

# Universidad de las Ciencias Informáticas

## Facultad 3



## **Título: Procedimiento para la validación de requisitos de software.**

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas.

**Autor(es):** Geidys Padrón Blanco.

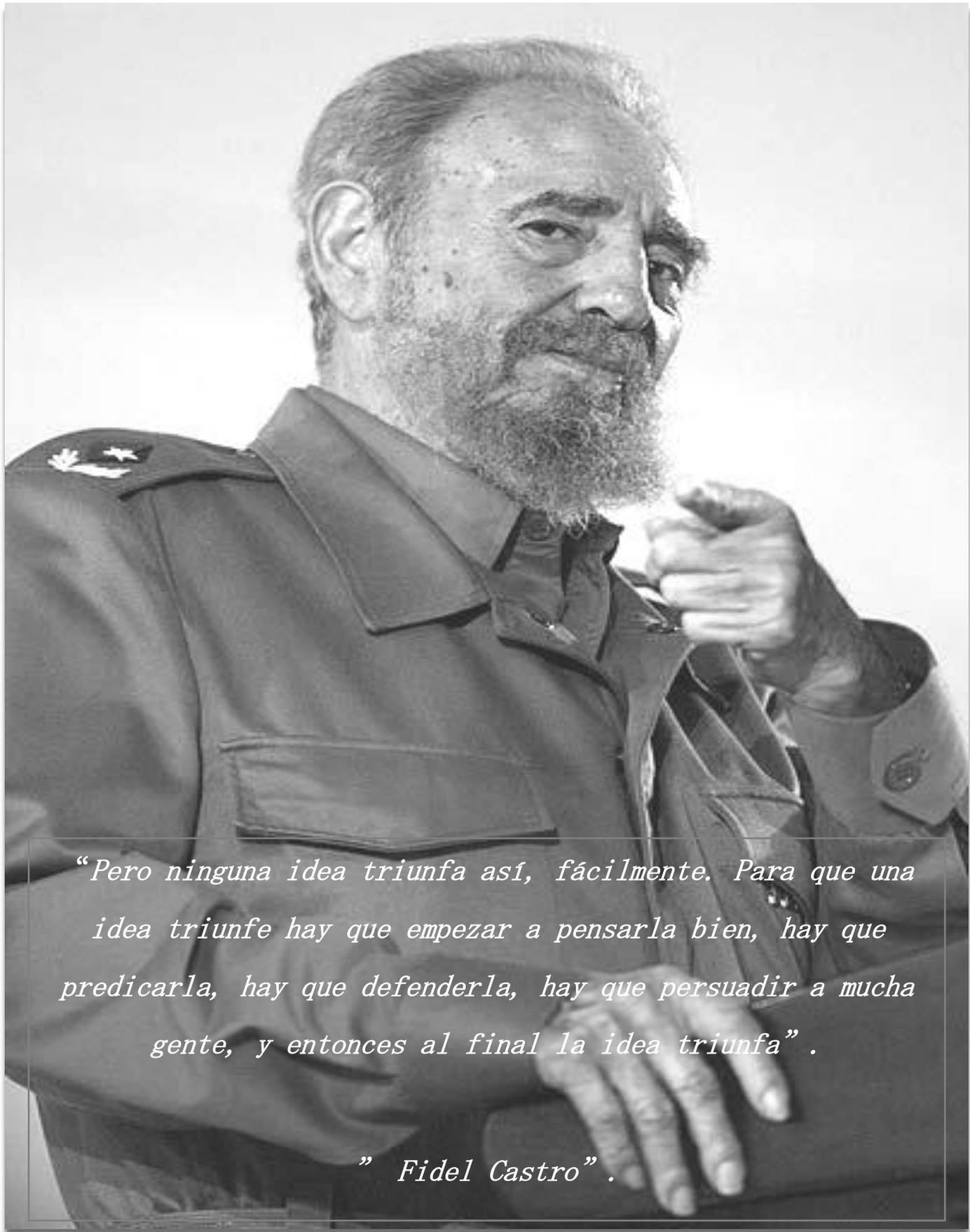
Ailec Rodríguez Iglesias.

**Tutor(es):** Ing. Diana Valdés González.

**Co-tutor:** Ing. Violena Hernández Aguilar.

Habana, Cuba

Junio, 2009



*“Pero ninguna idea triunfa así, fácilmente. Para que una idea triunfe hay que empezar a pensarla bien, hay que predicarla, hay que defenderla, hay que persuadir a mucha gente, y entonces al final la idea triunfa” .*

*” Fidel Castro” .*

## Datos de Contacto

Tutora: Ing. Diana Valdés González

- Ingeniero en Ciencias Informáticas
- Adiestrado
- 2 Tesis
- Analista del Sistema de Administración Financiera del Proyecto Registros y Notarias.

## Agradecimientos

*Hay momentos que convierten los días más comunes en verdaderos acontecimientos, y existen personas que convierten esos acontecimientos en escalones cada vez más altos en la escalera de la vida. Por todo esto hoy quiero agradecerles a todas aquellas personas que me hayan ayudado a convertirme en una gran personas.*

*A Mami por darme la vida y todo lo mejor de este mundo. A mi hermana por apoyarme siempre. A toda mi familia en general por creer en mí.*

*Al amor de mi vida (Addsel), por enseñarme tantas cosas de la vida, por darme tanto amor, apoyo, por tanta dedicación, entrega, paciencia, por ser mi ángel, mi guía, mi amigo, y sobre todo por amarme tanto.*

*A mi tío (Alberto) por ser como un padre para mí, por su amor, comprensión y por ayudarme en todo momento.*

*A mi compañera de tesis (Ailec), que más que amiga es como mi hermana, quiero agradecerle por aceptarme como soy, por tener tanta paciencia conmigo, por ayudarme, apoyarme y quererme tanto, y principalmente por confiar siempre en mí.*

*A mis amigas las jimaguas (Dianefys y Diarefys) que aunque las conocí hace poco, son como mis hermanas pequeñas y me han dado todo su apoyo y comprensión cuando más lo necesite.*

