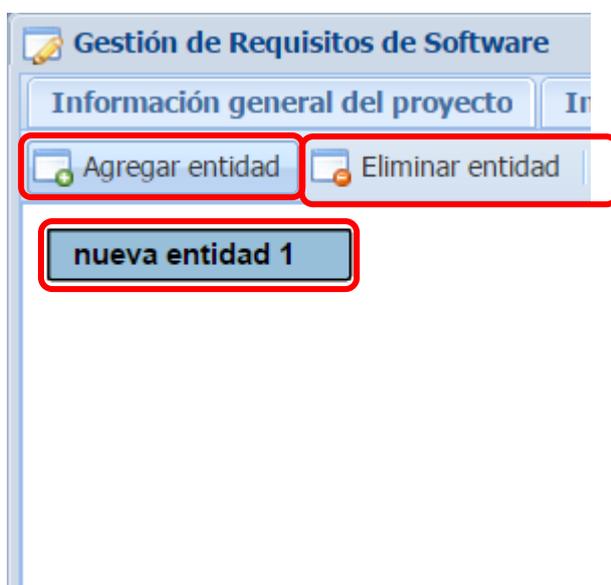


Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 15. Administración de entidades

Agregar entidad.

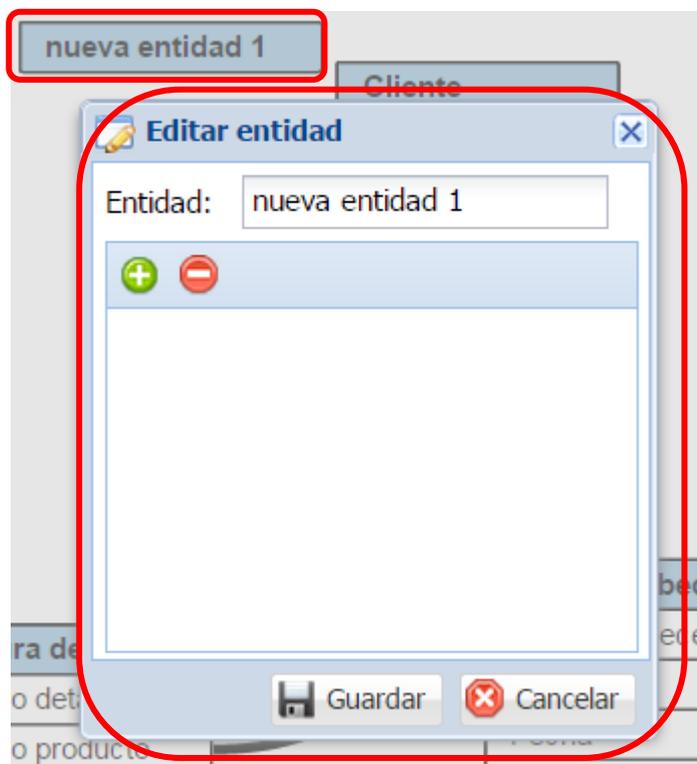
Esta opción permite agregar una entidad al área del modelo gráfico. Al seleccionar esta opción  automáticamente se creará una entidad con un nombre secuencial y se posicionará en la esquina superior izquierda del área del modelo gráfico, como lo muestra la figura 7.16.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 16. Agregar entidad

Para editar el contenido de nuevas entidades y cualquier entidad existente se debe dar doble clic en la entidad. Eso abrirá una pequeña ventana de edición, como lo indica la figura 7.17.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

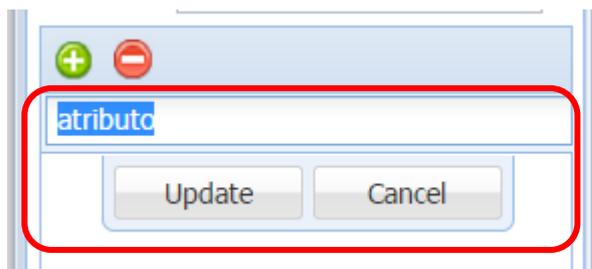
Figura 7. 17. Edición de Entidades

En esta ventana podemos editarlos valores de:

- **Entidad:** El nombre de la entidad. No puede ser el mismo que otras entidades.

Las acciones que se pueden realizar son:

- **+ Agregar atributo:** Los atributos son características propias de cada entidad. La figura 7.18 indica como agregar un atributo.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 18. Agregar atributo

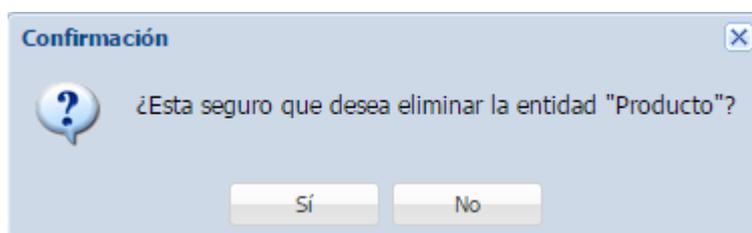
Ahí podremos escribir lo que deseemos y seleccionar una de las 2 opciones:

- **Update:** Registrar el valor como atributo de la entidad.
- **Cancel:** Cancela el cambio de valor de atributo. Finaliza el método de edición de atributo.

Eliminar entidad

Al seleccionar esta opción el indicador permanece activado, indicando que en ese momento usted se encuentra utilizando la herramienta de eliminación de entidades.

Mientras usted se encuentre con esa herramienta habilitada, cada entidad seleccionada se eliminará. Antes de esta acción, un mensaje de confirmación se mostrará como se indica en la figura 7.19.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 19. Eliminar entidad

Este mensaje presentará 2 opciones:

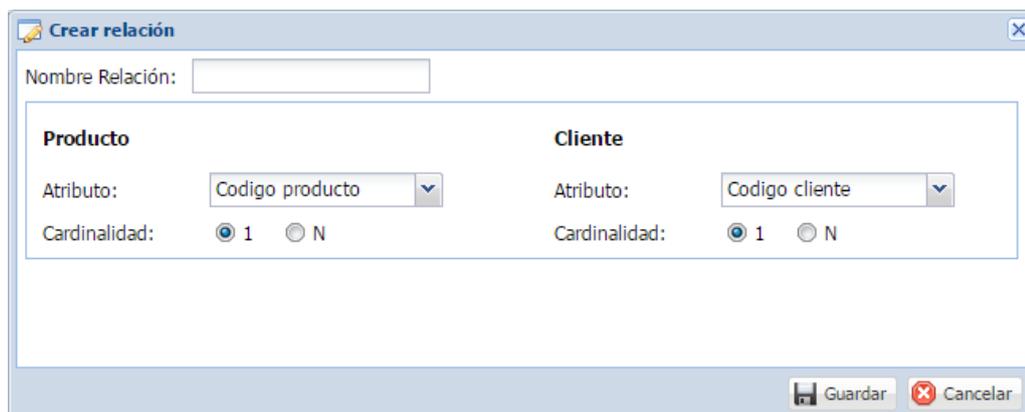
- **Sí:** La entidad se elimina y con ella cualquier relación que haya involucrado a la misma. La ventana de confirmación se cierra.
- **No:** El proceso de eliminación se cancela, la ventana de confirmación se cierra.

Es importante mencionar que, aunque una entidad haya sido eliminada, la opción de “Eliminar entidades” seguirá seleccionada. Esto en apoyo a la eliminación múltiple, ya que así el trabajo se simplifica y se evita estar presionando el botón de eliminar cada vez.

1. Crear relaciones.

La funcionalidad para crear funciones requiere que se indiquen las 2 entidades que se desean relacionar, es decir, al seleccionar la función de relaciones el sistema espera que se den 2 clics más, uno en cada entidad que se va a relacionar. La orden de selección en las entidades es irrelevante.

Al seleccionar las 2 entidades, aparecerá una pantalla como lo describe la figura 7.20.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

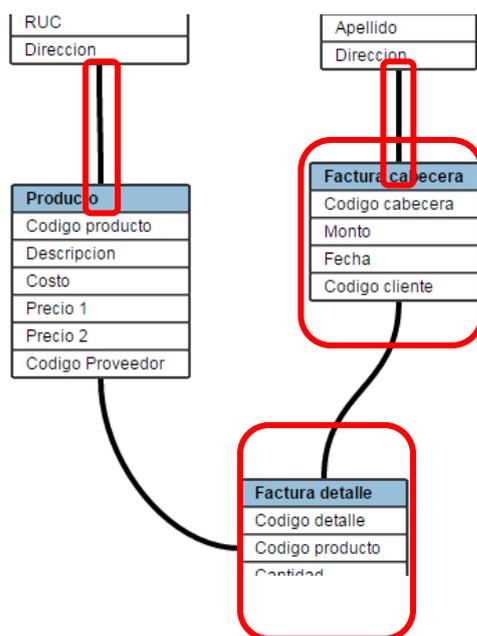
Figura 7. 20. Creación de relaciones

Este formulario de relaciones contiene los siguientes campos:

- **Nombre relación:** Cadena alfanumérica que representa el nombre de la relación que se está creando. Puede contener espacios.