*A lisandra (La piedra) por ayudarme tanto durante toda la carrera. A salvador (el bicho) por estar siempre ahí.*

*A yelaine por su apoyo y comprensión.*

*A las personas del apto, por ayudarme en todo, especialmente a mi sabandija (Maylen) y a Jenny, por dedicarle tantas horas de su tiempo libre tanto a mi tesis como a mi Prueba de Nivel.*

*A los primos (Odaymis y Felix) por contagiarme siempre con su alegría.*

*A mi tutora por tanta paciencia, apoyo y comprensión.*

*Gracias a todos. Geidys*

## Agradecimientos

*A mi mamita por ayudarme a ser quien soy, por ser mi amiga, por apoyarme en los momentos difíciles que me ha tocado vivir, por darme una vida llena de amor y comprensión, por ser tan luchadora y por ser la mejor de las madres.*

*A mi hermanito Idalnel, gracias por tu cariño, por verme como ejemplo y por mostrarme cuán importante soy para ti.*

*A mis abuelos, por quererme tanto, por malcriarme, por preocuparse tanto por mis estudios, por ser los abuelos perfectos.*

*A toda mi familia que a pesar de la distancia se ha mantenido al tanto de mis estudios, los tengo presentes en todo momento.*

*A mi compañera de tesis, Geidys, que ha sido mucho más que eso, es mi hermana, mi confidente, mi cómplice, mi mejor amiga. Gracias por aceptarme como soy, por tus esfuerzos para realizar la tesis, por quererme y ayudarme como nadie lo ha hecho en esta universidad.*

*A mis tíos Lázaro y Pupi, por preocuparse por mis estudios y mi familia, gracias por ayudarme a cumplir mis sueños.*

*A Rolando, mi BB, que me ha brindado su amistad estos 5 años, por ser tan especial, ocupas un lugar importante en mi corazón.*

*A Yaily y Marlen, mis ángeles, gracias por adornar mi vida con sus alegrías.*

*A mis amistades inolvidables: Any con sus incontenibles lágrimas, Yasmery con su felicidad contagiosa, Rosa con su locura, las quiero.*

*A Addsel, por ayudarnos tanto, por ser tan comprensivo, a Odaimy y Félix por ser tan divertidos y convertirse en tan poco tiempo en amigos especiales.*

*A mi novio Marlon, que me ha apoyado mucho, gracias por entenderme y por cuidarme, pero sobre todo por demostrarme en tan poco tiempo que la felicidad existe. Te quiero.*

*A mis compañeros del 3101 y del 3502, me enseñaron mucho a lo largo de la carrera.*

*A Osmín y Diliam, por preocuparse tanto por mi.*

*A mi tutora que siempre nos apoyó, gracias por su dedicación y entrega.*

*Gracias a todas las personas que de una forma u otra contribuyeron a mi formación durante todos estos años y a la realización de mi tesis.*

*Ailec*

## Dedicatoria

*Quiero dedicarle este logro tan importante para mí.*

*A mis abuelas que donde quiera que estén estarán orgullosas de mí.*

*A mi mamá por todo su amor, comprensión, dulzura, paciencia, apoyo y sacrificio.*

*A mis hermanos.*

*A mi Tata por servirme de guía y ejemplo, por guiarme siempre hacia los estudios y por apoyarme en todo.*

*A mis hermanos menores que siempre han confiado en mí y me han tomado como ejemplo.*

*A mis sobrinos que son la luz de mis ojos, que luchan siempre por lo que quieren, si es que de verdad lo quieren y que dios permita que se conviertan en hombres y mujeres de bien.*

*Geidys*

*Dedico esta tesis a mi padre que desde el cielo vela por mí. Por su sacrificio, su apoyo, su confianza, por dedicarse a mí en vida, por creer que yo si podía triunfar, por ser mi ejemplo y mi guía, ojalá hubieses podido ver mi sueño hecho realidad, estarás siempre en mi corazón.*

*Ailec*

## RESUMEN

En la actualidad, los diferentes proyectos productivos de la Universidad de las Ciencias Informáticas no cuentan con un procedimiento para la Validación de requisitos de software, siendo ésta una de las etapas esenciales de la Ingeniería de requisitos, lo que influye de manera directa en el desarrollo del software, afectando la calidad de los requisitos que le son entregados al cliente. Se hace necesario establecer un procedimiento que permita, mediante la aplicación de diferentes técnicas, comprobar que los requisitos satisfacen las necesidades de los clientes y que contribuya a la obtención de un producto de calidad.

En el presente trabajo se describe un procedimiento para la Validación de requisitos de software, teniendo como principal objetivo la detección y corrección de errores en los mismos. Para ello se realiza un estudio de la Ingeniería de requisitos, específicamente en la etapa de validación, con el fin de identificar los diferentes métodos y técnicas empleados y así elaborar una propuesta que las combine y organice en una secuencia de actividades y pasos claramente definidos que le aporten al proceso originalidad, claridad y calidad.

El diseño de dicho procedimiento consta de tres fases, cada una de las cuales contiene un conjunto de actividades, responsables, así como productos de entrada y salida.

El resultado fundamental se centra en la obtención de este procedimiento, que con su aplicación garantice mejoras en la calidad de los requisitos. Se pretende extender su utilización en todos los proyectos de la universidad para estandarizar el proceso de Validación de requisitos de manera eficiente.

Para validar la propuesta se aplicó el procedimiento en el módulo Generador de Reportes del proyecto Ministerio de Energía y el Petróleo de la República Bolivariana de Venezuela, perteneciente al centro de Tecnología y Almacenamiento de Datos de la UCI, obteniéndose resultados satisfactorios.

## PALABRAS CLAVE

Ingeniería de requisitos, Validación de requisitos de software, responsables, productos, técnicas de validación.