- **Atributo:** Este campo aparece dos veces, cada una bajo un nombre de las 2 entidades. Este campo es un valor que se selecciona de entre los atributos de la entidad en cuestión y representa el campo a través del cual se vincularán o relacionarán las 2 entidades.
- **Cardinalidad:** Es la propiedad que determina cuantos elementos de una entidad pueden interactuar con cuantos elementos de la otra entidad. Generalmente las opciones son 1 (solo interactúa con 1 elemento) o N (interactúa con 2 o más).

En la figura 7.21 se visualiza las relaciones creadas y el elemento gráfico que representa la misma (línea negra gruesa entre las entidades de la relación).

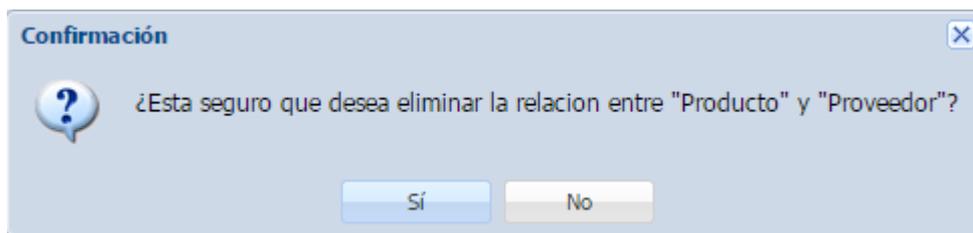


Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 21. Gráfica del modelo conceptual con relaciones

Eliminar relaciones

Mientras esté habilitada la herramienta, cada relación (línea negra gruesa seleccionada se eliminará. Antes de esta acción, un mensaje de confirmación se mostrará cómo se presenta en la figura 7.22.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 22. Eliminación de Relaciones

Este mensaje presentará 2 opciones:

- **Sí:** La relación se elimina de la base de datos y del modelo gráfico. La ventana de confirmación se cierra.
- **No:** El proceso de eliminación se cancela, la ventana de confirmación se cierra.

Es importante mencionar que, aunque una relación haya sido eliminada, la opción de “Eliminar relación” seguirá seleccionada. Esto en apoyo a la eliminación múltiple, ya que así el trabajo se simplifica y se evita estar presionando el botón de eliminación cada vez.

Guardar modelo

Al Guardar el modelo, el sistema iniciará un proceso que convierte el modelo gráfico en registros de base de datos (entidades, atributos, relaciones, etc.). Las posiciones de los elementos en el modelo también son registradas, esto para mantener la organización visual que se desee.

Este botón permite además guardar cambios en el modelo. Su proceso es similar al de guardado inicial, sin embargo la modificación evita entidades y relaciones existentes y agrega o elimina elementos en base a la modificación realizada.

1. Restablecer

Al seleccionar ésta opción , el sistema iniciará un proceso de carga del modelo tal y como se registra en la base de datos. Este botón es útil en casos de modificaciones no deseadas o involuntarias ya que,

siempre que el modelo no haya sido **guardado** aún, puede retornar a su forma original.

Esta restauración incluye posición de los elementos dentro del modelo.

2. Generación de Entidades, Relaciones y Atributos

Al seleccionar entidades, Relaciones y Atributos (), el sistema genera y descarga un archivo de texto plano con la información de Entidades, Relaciones o Atributos, según se haya seleccionado la opción.

Este archivo generado puede ser descargado y guardado en el equipo que invoca el reporte.

3. Modelo gráfico de entidades

Las entidades mantienen un formato gráfico genérico que distribuye la información de la misma en una serie de filas encerradas en un rectángulo. La primera fila, de color diferente a las demás, representa el nombre de la entidad. Las filas siguientes representan los atributos de entidad.

El tamaño de la entidad, tanto en ancho como en largo, se define automáticamente y no son editables por el usuario. Estos se ajustan perfectamente para optimizar espacio mientras encierran perfectamente la información. Sin embargo es recomendable utilizar nombres de entidades y atributos pequeños en la medida de lo posible. Esto permitirá mantener un modelo legible y funcional.

4. Modelo de relaciones.

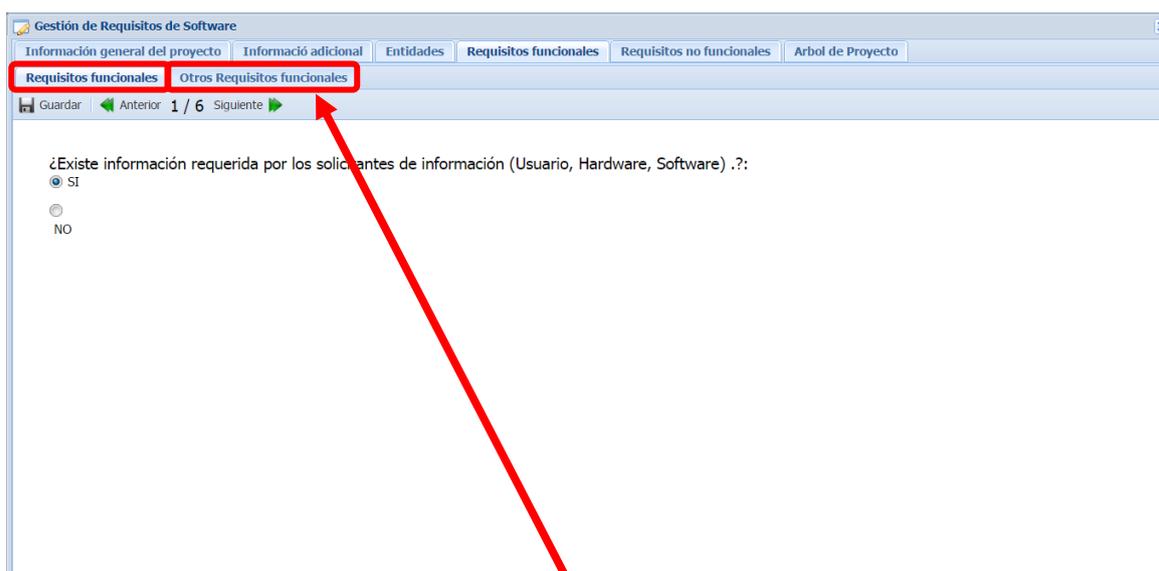
Como se ha explicado en el apartado de Relaciones, éstas se presentan gráficamente como líneas negras gruesas vinculando 2 relaciones. Cuando una entidad ligada mediante una relación es movida, la línea de relación la seguirá y se posicionará en cualquier

parte de la entidad que sea más conveniente para mantener un modelo visualmente atrayente.

Toda relación es editable. Para ello basta con dar doble clic en cualquiera de ellas para que aparezca un cuadro de dialogo con la información de la relación y la opción de edición o cancelación respectiva.

Requisitos funcionales

Este módulo define los requisitos funcionales que se aplican al proyecto actual. Al ingresar en esta sección se puede ver una serie de preguntas que inferirán en el sistema para requerimientos funcionales como lo describe la figura 7.23



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 23. Requerimientos funcionales

La ventana principal maneja 2 subsecciones.

1. Panel de navegación.

Este panel permite responder las 6 preguntas al navegar a través de ellas. Al seleccionar éstas opciones, la pregunta siguiente o precedente se mostrará luego de una elegante transición de movimiento.

2. Área de preguntas.

Esta área contiene todas las preguntas sobre requerimientos funcionales. Es importante leer y responder cada una de ellas ya que constituyen información importante para el procesamiento del módulo de Sistemas Expertos.