# ÍNDICE

|                                                                                          |           |
|------------------------------------------------------------------------------------------|-----------|
| <b>AGRADECIMIENTOS.....</b>                                                              | <b>4</b>  |
| <b>DEDICATORIA .....</b>                                                                 | <b>7</b>  |
| <b>ÍNDICE DE TABLAS. ....</b>                                                            | <b>11</b> |
| <b>INTRODUCCIÓN .....</b>                                                                | <b>13</b> |
| <b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....</b>                                           | <b>16</b> |
| INTRODUCCIÓN .....                                                                       | 16        |
| 1.1    REQUISITOS DE SOFTWARE. DEFINICIONES Y CONCEPTOS. ....                            | 16        |
| 1.1.1 <i>Requisitos funcionales y requisitos no funcionales.</i> .....                   | 17        |
| 1.2    INGENIERÍA DE REQUISITOS DE SOFTWARE.....                                         | 19        |
| 1.2.1 <i>Características de la Ingeniería de requisitos.</i> .....                       | 22        |
| 1.2.2 <i>Importancia de la Ingeniería de requisitos de software.</i> .....               | 22        |
| 1.2.3 <i>Etapas de la Ingeniería de requisitos de software.</i> .....                    | 23        |
| 1.3    VALIDACIÓN DE REQUISITOS DE SOFTWARE. PARTICULARIDADES. ....                      | 29        |
| 1.3.1 <i>Diferencia entre verificación y Validación de requisitos de software.</i> ..... | 29        |
| 1.4    MÉTODOS DE VALIDACIÓN DE REQUISITOS DE SOFTWARE. ....                             | 30        |
| 1.4.1 <i>Revisiones.</i> .....                                                           | 31        |
| 1.4.2 <i>Prototipos orientados a clientes y usuarios.</i> .....                          | 32        |
| 1.4.3 <i>Auditorias.</i> .....                                                           | 32        |
| 1.4.4 <i>Matrices de trazabilidad.</i> .....                                             | 32        |
| 1.5    CALIDAD DE SOFTWARE. DEFINICIONES. ....                                           | 33        |
| 1.6    NOTACIÓN DE MODELADO. ....                                                        | 34        |
| 1.6.1 <i>BPMN.</i> .....                                                                 | 35        |
| 1.6.2 <i>UML.</i> .....                                                                  | 37        |
| 1.7    HERRAMIENTA.....                                                                  | 38        |
| 1.7.1 <i>Visual Paradigm</i> .....                                                       | 38        |
| 1.7.2 <i>Rational Rose.</i> .....                                                        | 39        |
| CONCLUSIONES. ....                                                                       | 40        |
| <b>CAPÍTULO 2: PROPUESTA DE SOLUCIÓN. ....</b>                                           | <b>41</b> |
| INTRODUCCIÓN .....                                                                       | 41        |
| 2.1    ALCANCE DE LA PROPUESTA DE SOLUCIÓN.....                                          | 41        |

|                                                                              |                                                                                                              |            |
|------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------|------------|
| 2.1.1                                                                        | <i>Propósito de la propuesta de solución</i> .....                                                           | 42         |
| 2.2                                                                          | OBJETIVOS.....                                                                                               | 42         |
| 2.3                                                                          | CARACTERÍSTICAS .....                                                                                        | 43         |
| 2.4                                                                          | <i>Estructura</i> .....                                                                                      | 43         |
| 2.5                                                                          | <i>Responsables y productos de entradas y salidas</i> .....                                                  | 43         |
| 2.6                                                                          | <i>Descripción del procedimiento para la Validación de requisitos de software</i> .....                      | 50         |
| 2.7                                                                          | <i>Fases del procedimiento</i> .....                                                                         | 51         |
| 2.8                                                                          | <i>Fase I. Validación del listado de requisitos de software</i> .....                                        | 52         |
| 2.9                                                                          | <i>Fase II. Validación del documento de especificación de requisitos</i> .....                               | 53         |
| 2.10                                                                         | <i>Fase III. Validación del modelo de casos de uso del Sistema</i> .....                                     | 55         |
| 2.11                                                                         | <i>Generalidades del procedimiento para validar requisitos de software</i> .....                             | 57         |
| 2.12                                                                         | <i>Fase I. Validación del listado de requisitos de software. Particularidades</i> .....                      | 59         |
| 2.13                                                                         | <i>Fase II. Validación del documento de especificación de requisitos de software. Particularidades</i> ..... | 62         |
| 2.14                                                                         | <i>Validación del modelo de casos de uso del sistema. Particularidades</i> .....                             | 69         |
|                                                                              | CONCLUSIONES .....                                                                                           | 80         |
| <b>CAPÍTULO 3: APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADOS.....</b> |                                                                                                              | <b>81</b>  |
|                                                                              | INTRODUCCIÓN .....                                                                                           | 81         |
| 3.1                                                                          | PLANIFICACIÓN DEL PROCEDIMIENTO.....                                                                         | 81         |
| 3.2                                                                          | APLICACIÓN DE LA FASE I. RESULTADOS.....                                                                     | 82         |
| 3.3                                                                          | APLICACIÓN DE LA FASE II. RESULTADOS.....                                                                    | 84         |
| 3.4                                                                          | APLICACIÓN DE LA FASE III. RESULTADOS.....                                                                   | 88         |
|                                                                              | CONCLUSIONES .....                                                                                           | 103        |
|                                                                              | <b>CONCLUSIONES .....</b>                                                                                    | <b>104</b> |
|                                                                              | <b>RECOMENDACIONES.....</b>                                                                                  | <b>105</b> |
|                                                                              | <b>BIBLIOGRAFÍA.....</b>                                                                                     | <b>106</b> |
|                                                                              | <b>GLOSARIO DE TÉRMINOS.....</b>                                                                             | <b>108</b> |
|                                                                              | <b>ANEXOS .....</b>                                                                                          | <b>111</b> |

## Índice de Ilustraciones.

|                                                             |     |
|-------------------------------------------------------------|-----|
| ILUSTRACIÓN 1 ETAPAS DE LA INGENIERÍA DE REQUISITOS.....    | 24  |
| ILUSTRACIÓN 2 LISTADO DE REQUISITOS DE SOFTWARE.....        | 45  |
| ILUSTRACIÓN 3 ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE..... | 46  |
| ILUSTRACIÓN 4 MODELO DE CASOS DE USO DEL SISTEMA.....       | 47  |
| ILUSTRACIÓN 5 DESCRIPCIÓN GENERAL DEL PROCEDIMIENTO.....    | 51  |
| ILUSTRACIÓN 6 ACTIVIDADES DE LA FASE I.....                 | 53  |
| ILUSTRACIÓN 7 ACTIVIDADES DE LA FASE II.....                | 55  |
| ILUSTRACIÓN 8 ACTIVIDADES DE LA FASE III.....               | 57  |
| ILUSTRACIÓN 9 ATIVIDADES DEL PROCEDIMIENTO.....             | 58  |
| ILUSTRACIÓN 10 RESULTADOS DE LA FASE I.....                 | 84  |
| ILUSTRACIÓN 11 RESULTADOS DE LA FASE II.....                | 88  |
| ILUSTRACIÓN 12 MATRIZ DE TRAZABILIDAD.....                  | 89  |
| ILUSTRACIÓN 13 EXPORTAR INFORMACIÓN.....                    | 100 |
| ILUSTRACIÓN 14 CONFIRMACIÓN DE LA EXPORACIÓN.....           | 100 |
| ILUSTRACIÓN 15 BUSCAR MODELO.....                           | 101 |
| ILUSTRACIÓN 16 BUSCAR MODELO.....                           | 101 |
| ILUSTRACIÓN 17 BUSCAR REPORTES.....                         | 102 |

## Índice de tablas.

|                                                                                            |    |
|--------------------------------------------------------------------------------------------|----|
| TABLA 1 RESPONSABLES Y PRODUCTOS.....                                                      | 44 |
| TABLA 2 ESTRUCTURA DEL DOCUMENTO DE NO CONFORMIDADES.....                                  | 50 |
| TABLA 3 FASE I. VALIDACIÓN DEL LISTADO DE REQUISITOS DE SOFTWARE.....                      | 52 |
| TABLA 4 FASE II. VALIDACIÓN DEL DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE..... | 54 |
| TABLA 5 FASE III. VALIDACIÓN DEL MODELO DE CASOS DE USO DEL SISTEMA.....                   | 56 |
| TABLA 6 PRIMERA TAREA DE LA FASE I.....                                                    | 60 |
| TABLA 7 SEGUNDA TAREA DE LA FASE I.....                                                    | 60 |
| TABLA 8 TERCERA TAREA DE LA FASE I.....                                                    | 61 |
| TABLA 9 CUARTA TAREA DE LA FASE I.....                                                     | 62 |
| TABLA 10 PRIMERA TAREA DE LA FASE II.....                                                  | 63 |
| TABLA 11 SEGUNDA TAREA DE LA FASE II.....                                                  | 65 |
| TABLA 12 TERCERA TAREA DE LA FASE II.....                                                  | 66 |
| TABLA 13 CUARTA TAREA DE LA FASE II.....                                                   | 67 |

|                                                                                 |    |
|---------------------------------------------------------------------------------|----|
| TABLA 14 QUINTA TAREA DE LA FASE II. ....                                       | 68 |
| TABLA 15 SEXTA TAREA DE LA FASE II. ....                                        | 69 |
| TABLA 16 FACTORES DE COMPLEJIDAD.....                                           | 72 |
| TABLA 17 FACTORES DE CONSISTENCIA.....                                          | 76 |
| TABLA 18 PRIMERA TAREA DE LA FASE III.....                                      | 77 |
| TABLA 19 SEGUNDA TAREA DE LA FASE III.....                                      | 78 |
| TABLA 20 TERCERA TAREA DE LA FASE III. ....                                     | 78 |
| TABLA 21 CUARTA TAREA DE LA FASE III. ....                                      | 79 |
| TABLA 22 PLANIFICACIÓN DEL PROCEDIMIENTO.....                                   | 82 |
| TABLA 23 ELEMENTOS DEFINIDOS EN LA LISTA DE CHOQUEO POR LA METODOLOGÍA RUP..... | 86 |
| TABLA 24 SEMÁNTICA DEL DOCUMENTO.....                                           | 87 |
| TABLA 25 FACTOR DE COMPLETITUD.....                                             | 93 |
| TABLA 26 FACTOR DE CONSISTENCIA.....                                            | 96 |
| TABLA 27 FACTOR DE CORRECTITUD.....                                             | 98 |
| TABLA 28 FACTOR DE COMPLEJIDAD. ....                                            | 98 |

## INTRODUCCIÓN

Actualmente la producción de software fiable, de calidad, y de fácil mantenimiento, es todavía uno de los retos pendientes de la Ingeniería de software. La mayoría de los errores cometidos provienen de las primeras fases de su ciclo de vida, debido a que el coste de resolver estos problemas crece exponencialmente según avanza el proyecto, resulta de vital importancia centrarse en la mejora de las primeras etapas, en lugar de postergarlo a etapas más tardías. La mayor parte de los enfoques sobre estos temas debería centrarse en la Ingeniería de requisitos, la cual es de suma importancia dentro del proceso de construcción del software.

El desarrollo de la producción de software en la Universidad de las Ciencias Informáticas (UCI), ha incrementado su competitividad a través de la difusión, la mejora continua y el conocimiento en tecnologías de la información. Hoy en día se orientan los esfuerzos hacia el entrenamiento, diagnóstico, consultoría y formación según la demanda de los proyectos productivos de la Universidad, los mismos se ven obligados a realizar grandes cambios en productos prácticamente terminados, debido a inconformidades con las solicitudes de los clientes que en etapas iniciales no fueron entendidas correctamente. Los clientes, por su parte no siempre presentan la preparación y el conocimiento requeridos para comprender los términos que se emplean y las actividades en las que participarán para dar a conocer lo que realmente necesitan. El equipo de desarrollo, muchas veces no presta el interés y el tiempo requerido en las etapas tempranas de desarrollo, a pesar de ser fundamentales y definitorias en la elaboración de todo producto de software, cuya realización ineficiente e inadecuada, influye negativamente en la satisfacción de las necesidades del cliente, en la entrega en tiempo y en la calidad de los productos, así como en los presupuestos destinados a la confección de los mismos.