Otros requisitos funcionales

La figura 7.24 describe la subsección en la que se registran todos los requisitos funcionales adicionales que tiene el sistema. Por defecto cada entidad que se creó en el apartado anterior genera 5 requisitos funcionales: Alta, Baja, Modificación, Consulta general y Consulta específica. De igual forma que los módulos anteriores permite realizar acciones de mantenimiento como lo indica los numerales de la figura 7.23

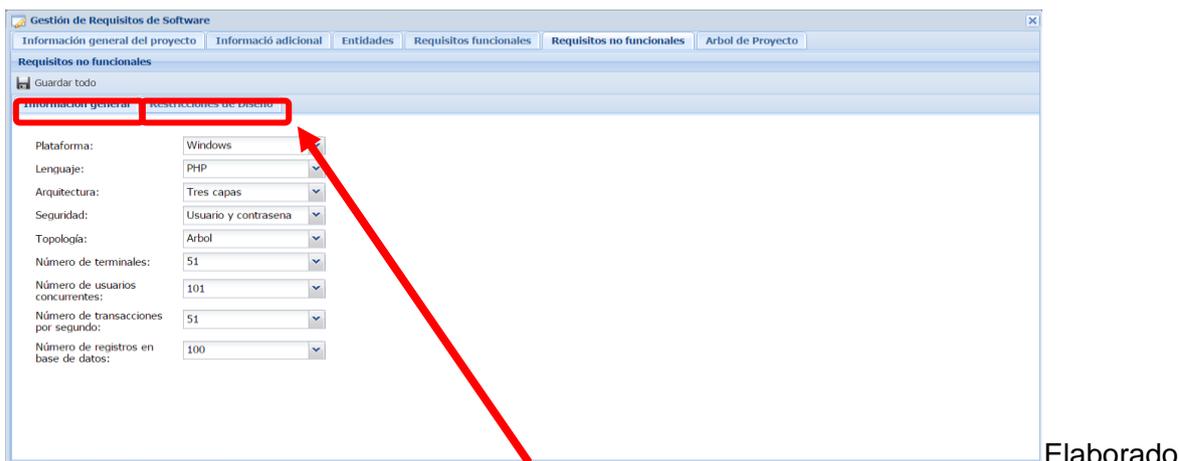
Tipo requerimiento	Entidad	Descripción requerimiento
Alta	Producto	
Baja	Producto	
Modificación	Producto	
Consulta general	Producto	
Consulta específica	Producto	
Alta	Cliente	
Baja	Cliente	
Modificación	Cliente	
Consulta general	Cliente	
Consulta específica	Cliente	
Alta	Proveedor	
Baja	Proveedor	
Modificación	Proveedor	
Consulta general	Proveedor	
Consulta específica	Proveedor	
Alta	Factura cabecera	
Baja	Factura cabecera	
Modificación	Factura cabecera	

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 24. Requisitos funcionales adicionales

Requisitos no funcionales

Este módulo define los requisitos no funcionales que se aplican al proyecto actual. La figura 7.25 describe la pantalla con atributos no funcionales.



Elaborado

por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 25. Requerimientos no funcionales

La ventana principal maneja 2 subsecciones.

Información general

Esta subsección registra información general de Plataforma, Lenguaje, Arquitectura, Seguridad, Topología, Número de terminales, Número de usuarios concurrentes, Número de transacciones por segundo y Número de registros en la base de datos. La pantalla principal del sub módulo se presenta en la figura 7.26

The screenshot shows the 'Información general' sub-section of the 'Requisitos no funcionales' form. The form fields are highlighted with red boxes and numbered 1 through 9:

1. Lenguaje: PHP
2. Arquitectura: Tres capas
3. Seguridad: Usuario y contraseña
4. Topología: Arbol
5. Número de terminales: No especificado
6. Número de usuarios concurrentes: No especificado
7. Número de transacciones por segundo: No especificado
8. Número de registros en base de datos: No especificado
9. (The entire form area is highlighted)

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 26. Información general Requerimientos no funcionales.

A continuación se describe cada uno de estos apartados.

1. Plataforma.

Aquí se define la plataforma que utilizará el sistema a implementar. Los datos son obtenidos del catálogo de plataformas.

2. Lenguaje

Aquí se define el lenguaje que utilizará el sistema a implementar. Los datos son obtenidos del catálogo de lenguajes.

3. Arquitectura

Aquí se define la arquitectura que utilizará el sistema a implementar. Los datos son obtenidos del catálogo de arquitecturas.

4. Seguridad

Aquí se define la seguridad que protegerá al sistema a implementar. Los datos son obtenidos del catálogo de seguridades.

5. Topología

Aquí se define la topología en la cual se implementará el sistema. Los datos son obtenidos del catálogo de topologías.

6. Número de terminales

Especifica cuantos equipos en promedio se manejarán con el sistema a implementar. Las opciones son:

- No especificada
- 1 – 50
- 51 – 100
- 101 – 150
- Más de 150

7. Número de usuarios concurrentes

8. Especifica cuantos usuarios al mismo tiempo pueden ocupar el sistema sin dar lugar a dificultades en el desempeño. Las opciones son:

- No especificada
- 1 – 100
- 101 – 500
- 501 – 1000
- Más de 1000
-

9. Número de transacciones por segundo

Especifica la velocidad de procesamiento del sistema medido en transacciones. Las opciones son:

- No especificada
- 1 – 50
- 51 – 100
- 101 – 150
- Más de 150
-

10. Número de registros en base de datos

Especifica cuantos registros pueden almacenarse en la base de datos sin que el desempeño del sistema se vea notablemente comprometido.

Las opciones son:

- No especificada
- 1 – 10.000
- 10.001 – 100.000
- 100.001 – 1.000.000
- Más de 1.000.000
-

Al finalizar la modificación, los cambio pueden ser guardados . Es importante recalcar que si no se guardan los cambios en este punto, los mismos pueden perderse.

Restricciones de diseño

En esta subsección se pueden proveer información sobre las restricciones externas que tiene el proyecto. Esta información se recopila en base a un cuestionario de varias preguntas, en las que el sistema permitirá inferir sobre requerimientos no funcionales. La figura 7.27 describe el formato de preguntas.

The screenshot shows a web-based form interface. At the top left, there is a button labeled 'Guardar todo' with a floppy disk icon. Below it, there are two tabs: 'Información general' and 'Restricciones de Diseño'. Under the 'Restricciones de Diseño' tab, there is a navigation bar with a left arrow, the text 'Anterior 1 / 8 Siguiente', and a right arrow. A red box highlights the 'Guardar todo' button, the navigation bar, and the question area below. The question area contains the text 'La política de Adquisición de Software para toda la empresa es:' followed by two radio button options: 'Software Libre' and 'Software Propietario'. A red '3' is placed to the right of the second option.

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 27. Restricciones de Diseño.

A continuación se describe cada subsección indicada en la figura 7.21

1. Botón Guardar todo.

EL botón Guardar todo  permite registrar (como primera vez o como modificación) la información provista en esta subsección.

2. Panel de navegación.

Este panel permite responder las 8 preguntas al navegar a través de ellas. Al seleccionar ésta opción, la pregunta siguiente o precedente se mostrará luego de una elegante transición de movimiento.

3. Área de preguntas.

Esta área contiene todas las preguntas. Es importante leer y responder cada una de ellas ya que constituyen información importante para el procesamiento del módulo de Sistemas Expertos.

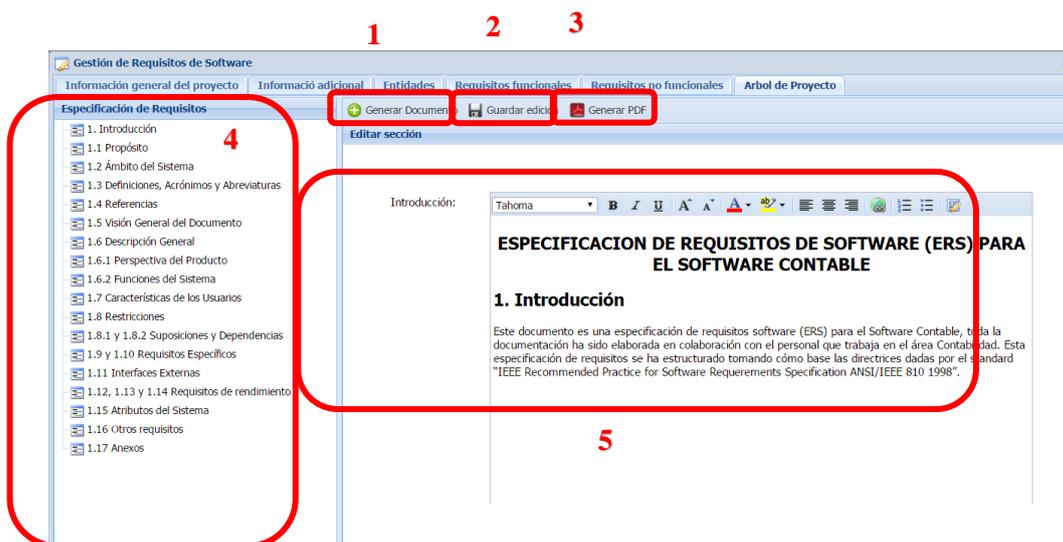
Árbol del Proyecto

Este módulo es el más importante del proyecto ya que es el encargado de generar el documento de Especificación de Requisitos de Software en su totalidad, en base a los datos ingresados en las secciones anteriores. El módulo en general tiene las funciones de:

- Generar el documento de ERS
- Permitir la revisión final del documento de ERS.
- Editar cada elemento del documento final.
- Crear, modificar y eliminar anexos.
- Exportar el documento a formato PDF

Cada parte del árbol de segmentos del ERS es editable en su totalidad. El formato tiene una estructura de la IEEE std 830.