Es meritorio destacar que dentro de la Ingeniería de requisitos, debido a los problemas de comunicaciones existentes y a la falta de experiencia acumulada por cada cual, hacen que el proceso de Validación de requisitos sea dentro de esta etapa uno de los más importantes.

El concepto de Validación de requisitos en los sistemas de software es un área de investigación que ha pasado a ser vital dentro de la gestión de requisitos. Es el proceso mediante el cual se especifican y validan los servicios que debe proporcionar el sistema así como las restricciones sobre las que se deberá operar. Consiste en un proceso iterativo y cooperativo de análisis del problema, documentando los resultados en

una variedad de formatos y probando la exactitud del conocimiento adquirido (Ferreira & Loucopoulos, 2001).

En la UCI las técnicas de Validación de requisitos no son utilizadas adecuadamente, ya que los requisitos no son validados a profundidad antes de continuar con el proceso de desarrollo de software. Todos aquellos errores que se han cometido en la etapa de requisitos tienen luego una inmensa repercusión, es decir, son altamente perjudiciales; ante un error, omisión o mala interpretación en los requisitos, los desarrolladores se han visto obligados a rehacer todo el trabajo hecho sobre la base de requisitos incorrectos, provocando un arduo y costoso trabajo de reprogramación para corregirlos.

De acuerdo a la **situación problemática** antes planteada el siguiente trabajo se da inicio con la necesidad de darle solución a lo previamente expuesto, por tal motivo el **Problema Científico** reside en: ¿Cómo mejorar la calidad de los requisitos de software que se entregan al cliente?

Se define como **objeto de estudio** la Ingeniería de Requisitos, profundizando en el estudio de la Validación de requisitos la cual ha sido identificada como el **campo de acción** del presente trabajo.

Para darle solución al problema anteriormente definido se ha determinado como **objetivo general**: Desarrollar un procedimiento para la Validación de requisitos de software que garantice la calidad de los requisitos que le son entregados a los clientes.

Del objetivo general se derivaron los siguientes **objetivos específicos**:

- Elaborar el marco teórico de la investigación.
- Realizar un procedimiento para la Validación de requisitos de software.
- Aplicar el procedimiento en el proyecto MENPET perteneciente al Centro de Tecnologías de Almacenamiento y Análisis Inteligente de Datos de nuestra universidad.
- Analizar los resultados obtenidos.

Teniendo como **Hipótesis**:

Si se desarrolla un procedimiento para la Validación de requisitos de software, se contribuiría a una mejora en la calidad de los requisitos que le son entregados al cliente.

Los métodos científicos utilizados en el desarrollo de la investigación fueron:

Los **métodos empíricos** a través de las entrevistas realizadas a estudiantes, profesores y especialistas en la rama que trabajan en la UCI y que tienen conocimientos acerca del tema de calidad de requisitos e Ingeniería de requisitos, específicamente a aquellos que se dedican en esencia a estudiar la Validación de requisitos o a aquellos que hayan hecho trabajos sobre dichos temas.

Los **métodos teóricos** usados en la investigación sirvieron para estudiar las características del objeto de la investigación, el **método histórico – lógico** se utilizó para analizar la trayectoria del desarrollo de los procesos de Ingeniería de Requisitos en el tema de Validación de requisitos en los diferentes períodos

El **analítico sintético** es el método empleado para el procesamiento de la documentación, así como para precisar las diferentes características y métodos para desarrollar procedimientos de Validación en los temas de Ingeniería de Requisitos.

El presente trabajo de diploma se organizó en 3 capítulos:

En el Capítulo 1. Fundamentación Teórica. Se realiza la fundamentación teórica del tema, haciendo mayor énfasis en el estudio de la Ingeniería de requisitos y sus actividades, destacándose la Validación de requisitos de software como principal objetivo de esta investigación, y algunos aspectos importantes relacionados con las diferentes técnicas de Validación de requisitos utilizadas en el ámbito mundial y en la Universidad de las Ciencias Informáticas.

En el Capítulo 2: Descripción de la propuesta. Se propone el procedimiento a ser utilizado, explicando detalladamente cada una de las actividades, participantes, técnicas y los artefactos de entrada y salida generados en cada actividad.

En el Capítulo 3: Validación de la propuesta. Se hace la validación del procedimiento aplicándose al módulo de gestión de reportes del proyecto MENPET. Además se hace un análisis de los resultados obtenidos, con el objetivo de que aporten experiencias para posteriores proyectos productivos.

### Capítulo 1: Fundamentación Teórica.

#### Introducción

En el presente capítulo se realizará el marco teórico de la investigación, incluyendo además el marco conceptual sobre temas como la Ingeniería de requisitos y calidad de software, centrándose en temas como sus características, importancia, así como las etapas de la misma, haciendo mayor énfasis en la Validación de requisitos de software. Se estudiarán diferentes métodos de Validación de requisitos de software existentes a nivel mundial y en la UCI, dentro de estos métodos se examinarán varias técnicas de Validación de requisitos de software y finalmente se precisará la notación de modelado y herramienta CASE a utilizar.

#### 1.1 Requisitos de software. Definiciones y conceptos.

A continuación se hace referencia a algunas definiciones sobre requisitos de software, citada por autores de diferentes literaturas.

Un requisito es una representación de lo que debe hacer el sistema, o de las propiedades o atributos que debe tener para satisfacer las necesidades y expectativas de los clientes, sirve como puente de entendimiento entre estos y el equipo de desarrollo; y cuya presencia y correcta implementación en el producto o sistema pueden ser comprobadas mediante un procedimiento finito y a costes razonables. (Sommerville, Septima Edición 2008)

“i) Una condición o capacidad que necesita el usuario para resolver algún problema o alcanzar un objetivo.

ii) Condición o capacidad que debe cumplir o poseer un sistema o componente del sistema para satisfacer un contrato, estándar, especificación u otro documento formalmente impuesto.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

iii) Una representación documental de una condición o capacidad como en i o en ii. (IEEE, Marzo 2006)”

“Los requerimientos del sistema están influenciados por las preferencias, aversiones y prejuicios de los usuarios y por cuestiones políticas y organizacionales.

Los requerimientos para un sistema son la descripción de los servicios proporcionados por el sistema y sus restricciones operativas. Estos requerimientos reflejan las necesidades de los clientes de un sistema que ayude a resolver algún problema” (Somerville, Sexta Edición)

“Los requisitos son especificaciones de lo que debe ser implementado. Son descripciones de lo que debe hacer el sistema, o de propiedades o atributos que debe tener. Ellos pueden constituir restricciones al proceso de desarrollo del sistema.

Especificaciones de las características, cualidades, capacidades y comportamiento de un sistema, descritos de manera entendible para el equipo de desarrollo, clientes y usuarios y en función de la satisfacción de sus necesidades. Constituyen la base para la medición de la calidad del producto final.” (Torres)

En el marco del presente trabajo se adopta una definición para requisitos apoyada en los conceptos de las autoras, las cuales definen que un requisito es una representación de lo que debe hacer el sistema, o de las propiedades o atributos que debe tener para satisfacer las necesidades y expectativas del cliente y usuarios finales, que sirve como puente de entendimiento entre estos y el equipo de desarrollo; y cuya presencia y correcta implementación en el producto o sistema pueden ser comprobadas mediante un procedimiento finito. (Teruel)

### **1.1.1 Requisitos funcionales y requisitos no funcionales.**

Los requisitos de software están divididos en funcionales y no funcionales, seguidamente se muestran algunas de las características específicas de cada uno de estos tipos de requisitos.

Los requisitos funcionales son parte de los requerimientos del sistema e indican características y restricciones sobre la funcionalidad del software.

### **Requisitos funcionales:**

Algunos tipos de requisitos funcionales son:

**Requisitos sobre la actualización de datos:** características sobre las funciones que cambian la información del sistema. Debe estudiarse de qué forma y bajo qué restricciones el usuario desea que se introduzcan nuevos datos, se cambien los que ya existen o se eliminen.

**Requisitos sobre la estructura de la información:** características de los datos que el software maneja, Requisitos sobre las consultas y los informes: requisitos sobre la interacción con otros sistemas. (Sommerville, 1997)

Los requisitos no funcionales son restricciones de los servicios o funciones ofrecidos por el sistema, incluyen restricciones de tiempo sobre el proceso de desarrollo y estándares.

### **Requisitos No Funcionales:**

Los requerimientos no funcionales a menudo se aplican al sistema en su totalidad, normalmente apenas se aplican a características o servicios individuales del sistema.

Los Requisitos no funcionales forman una parte importante en el logro de la calidad de los productos de software, principalmente para que clientes y usuarios puedan valorar las características no funcionales del producto. Cumpliendo con toda la funcionalidad requerida entonces las propiedades no funcionales, como cuán usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

Los Requisitos no funcionales luego de ser evaluados son una parte indispensable en la evaluación de la arquitectura, siendo esta tarea una manera más económica de evitar desastres. Cuanto más temprano se encuentre un problema en un proyecto de software,

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

mejor 'El costo de arreglar un error durante las fases tempranas, es mucho menor al costo de arreglar ese mismo error en la fase final. (Sommerville, 1997)

Algunos tipos de requisitos no funcionales:

**Requisitos de rendimiento:** Límites al rendimiento (para aquellas aplicaciones donde existan) y volúmenes de información que el software debe tratar.

**Requisitos de frecuencia de tratamiento:** Características sobre la frecuencia con que se ejecutan las diferentes funciones del software.

**Requisitos de seguridad:** Características de control de acceso al software y copias de seguridad, entre otros relacionados con la seguridad del sistema y la información.

Existen otros dos grupos de requisitos los cuales contienen los requisitos funcionales y los no funcionales.

**Requerimientos de usuarios:** son declaraciones en un lenguaje natural y en diagramas de los servicios que se espera que el sistema proporcione y de las restricciones bajo las cuales debe funcionar.

**Requerimientos del sistema:** establecen con detalles las funcionalidades, servicios y restricciones operativas del sistema.”

En el marco del presente trabajo se adopta una definición para requisitos apoyada en los conceptos de las autoras, planteando que lo ideal sería emplear una combinación de enfoques que permita que los requisitos sean lo más comprensibles posible para los clientes y usuarios finales y que estén lo suficientemente claros, precisos y concisos para el equipo de desarrollo, de manera que lo guíen en la construcción del sistema correcto. (Sommerville, Séptima Edición 2008)

### 1.2 Ingeniería de requisitos de software.

A continuación se muestran una serie de conceptos referentes a la Ingeniería de requisitos, destacando además las características, importancia, y etapas de la misma.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

La fase de captura de requisitos de cualquier ciclo de desarrollo de sistemas ha evolucionado en la última década a pasos agigantados al pasar de una concepción reduccionista donde se asumía que los requisitos eran proporcionados única y exclusivamente por los clientes, a una serie compleja de actividades de comunicación y descubrimiento, por parte de los ingenieros de requisitos, de las necesidades de los usuarios. La relativa corta vida de la disciplina de la Ingeniería de requisitos (IR) y el afán de la comunidad en resolver los problemas atinentes a ella, ha generado un sinnúmero de métodos, metodología, procesos y herramientas que en realidad no han llegado a resolver de manera satisfactoria todos los inconvenientes que se presenta en esta fase de cualquier proyecto de desarrollo, entre ellos están: falla en los mecanismos de comunicación entre clientes y analistas; etapas, actividades y documentos sin clara delimitación conceptual, escasa reutilización de requisitos y poca gestión del proceso de la IR .(Loucopoulos, 1995)

“La Ingeniería de requisitos de software es la disciplina encargada de establecer los servicios que un sistema debe suministrar, así como, las restricciones bajo las cuales el sistema debe operar.” (Somerville, 1996).