La pantalla principal de la sección se describe en la figura 7.28.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 28. Árbol del documento de especificación de requerimientos

1. Generar documento.

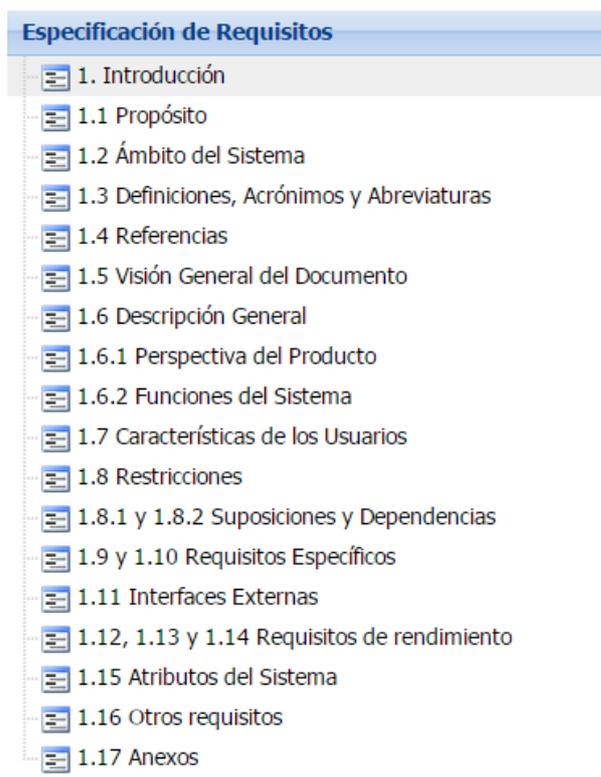
La opción Generar documento  es el alma del sistema ya que se genera una rutina que es la encargada de tomar toda la información registrada en el sistema y generar el documento final.

Esta generación de documento se hace siempre en base a la información proporcionada en los otros módulos y omite cualquier edición realizada directamente en los segmentos dentro de esta sección; en otras palabras, si se ha realizado modificaciones directas en algún segmento (más adelante se analiza cómo hacerlas), dichas modificaciones se perderán al seleccionar la opción Generar documento. Se recomienda poner mucho cuidado a este tema ya que el hecho de Generar el documento altera directamente los registros y por tanto elimina toda información editada por el usuario.

2. Guardar edición.

La opción Guardar edición  es la encargada de gestionar el registro de toda modificación directa que se realice al documento en cada una de sus segmentos. Para realizar la edición de un segmento puede seguir los siguientes pasos:

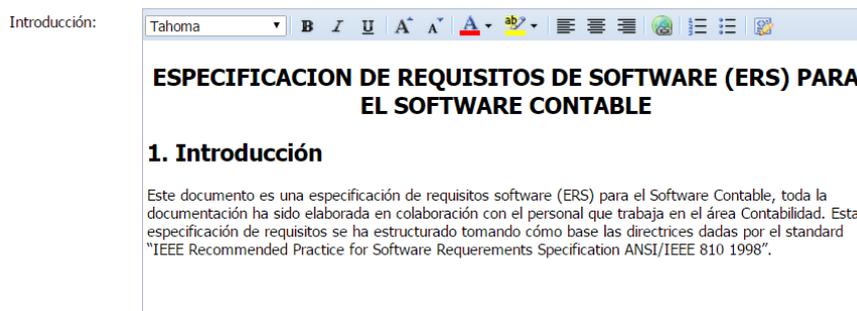
- a) **Seleccione el segmento que desea modificar.** Para ello basta seleccionar en cualquier segmento del panel izquierdo con el árbol de contenido del documento. Esto automáticamente mostrará el contenido de dicho segmento en el área de edición, como se indica en la figura 7.29.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 29. Área de edición

b) Modifique la información. Permite editar la información del segmento. Borrar, agregar o modificar son operaciones válidas en cualquier segmento. En la figura 7.30 se visualiza una parte de la salida del documento de Especificación de Requerimientos

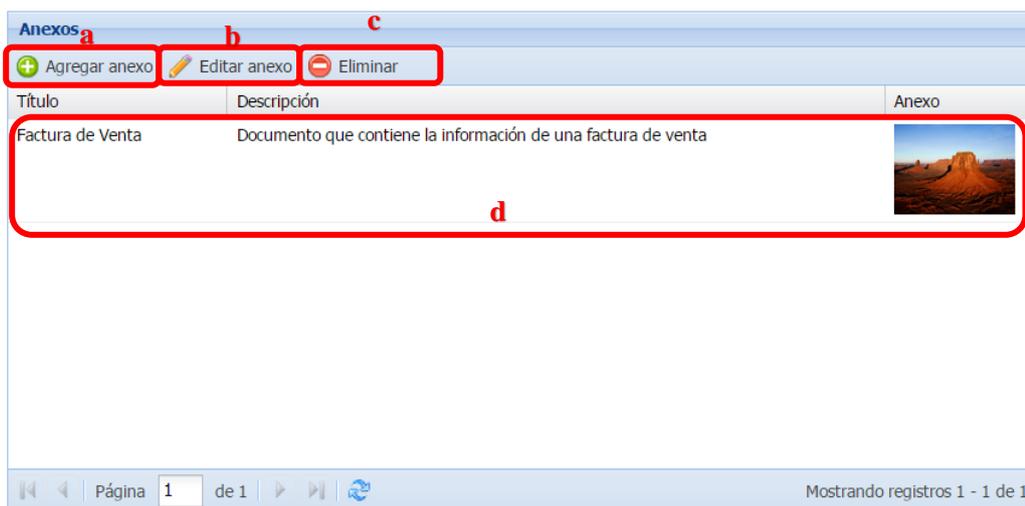


Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 30. Documento de Especificación de Requerimientos

Es indispensable luego de cualquier modificación el guardar el documento. Esto asegura la consideración de los cambios en la generación del archivo PDF.

Una sección que tiene gran importancia es “Anexos”. A continuación se presenta la pantalla principal como lo describe la figura 7.31.



Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 7. 31. Anexos del Documento de Requerimientos

3. Generar PDF.

La opción Generar PDF  permite crear el archivo PDF descargable al momento que finaliza la generación del mismo. Este documento contiene toda la información generada (y en ocasiones editada) del proyecto creado.

CAPÍTULO VIII

EVALUACIÓN DEL SISTEMA EXPERTO

8.1 INTRODUCCIÓN

Mostrar que un Sistema Experto en cierto sentido es "correcto" es una tarea crítica. Un sistema incorrecto puede cometer errores costosos, o puede no funcionar con las expectativas deseadas por el usuario, en cualquiera de los casos las decisiones generadas por el sistema pueden ser inapropiado, pudiendo provocar daños considerables, como la pérdida financiera o humana; por ejemplo, el Sistema Experto para el diagnóstico médico y sistemas de impuesto de renta han sido difíciles de implantar debido a las preocupaciones sobre la responsabilidad de los diagnósticos propuestos por el sistema (O'Keefe and O'Leary 1993).

La etapa de verificación y validación de un sistema experto es un proceso de retroalimentación que permite probar la exactitud y utilidad del programa. En el caso del sistema experto desarrollado, para asistir al ingeniero de software en la especificación de requerimientos, la etapa de verificación y validación se realizó utilizando la técnica de inspección formal, técnica que asegura que los requerimientos sean precisos y consistentes (Agarwal and Goel 2014) y en la que los usuarios e interesados estén de acuerdo con el documento generado: Plan de Especificación de Requerimientos.

Los sistemas expertos por ser modelos iterativos en cada fase del desarrollo se realizan evaluaciones específicas con el fin de comprobar su validez. En este capítulo se presentan los casos de prueba con los que se ha evaluado el sistema.

Para el desarrollo del sistema experto se ha utilizado la metodología ideal en el cual se encuentran inmersas las evaluaciones en cada fase de desarrollo, las que se describen a continuación:

- Evaluación de la Adquisición del conocimientos

- Evaluación de la Conceptualización de Conocimientos
- Evaluación de la formalización de conocimientos
- Evaluación de la Implementación de conocimientos.

Para el proceso de verificación y validación del sistema experto se han considerado dos tipos de usuarios, los expertos del dominio y los usuarios no expertos.

8.2 Evaluación de la Adquisición de los conocimientos.

La fase de adquisición de conocimientos permite educir los conocimientos del experto acerca del dominio de la aplicación para el desarrollo del Sistema Experto. Por lo que se realizó sesiones de trabajo entre el experto y el Ingeniero del Conocimiento, con el fin de adquirir conocimientos, mismos que deben ser: comprensibles, consistentes, completos y no redundantes.

La fase de adquisición se la llevó a cabo a través de entrevistas y análisis de textos. Para la evaluación de esta fase, en cada sesión de trabajo se realizó una retroalimentación de conceptos, definiciones y casos de estudio. (Para más detalle ver capítulo IV).

8.3 EVALUACIÓN DE LA CONCEPTUALIZACIÓN DE CONOCIMIENTOS.

Para la evaluación de la conceptualización de conocimientos hemos utilizado cinco tipos de comprobación:

- Comprobación de los conocimientos estratégicos en la que se evalúa el modelo de descomposición funcional mediante de revisiones parciales del experto.
- Comprobación de los conocimientos tácticos, para lo cual se revisaron en detalle las seudoreglas en las revisiones parciales a través de la aceptación del experto.
- Comprobación de los conocimientos fácticos, para lo cual se revisaron glosario de términos, diccionario de conceptos, tabla de concepto-atributo-valor y definición de atributos.

- Comprobación del modelo dinámico y estático, se revisaron y aceptaron los modelos por el experto.
- Comprobación de la fase de conceptualización, se seleccionó un caso de prueba el mismo que están referenciados en la evaluación de la implementación de conocimientos.

8.4 Evaluación de la Formalización de Conocimientos

En ésta evaluación se realizó la formalización de conocimientos con respecto a la conceptualización de los mismos, seleccionándose los mismos casos de prueba usados en la comprobación de la fase de conceptualización. Cabe notar que ésta evaluación se facilitó debido a que el experto entiende de formalismos computacionales.

La evaluación se realizó teniendo en cuenta dos criterios, según Gómez:

Marcos: se ha revisado si coincide con el modelo conceptual y se analizó que no exista redundancias, incompletitud e inconsistencias.

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

8.5 Evaluación de la Implementación de conocimientos.

Para la evaluación del Sistema Experto se compararon los resultados obtenidos por el sistema y por el experto, para lo cual se desarrolló un caso de prueba en donde se puede encontrar resultados particulares de requerimientos funcionales y no funcionales. En los funcionales se obtiene resultados de Marcos de problemas y los tipos de requerimientos, en los no funcionales tenemos características del sistema como son: capacidad, seguridad, rendimiento, etc.

El sistema experto se encuentra embebido en su sistema de gestión que permite administrar los distintos proyectos de prueba de casos reales.

Para estos casos de prueba (Proyectos) primero se realiza una evaluación del sistema de Gestión analizando que los valores no sean vacíos y que

tengan coherencia con el sistema a desarrollar (Tabla 8.1). Esta evaluación la realiza el Ingeniero de conocimientos de tal manera que sea legible para las partes involucradas.

Para el sistema experto se evaluará las preguntas realizadas (Tabla 8.2: Requerimientos Funcionales y la Tabla 8.5: Requerimientos no Funcionales), las reglas de inferencias (Tabla 8.3: Requerimientos Funcionales y la Tabla 8.6: Requerimientos no Funcionales), los metaconocimientos y los resultados obtenidos (Tabla 8.4: Requerimientos Funcionales y la Tabla 8.7: Requerimientos no Funcionales).

8.6 Caso ejemplo 1.1: Facturación Electrónica

Proyecto: Facturación Electrónica

Objetivo del proyecto: Este proyecto pretende producir un software que permita gestionar la facturación directamente con el SRI de manera que la administración de procesos tributarios se optimice.

Caso 1: Evaluación del Sistema de Gestión de Proyecto

Tabla 8. 1 Caso prueba Gestión de proyectos

CASO 1: ETAPA DE GESTIÓN DE PROYECTOS			
INFORMACIÓN DEL PROYECTO			
ID	Entrada/Pregunta	Descripción	Opciones
GP1	Nombre proyecto	Nombre largo del Proyecto	Sistema de Facturación Electrónica
GP2	Área	Área involucrada	Contable
GP3	Nombre del Sistema	Abreviatura del nombre del Proyecto	SFE
GP4	Empresa Solicitante	Nombre de la empresa que solicita el sistema	Empresa "Supermercado Patricia"
GP5	Empresa proveedora	Nombre de la empresa de desarrollo de proyectos	REDLeaf
GP5	¿La empresa posee red?	Permite determinar si tiene una infraestructura tecnológica de red	Si
GP6	¿El sistema interactúa con otros sistemas?	Permite determinar si el sistema a desarrollarse interactúa con otros sistemas externo	Si
GP7	Selección de la arquitectura	Si existe infraestructura de red permite seleccionar el tipo de arquitectura	Tres capas

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

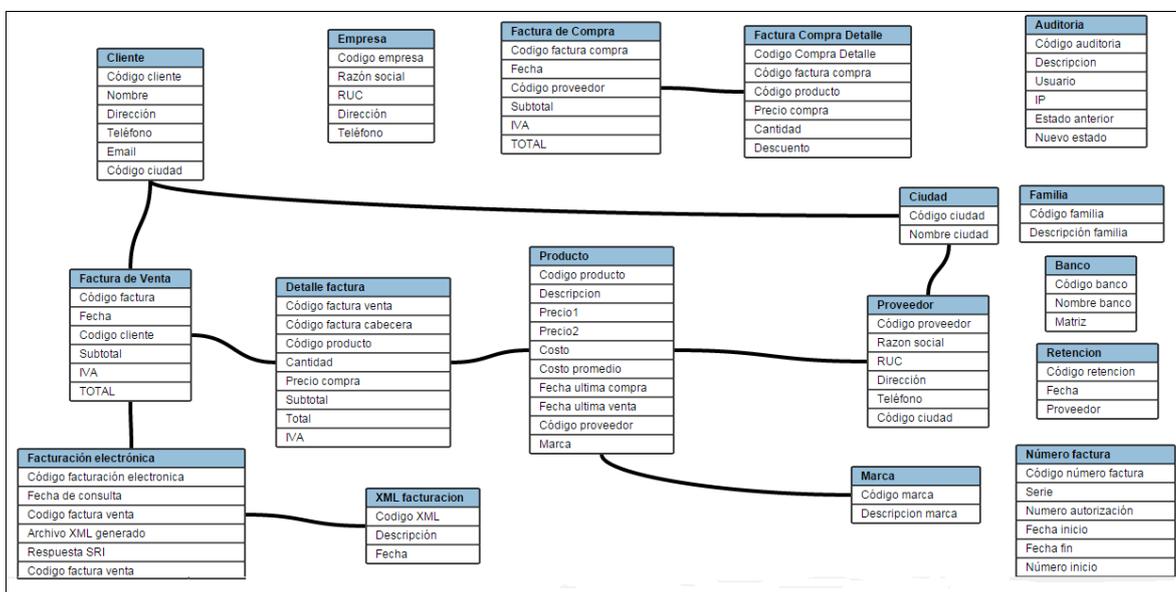
La Figura 8.1 ilustra el ingreso de datos, Información general del Proyecto de acuerdo al caso de prueba descrito en la tabla 8.1.

Gestión de Requisitos de Software					
Información general del proyecto		Información adicional		Entidades	
Definiciones, acrónimos y abreviaturas.					
+ Nueva definición ✎ Editar definición - Eliminar		+ Nuevo acrónimo ✎ Editar acrónimo - Eliminar		+ Nueva abreviatura ✎ Editar abreviatura - Eliminar	
Palabra	Definición	Acrónimo	Descripción	Abreviatura	Descripción
Facturación electrónica	Proceso que gestiona la facturación en línea y en tiempo real con las dependencias del SRI encargadas.	ERS	Especificación de requisitos de Software	SFE	Sistema de Facturación Electrónica
Servicio WEB	Componente informático que permite la ejecución de funciones o procedimientos ofrecidos por una institución para el uso de clientes quienes consumen el mismo de forma remota.	ARS	Análisis de requisitos del Sistema		
Internet	Red global que permite la compartición de recursos informáticos.	SRI	Servicio de Rentas Internas		
Lenguaje de Programación	Instrumento utilizado por ingenieros informáticos para crear un conjunto de instrucciones entendibles y ejecutables por un computador.	WSDL	Web Services Description Language		
		XML	eXtensible Markup Language		
		HTML	HyperText Markup Language		

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Figura 8. 2. Definiciones-Acrónimos-Abreviaturas

En la figura 8.3 se demuestra el modelo conceptual abalizado por el Ingeniero de Conocimientos y el experto.