Proceso de estudio de las necesidades de los usuarios con el objeto de llegar a una definición del sistema hardware/software. El proceso de estudio y refinamiento de un sistema. (IEEE, Marzo 2006)

Trabajo sistemático de desarrollo de requisitos, a través de un proceso iterativo y cooperativo de análisis del problema, documentando los resultados en una variedad de formatos y probando la exactitud del conocimiento adquirido. (Loucopoulos, et al., 2000)

El Software Engineering Institute en su glosario de términos define Ingeniería de requisitos de la siguiente manera: “La Ingeniería de requisitos involucra todas las actividades del ciclo de vida dedicadas a la identificación de los requisitos de usuario, análisis de requisitos para derivar requisitos adicionales, documentación de los requisitos como una especificación y Validación de los requisitos documentados contra las necesidades del usuario, así como los procesos que soportan estas actividades.” (SEI, 2008).

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

“La Ingeniería de requisitos es el uso sistemático de procedimientos, técnicas, lenguajes y herramientas para obtener con un coste reducido el análisis, documentación, evolución continua de las necesidades del usuario y la especificación del comportamiento externo que satisfaga las necesidades del usuario” (PRESSMAN '05) El resultado es una “colección estructurada de información, que contiene los requisitos del sistema” (IEEE '98a), esto es, una Especificación de Requisitos de Software (ERS).

La Ingeniería de requisitos es la disciplina de la Ingeniería de Software, donde se identifica el propósito del sistema, dirección y alcance. Consiste en un conjunto de actividades y transformaciones que pretenden comprender las necesidades de un sistema software y convertir la declaración de estas necesidades en una descripción completa, precisa y documentada de los requerimientos del sistema siguiendo un determinado estándar. Los requisitos constituyen el enlace entre las necesidades reales de los clientes, usuarios y otros participantes vinculados al sistema (stakeholders). La Ingeniería de requisitos se ubica en el punto de encuentro entre lo informal y lo formal del desarrollo de software, facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional. Este proceso debe lidiar con diferentes puntos de vista, usar una combinación de procedimientos, herramientas y personas. Todo este proceso acontece en un universo de discurso con actores reales, por lo que se puede considerar un proceso centrado en las personas.

El objetivo principal de la Ingeniería de requisitos es determinar lo que se pretende de un artefacto de software. Para ello, se proponen métodos, técnicas y herramientas cuyo fin es sistematizar el proceso de definición de requisitos Si bien estos métodos intentan asegurar un máximo de completitud y calidad en los modelos resultantes, pretender obtener un documento de requisitos completo es una utopía. El ingeniero de requisitos se enfrenta con un Universo de Discurso (UdeD) que nunca parece terminar de conocer.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

A manera de resumen se puede decir que el proceso de Ingeniería de requisitos es un conjunto estructurado de actividades que sirven para derivar, analizar, documentar, validar y mantener los requisitos de un sistema (hardware, software o hardware + software) haciendo uso de procedimientos técnicas y herramientas y en función de las necesidades del usuario.

### **1.2.1 Características de la Ingeniería de requisitos.**

Las metodologías y procedimientos consultados referentes a la IR están de acuerdo con las siguientes características.

La IR es un proceso iterativo, ya que al ser un proceso de descubrimiento y comunicación, difícilmente llegará a realizarse en forma lineal.

Los requisitos no siempre son entregados en su totalidad por los clientes y usuarios, así que los ingenieros de requisitos también deben saber descubrirlos.

Los límites de las actividades de IR son difíciles de establecer por la misma naturaleza del proceso.

No hay claridad ni consenso en cuanto a los productos que se deben tener al final del flujo de trabajo. Los requisitos pueden evolucionar tan rápidamente que pueden cambiar antes de haber concluido el desarrollo del sistema. (Chaves, 1997)

### **1.2.2 Importancia de la Ingeniería de requisitos de software.**

Los principales beneficios que se obtienen de la Ingeniería de requisitos son:

- Permite gestionar las necesidades del proyecto en forma estructurada: Cada actividad de la Ingeniería de requisitos consiste de una serie de pasos organizados y bien definidos.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: La Ingeniería de requisitos proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

- Disminuye los costos y retrasos del proyecto: Muchos estudios han demostrado que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la Especificación de Requisitos.
- Mejora la calidad del software: La calidad en el software tiene que ver con cumplir un conjunto de requisitos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.).
- Mejora la comunicación entre equipos: La especificación de requisitos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.
- Evita rechazos de usuarios finales: La Ingeniería de requisitos obliga al cliente a considerar sus requisitos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto. (Chaves, 1997)

### 1.2.3 Etapas de la Ingeniería de requisitos de software.

El proceso de Ingeniería de requisitos, consta de un conjunto de etapas las cuales proporcionan una mayor organización durante todo el flujo de trabajo: Elicitación de requisitos, Análisis y negociación de requisitos, Especificación de requisitos, Validación de requisitos, Administración de requisitos. Todas estas etapas se muestran en la figura 1.