Elaborado por: Corral, María Alexandra, Carrillo José Luis

Figura 8. 3. Modelo Conceptual del Sistema Facturación Electrónica

Caso 1: Evaluación del sistema experto para el análisis de requerimientos.

Caso 1.1: Especificación de requerimientos funcionales

Tabla 8. 2 Valores de caso de prueba especificación de requerimientos funcionales

CASO 1: Valores de caso de prueba			
Entradas para el Sistema experto.			
Especificación 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	Permite determinar si el sistema requiere de una herramienta para crear y	No

	crear y editar artefactos intangibles tales como textos o gráficos?	editar artefactos intangibles tales como texto, gráficos o estructuras similares para que puedan ser posteriormente copiadas, impresas, analizadas o usadas de otra manera	
--	---	--	--

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Siglas: ER (Especificación de Requerimientos)

La Tabla 8.3 describe las reglas de inferencia que permiten deducir cuatro tipos de Marcos de problemas y cuatro tipos de requerimientos funcionales basados en Michael Jackson los cuales son:

Requerimientos Funcionales

Requerimientos de consulta

Requerimientos de mapeo

Requerimientos de Reglas de Comportamiento

Requerimientos de Operaciones sobre dominios creados

Marcos de Problemas

Marcos de Información

Marcos de Transformación

Marcos de Control

Marcos de Workpieces.

Tabla 8. 3 Reglas de inferencia para requerimientos funcionales

```
;Delimitación y restricciones: Requerimientos, Marcos de Problemas
```

```
;Regla #21
```

```
(defrule Req-func-Consultas
```

```
  (declare (salience -1))
```

```
  (Es SolicitantesInformacion ?ent1)
```

```
  (Es InformacionEntregada ?ent2)
```

```
  ?ret1 <- (Es SolicitantesInformacion ?ent1)
```

```
  ?ret2 <- (Es InformacionEntregada ?ent2)
```

```
=>
```

```
  (open "outReqnoFunc.txt" outdata "a")
```

```
  (if (eq ?ent1 Si) then
```

```
    (if (eq ?ent2 Si) then
```

```
      (assert (Es UnRequerimiento ConsultaGlobal))
```

```
      (retract ?ret1)
```

```
      (retract ?ret2)))
```

```
  (if (eq ?ent1 Si) then
```

```
    (if (eq ?ent2 No) then
```

```
      (assert (Es UnRequerimiento ConsultaSolInf))
```

```
      (retract ?ret1)
```

```
      (retract ?ret2)))
```

```
  (if (eq ?ent1 No) then
```

```
    (if (eq ?ent2 Si) then
```

```
      (assert (Es UnRequerimiento ConsultaInfEnt))
```

```
      (retract ?ret1)
```

```
      (retract ?ret2)))
```

```
  (close outdata)
```

```
)
```

```
;Regla #22
```

```
(defrule Req-func-Regla-Comportamiento
```

```
  (declare (salience -1))
```

```
  (Es DispositivosExternos ?ent1)
```

```
  ?ret1 <- (Es DispositivosExternos ?ent1)
```

```

=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 Si) then
    (assert (Es UnRequerimiento Reglas-Comportamiento))
    (retract ?ret1))
  (close outdata)
)

;Regla #23
(defrule Req-func-Mapeo
  (declare (salience -1))
  (Es OperacionesTransformacion ?ent1)
  (Es OperacionesdeCalculo ?ent2)
  ?ret1 <- (Es OperacionesTransformacion ?ent1)
  ?ret2 <- (Es OperacionesdeCalculo ?ent2)
=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 Si) then
    (if (eq ?ent2 Si) then
      (assert (Es UnRequerimiento MapeoGlobal))
      (retract ?ret1)
      (retract ?ret2)))

    (if (eq ?ent1 Si) then
      (if (eq ?ent2 No) then
        (assert (Es UnRequerimiento MapeoOpCal))
        (retract ?ret1)
        (retract ?ret2)))

    (if (eq ?ent1 No) then
      (if (eq ?ent2 Si) then
        (assert (Es UnRequerimiento MapeoOpTra))
        (retract ?ret1)
        (retract ?ret2)))
  (close outdata)
)

```

```
;Regla #24
```

```
(defrule Req-func-Operaciones-Dominio
  (declare (salience -1))
  (Es ArtefactosSoftware ?ent1)
  ?ret1 <- (Es ArtefactosSoftware ?ent1)
=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 Si) then
    (assert (Es UnRequerimiento OperacionesDominios))
    (retract ?ret1))
  (close outdata)
)
```

```
;Regla #25
```

```
(defrule Req-func-MarcoProblema
  (declare (salience -10))
  (Es UnRequerimiento ?ent1)
  ?ret1 <- (Es UnRequerimiento ?ent1)
=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 ConsultaGlobal) then
    (assert (Es UnMarcoProblema Informacion))
    (printout outdata "Requerimiento de Consulta" crlf)
    (printout outdata "Marco de Problema de Información " crlf)
    (printout outdata "Reporte detallado de consultas, en varios formatos de salida
(pdf, doc, xls, etc.), los 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 ConsultaSolInf) then
    (assert (Es UnMarcoProblema Informacion))
    (printout outdata "Requerimiento de Consulta" crlf))
```

```
(printout outdata "Marco de Problema de Información " crlf)
```

```
(printout outdata "Reporte detallado de consultas, 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 ConsultaInfEnt) then
```

```
  (assert (Es UnMarcoProblema Informacion))
```

```
  (printout outdata "Requerimiento de Consulta" crlf)
```

```
  (printout outdata "Marco de Problema de Información " crlf)
```

```
  (printout outdata "Reporte detallado de consultas, en varios formatos de salida
(pdf, doc, xls, etc.), los filtros de búsqueda de información deben tener varios
criterios como: rangos de fechas, claves principales o palabras claves." crlf))
```

```
(if (eq ?ent1 Reglas-Comportamiento) then
```

```
  (assert (EsUn MarcoProblema Control))
```

```
  (printout outdata "Requerimiento de Regla de Comportamiento" crlf)
```

```
  (printout outdata "Marco de Problema de Control" crlf)
```

```
  (printout outdata "Utilizar los drives de Instalación que viene por defecto dentro
del equipo que vamos a controlar en nuestro software, procurar que la información
que ingrese desde el dispositivo externo llegue como 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." crlf))
```

```
(if (eq ?ent1 MapeoGlobal) then
```

```
  (assert (Es UnMarcoProblema Transformacion))
```

```
  (printout outdata "Requerimiento de Mapeo" crlf)
```

```
  (printout outdata "Marco de Problema de Transformación" crlf)
```

```
  (printout outdata "Revisar los datos de entrada con los cuales se realizan
cálculos generando nuevos tipos de datos en las salidas y/o generar nuevos
formatos de archivos de acuerdo con las necesidades de los clientes" crlf))
```

```
(if (eq ?ent1 MapeoOpCal) then
```

```
  (assert (Es UnMarcoProblema Transformacion))
```

```

(printout outdata "Requerimiento de Mapeo" crlf)
(printout outdata "Marco de Problema de Transformación" crlf)
(printout outdata "Revisar los datos de entrada con los cuales se realizan
cálculos generando nuevos tipos de datos en las salidas" crlf))

(if (eq ?ent1 MapeoOpTra) then
  (assert (Es UnMarcoProblema Transformacion))
  (printout outdata "Requerimiento de Mapeo" crlf)
  (printout outdata "Marco de Problema de Transformación" crlf)
  (printout outdata "Generar nuevos formatos de archivos de acuerdo con las
necesidades de los clientes" crlf))

(if (eq ?ent1 OperacionesDominios) then
  (assert (EsUn MarcoProblema WorkPiece))
  (printout outdata "Requerimiento de Operaciones Dominios" crlf)
  (printout outdata "Marco de Problema de WorkPiece" crlf)
  (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 se debe realizar el
procesamiento de imágenes dentro de la capa de usuario sino en la capa de
negocios para que no afecte al performance de la aplicación. Los documentos
generados pueden ser impresos en varios formatos para facilidad del cliente final."
crlf))
  (retract ?ret1)
  (close outdata)
)

;Regla #26
(defrule Borr-Sol-MarcoProblema
  (declare (salience -20))
  (Es UnMarcoProblema ?ent1)
  ?ret1 <- (Es UnMarcoProblema ?ent1)

=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 Informacion) then

```

```

(retract ?ret1))

(if (eq ?ent1 Control) then
  (retract ?ret1))

(if (eq ?ent1 Transformacion) then
  (retract ?ret1))

(if (eq ?ent1 WorkPiece) then
  (retract ?ret1))
(close outdata)
)

```

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

La Tabla 8.4 muestra los resultados de las inferencias producidas al instanciar los valores ingresados por los usuarios y las reglas de conocimiento generadas para los requerimientos funcionales.

Tabla 8. 4 Resultados obtenidos de las inferencias para el sistema SFE

RESULTADOS DEL CASO DE PRUEBA			
ESPECIFICACIÓN DE REQUERIMIENTOS FUNCIONALES			
ID	Pregunta/Metaregla	Inferencia	Metaconocimiento/Resultado
ERF1	ER1 y ER2	Regla 21	Marco de Problema de Información.
ERF2	Marco de Problema de Información.	Regla 25	Requerimiento de Consulta 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 mejorar el performance y la utilidad de las consultas.
ERF3	ER3 y ER4	Regla 23	Marco de Problema de Transformación.
ERF4	Marco de Problema de Transformación	Regla 26	Requerimiento de Mapeo 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.

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Siglas: ERF (Educación de Requerimientos Funcionales)

De las inferencias realizadas por el sistema experto podemos indicar que existe en el proyecto SFE (Sistema de Facturación Electrónica) un marco de problema de Información y un marco de problema de transformación. Lo que permite determinar que existen requerimientos de consulta y de mapeo, especificados en las filas ERF2 y ERF4, columna resultados.

Por lo tanto se puede indicar que el sistema experto infiere y deduce satisfactoriamente los requerimientos funcionales de acuerdo a los resultados obtenidos y abalizados por el Ingeniero en Conocimiento y el experto:

Caso 1.2: Especificación de requerimientos no funcionales

Tabla 8. 5 Valores de caso de prueba especificación de requerimientos

No funcionales

CASO 1: Valores de caso de prueba			
Entradas para el Sistema experto.			
Especificación 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	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	Permite determinar el nivel de seguridad del sistema	3

	información que manejará el sistema		
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

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Siglas: ER (Especificación de Requerimientos)

La Tabla 8.6 describe las reglas de inferencia que permiten deducir requerimientos no funcionales, incluidas las restricciones de diseño como son:

Los requisitos de lenguaje
 Las limitaciones del Hardware
 Las Interfaces a otras aplicaciones
 Los protocolos de comunicación
 Disponibilidad
 Seguridad
 Capacidad

Tabla 8. 6 Reglas de inferencia para requerimientos no funcionales

```

;Regla #1
(defrule Req-no-func-Lenguaje
  (Es Herramienta ?ent1)
  (Es Sistema ?ent2)
  =>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 SoftwareLibre) then
    (if (eq ?ent2 Interno) then
      (printout outdata "Java con Herramientas de Escritorio" crlf)))
  (close outdata)

```

```

)

;Regla #2
(defrule Req-no-func-Lenguaje-BDD-Internet
  (Es Herramienta ?ent1)
  (Es Computadoras ?ent2)
  (Es BaseDatos ?ent3)
  (Es Internet ?ent4)

=>
  (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"
            crf))))))
    (close outdata)
  )

;Regla #3
(defrule Req-no-func-Lenguaje-y-BDD
  (Es Herramienta ?ent1)
  (Es Computadoras ?ent2)
  (Es BaseDatos ?ent3)
  (Es Internet ?ent4)

=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 SoftwareLibre) then
    (if (eq ?ent2 Escritorio) then
      (if (eq ?ent3 Si) then
        (if (eq ?ent4 No) then
          (printout outdata "Java con Herramientas de Escritorio y MySql" crf))))))
    (close outdata)
  )
)

```

```

;Regla #4
(defrule Req-no-func-Internet-con_Dispositivos-Moviles
  (Es Herramienta ?ent1)
  (Es BaseDatos ?ent2)
  (Es Internet ?ent3)
  (Es Dispositivos ?ent4)
=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 SoftwareLibre) then
    (if (eq ?ent2 Si) then
      (if (eq ?ent3 Si) then
        (if (eq ?ent4 Moviles) then
          (printout outdata "Java J2ME con Herramientas Web y MySql" crlf))))))
  (close outdata)
)

```

```

;Regla #5
;(defrule Req-no-func-Lenguaje-Herramientas
;  (Es Herramienta ?ent1)
;  (Es Computadoras ?ent2)
;  (Es BaseDatos ?ent3)
;  (Es Internet ?ent4)

; =>
;  (open "outReqnoFunc.txt" outdata "a")
;  (if (eq ?ent1 SoftwareLibre) then
;    (if (eq ?ent2 Escritorio) then
;      (if (eq ?ent3 No) then
;        (if (eq ?ent4 Si) then
;          (printout outdata "Java con Herramientas de Escritorio" crlf))))))
;  (close outdata)

;)

```

```

;Regla #6
(defrule Req-no-func-Software-Libre-Herramientas-Internet

```

```

(Es Herramienta ?ent1)
;(Es Computadoras ?ent2)
(Es BaseDatos ?ent3)
(Es Internet ?ent4)
=>
(open "outReqnoFunc.txt" outdata "a")
(if (eq ?ent1 SoftwareLibre) then
  ;(if (eq ?ent2 Escritorio) then
    (if (eq ?ent3 No) then
      (if (eq ?ent4 Si) then
        (printout outdata "PHP con Herramientas Web" crlf))))
    (close outdata)
  )
)

;Regla #7
(defrule Req-no-func-Software-Propietario-Escritorio
  (Es Herramienta ?ent1)
  (Es Computadoras ?ent2)
  (Es BaseDatos ?ent3)
=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 SoftwarePropietario) then
    (if (eq ?ent2 Escritorio) then
      (if (eq ?ent3 Si) then
        (printout outdata "C# con Herramientas de Escritorio y MySql" crlf))))
    (close outdata)
  )
)

;Regla #8
(defrule Req-no-func-Software-Propietario-BDD-Int
  (Es Herramienta ?ent1)
  (Es BaseDatos ?ent2)
  (Es Internet ?ent3)
=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 SoftwarePropietario) then

```

```

    (if (eq ?ent2 Si) then
      (if (eq ?ent3 Si) then
        (printout outdata "C# con Herramientas Web y MySql" crlf))))
    (close outdata)
  )

;Regla #9
(defrule Req-no-func-Software-Propietario-BDD-Internet
  (Es Herramienta ?ent1)
  (Es BaseDatos ?ent2)
  (Es Internet ?ent3)
  =>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 SoftwarePropietario) then
    (if (eq ?ent2 Si) then
      (if (eq ?ent3 Si) then
        (printout outdata "C# con Herramientas Web y MySql" crlf))))
    (close outdata)
  )

;Regla #10
(defrule Req-no-func-Software-Propietario-Escritorio-Internet
  (Es Herramienta ?ent1)
  (Es Computadoras ?ent2)
  (Es BaseDatos ?ent3)
  (Es Internet ?ent4)
  =>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 SoftwarePropietario) then
    (if (eq ?ent2 Escritorio) then
      (if (eq ?ent3 No) then
        (if (eq ?ent4 Si) then
          (printout outdata "C# con Herramientas Web" crlf))))
    (close outdata)
  )

```

```
;Regla #11
```

```
(defrule Req-no-func-Software-Propietario-Computadores-Escritorio
```

```
  (Es Herramienta ?ent1)
```

```
  (Es Computadoras ?ent2)
```

```
  (Es BaseDatos ?ent3)
```

```
  (Es Internet ?ent4)
```

```
=>
```

```
  (open "outReqnoFunc.txt" outdata "a")
```

```
  (if (eq ?ent1 SoftwarePropietario) then
```

```
    (if (eq ?ent2 Escritorio) then
```

```
      (if (eq ?ent3 No) then
```

```
        (if (eq ?ent4 No) then
```

```
          (printout outdata "C# con Herramientas de Escritorio" crlf))))))
```

```
  (close outdata)
```

```
)
```

```
;Regla #12
```

```
(defrule Req-no-func-Multiples-Lenguajes
```

```
  (Es MultiplesLenguajes ?ent1)
```

```
=>
```

```
  (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)
```

```
=>
```

```
  (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 #14
(defrule Req-no-func-Conexion-Externa-Envio-Tramas
  (Es ConexionSistemaExterno ?ent1)
  (Es EnvioTramas ?ent2)
=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 Si) then
    (if (eq ?ent2 Si) then
      (printout outdata "Archivos Plano con: dirección del FTP, user, pwd para
acceder al Servidor FTP, Hora del archivo que esta listo para consumir, Formato del
archivo, archivo WDSL" crlf)))
    (close outdata)
  )

;Regla #15
(defrule Req-no-func-Conexion-Externa-BDD
  (Es ConexionSistemaExterno ?ent1)
  (Es ClonandoBaseDatos ?ent2)
  (Es ConsultasVistasBaseDatos ?ent3)
=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 Si) then
    (if (eq ?ent2 Si) then
      (if (eq ?ent3 No) then
        (printout outdata "Archivo de la estructura de la BDD, user y pwd del
servidor, puerto de comunicación" crlf))))
    (close outdata)
  )

;Regla #16
(defrule Req-no-func-Conexion-Externa-BDD-Vistas-Consultas

```

```

(Es ConexionSistemaExterno ?ent1)
(Es ClonandoBaseDatos ?ent2)
(Es ConsultasVistasBaseDatos ?ent3)
=>
(open "outReqnoFunc.txt" outdata "a")
(if (eq ?ent1 Si) then
  (if (eq ?ent2 Si) then
    (if (eq ?ent3 Si) then
      (printout outdata "Estructuras de consultas a ejecutar, user y pwd del BDD"
crf))))
  (close outdata)
)

;Regla #17
(defrule Req-no-func-Sistemas-Externos-COM+
  (Es ConexionSistemaExterno ?ent1)
  (Es SistemasExternosCOM+ ?ent2)
=>
  (open "outReqnoFunc.txt" outdata "a")
  (if (eq ?ent1 Si) then
    (if (eq ?ent2 Si) then
      (printout outdata "Proporcionar COM+ Cliente al Servidor, parámetros de
Entrada y Salida" crf)))
  (close outdata)
)

;Regla #18
(defrule Req-no-func-Seguridad
  (Es Sensibilidad ?ent1)
=>
  (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" crf))
  (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 hara encriptando el canal de envio" crlf))
  (if (eq ?ent1 2) then
    (printout outdata "Sensibilidad Media: Manejar autenticación de usuarios 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 caida 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)
)

```

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

La Tabla 8.7. Muestra los resultados de las inferencias producidas al instanciar los valores ingresados por los usuarios y las reglas de conocimiento generadas para los requerimientos no funcionales.

Tabla 8. 7 Resultados obtenidos de las inferencias del sistema experto

RESULTADOS DEL CASO DE PRUEBA			
ESPECIFICACIÓN DE REQUERIMIENTOS NO FUNCIONALES			
RESTRICCIONES DE DISEÑO			
ID	Pregunta/Metaregla	Inferencia	Metaconocimientos/Resultado
ERNF1	ER7 y ER9 Y ER10	Regla 2	PHP con Herramientas Web y MySql
ERNF2	ER8	Regla 12	Archivos de Globalización
ERNF3	ER11 Y ER11.1	Regla 13	Enviar requisitos como dirección del servicio Web, Archivos WDSL
ERNF4	ER12	Regla 19	Disponibilidad Media: Utilizar un servidor robusto para el alojamiento de la aplicación y programar puntos de restauración y respaldo
ERNF5	ER13	Regla 18	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 encriptado el canal de envío
ERNF6	ER14	Regla 20	Capacidad Media: Su Software necesita un servidor robusto de 500GB de almacenamiento y 2 procesadores sin la necesidad de virtualización y sistemas de respaldo

Elaborado por: Corral, María Alexandra, Carrillo José Luis.

Siglas: ERNF (Educación de Requerimientos no Funcionales)

De acuerdo a la tabla de evaluación 8.7 podemos indicar que el proyecto SFE (Sistema de Facturación Electrónica) tienen las siguientes características:

- 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 encriptado 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

Por lo tanto se puede 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:

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.

El anexo 1 se muestra la salida del Documento de Especificación de Requisitos de Software (ERS) para el Sistema de Facturación Electrónica generado por el Sistema Experto

En el anexo 2 se adjunta una certificación de verificación y validación del Sistema por la empresa REDleaf desarrolladora del Sistema de Facturación Electrónica para el Supermercado "Patricia". .

CAPÍTULO VIII

CONCLUSIONES

- 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 tesis 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, tipificación de requerimientos.
- En ésta tesis 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-externo propuesta por Michael Jackson, en la que los requerimientos son identificados por los expertos para deducir a un tipo de marco de problema, permitiendo al sistema dar recomendaciones sobre requerimientos funcionales indicando el porqué de las mismas, lo que asegura la confiabilidad de los requisitos.
- En lo que respecta a los requerimientos no funcionales fueron inferidos utilizando características propias de todos los sistemas software.

- De las evaluaciones y de los resultados obtenidos se puede concluir que éstos fueron satisfactorios para el Ingeniero de Conocimiento y el Experto, siendo un sistema funcional.
- La herramienta desarrollada genera el documento de especificación de requisitos de software basada en el estándar IEEE 830, artefacto que resulta de gran utilidad y apoyo para el análisis y diseño de proyectos de software, el cual ha sido utilizado como referente de negociación para el desarrollo de un producto de software.

CAPÍTULO IX

BIBLIOGRAFIA

Agarwal, M. and S. Goel (2014). Expert system and it's requirement engineering process. Recent Advances and Innovations in Engineering (ICRAIE), 2014, IEEE.

Committee, I. C. S. S. E. S. and I.-S. S. Board (1998). IEEE Recommended Practice for Software Requirements Specifications, Institute of Electrical and Electronics Engineers.

Committee, I. S. C. (1990). "IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). Los Alamitos." CA: IEEE Computer Society.

Gómez-Pérez, A., M. Fernández, et al. (1996). "Towards a method to conceptualize domain ontologies."

IEEE Computer Society. Software Engineering Standards Committee, & IEEE-SA Standards Board. (1998). IEEE Recommended Practice for Software Requirements Specifications. Institute of Electrical and Electronics Engineers.

IEEE Standards Coordinating Committee. (1990). IEEE Standard Glossary of Software Engineering Terminology (IEEE Std 610.12-1990). Los Alamitos. CA: *IEEE Computer Society*.

Jackson, M. (1995). Software requirements & specifications: A Lexicon of practice, principles, and prejudices. Addison-Wesley. 1995.

Jackson, M. (1999). "Problem analysis using small problem frames." South African Computer Journal: 47-60.

Jackson, M. (2010). "Representing structure in a software system design." Design Studies 31(6): 545-566.

Kim, J., S. Park, et al. (2008). "DRAMA: A framework for domain requirements analysis and modeling architectures in software product lines." Journal of Systems and Software 81(1): 37-55.

Molinero, L. D. H. "Tutorial de Clips."

Nazareth, D. L. (1989). "Issues in the verification of knowledge in rule-based systems." *International Journal of Man-Machine Studies* 30(3): 255-271.

O'Keefe, R. M. and D. E. O'Leary (1993). "Expert system verification and validation: a survey and tutorial." *Artificial Intelligence Review* 7(1): 3-42.

O'Keefe, R. M. and D. E. O'Leary (1993). "Expert system verification and validation: a survey and tutorial." *Artificial Intelligence Review* 7(1): 3-42.

Peña-Ayala, A. (2006). *Sistemas basados en Conocimiento: Una Base para su Concepción y Desarrollo* [PhD Thesis]. México: *Instituto Politécnico Nacional*.

Preece, A. D. (1990). "Towards a methodology for evaluating expert systems." *Expert Systems* 7(4): 215-223.

Pressman, R. S. (1997). *Ingeniería del Software: Un enfoque práctico*. Mikel Angoar. //libro
Rosique Contreras, M. F., M. Jiménez Buendía, et al. (2010). "Evaluación de herramientas de gestión de requisitos."

Preece, A. D. (1990). "Towards a methodology for evaluating expert systems." *Expert Systems* 7(4): 215-223.

Prieto-Diaz, R. Arango, G., *Domain Analysis: Acquisition of Reusable Information for Software Construction*, Los Alamitos, IEEE Computer Society Press, 1989.

Sampaio do Prado Leite, J. C., & Freeman, P. A. (1991). Requirements validation through viewpoint resolution. *Software Engineering, IEEE Transactions on*, 17(12), 1253-1269.

CAPÍTULO X

ANEXOS

ANEXO 1

Documento de Especificación de Requisitos de Software (ERS) para el Sistema de Facturación Electrónica generado por el Sistema Experto

ANEXO 2

Certificación de verificación y validación del Sistema por la empresa REDleaf desarrolladora del Sistema de Facturación Electrónica para el Supermercado "Patricia".

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
MAESTRÍA EN INGENIERÍA DE SOFTWARE

CERTIFICADO

Se certifica que el presente trabajo fue desarrollado por los Srs. Ing. María Alexandra Corral Díaz y por el Ing. José Luis Carrillo Medina.

En la ciudad de Latacunga, a los 10 días del mes de marzo del 2015.

MSc. Eddie Galarza Zambrano
Director de Tesis

MSc. Lucas Garcés G.
Coordinador de la Maestría

Dr. Rodrigo Vaca
Secretario Académico