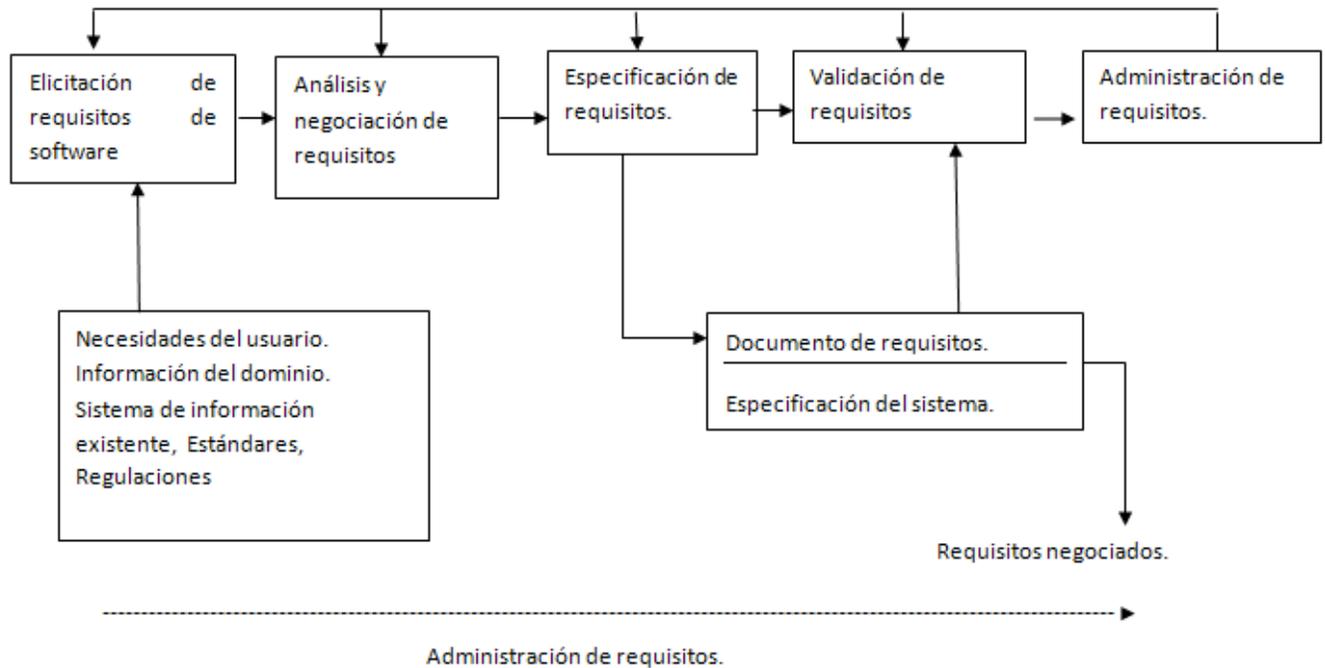


Ilustración 1 Etapas de la Ingeniería de requisitos

### 3.1 Elicitación de requisitos de software.

La etapa de elicitación de requisitos abarca la primera y quizás más importante fase dentro del desarrollo de un sistema informático. Uno de los retos más importantes de la elicitación de requisitos es garantizar que los requisitos del sistema sean consistentes con las necesidades de la organización donde se utilizará el mismo y con las futuras necesidades de los usuarios. En esta etapa se obtiene información sobre el dominio del problema y sistema actual, se identifican y revisan los objetivos del sistema. Es la actividad de la IR en la cual se estudia el dominio del problema y se interactúa con los clientes y usuarios para obtener y registrar información sobre sus necesidades. Antes de identificar los requisitos que el sistema debe cumplir, es conveniente conocer el ambiente y los procesos que se desarrollan dentro de la organización donde el sistema a construir va a ofrecer sus servicios, ésta es la

etapa para recolectar y obtener toda la información posible y necesaria para modelar la organización en estudio. (Sommerville, Septima Edición 2008)

Tareas de la elicitación de requisitos:

- Obtener información sobre el dominio del problema y el sistema actual.
- Preparar y realizar las secciones de elicitación/negociación.
- Identificar/Revisar los objetivos del sistema.
- Identificar/Revisar los requisitos de información.
- Identificar/Revisar los requisitos funcionales.
- Identificar/Revisar los requisitos no funcionales.
- Priorizar objetivos y requisitos.

### ***3.2 Análisis y negociación de los requisitos de software.***

Una vez recopilados los requisitos, el producto obtenido configura la base del análisis de requisitos. Los requisitos se agrupan por categorías y se organizan en subconjunto, se estudia cada requisito en relación con el resto, se examinan los requisitos en su consistencia, completitud y ambigüedad, y se clasifican en base a las necesidades de los clientes/usuarios.

Es corriente en clientes y usuarios solicitar más de lo que puede realizarse, consumiendo recursos de negocios limitados. También es relativamente común en clientes y usuarios el proponer requisitos contradictorios, argumentando que esa versión es “esencial por necesidades especiales”. (M. Griselda Báez, 2001)

El ingeniero del sistema debe resolver estos conflictos a través de un proceso de negociación. Los clientes, usuarios y el resto de intervinientes deberán clasificar sus requisitos y discutir los posibles conflictos según su prioridad. Los riesgos asociados con cada requisito serán identificados y analizados. Se efectúan estimaciones del esfuerzo de desarrollo que se utilizan para valorar el impacto de cada requisito en el costo del proyecto y en el plazo de entrega. Utilizando un procedimiento iterativo, se irán eliminando requisitos, se

---

irán combinando y/o modificando para conseguir satisfacer los objetivos planteados. (M. Griselda Báez, 2001)

### ***3.3 Especificación de requisitos de software.***

La especificación de requisitos es la actividad donde se genera un documento que contiene una descripción detallada de las funcionalidades y necesidades del sistema que se está desarrollando, describe la forma en cómo se harán las funciones, describiendo los requerimientos funcionales y los no funcionales, además describe el alcance del sistema.

Especificación de Requisitos Software (ERS) es una descripción completa del comportamiento del sistema que se va a desarrollar. (Torres)

El objetivo de esta etapa es la formalización (definición, análisis y verificación) de los requisitos de sistema de una aplicación, a partir de los requisitos de los usuarios, objetivos de negocio, restricciones de diseño, estándares externos, etc. identificado en la fase de viabilidad de la misma, generando los siguientes entregables:

- Documento de requisitos funcionales y técnicos de la aplicación.
- Plan de revisión y seguimiento del documento durante el desarrollo del proyecto.
- Matriz de trazabilidad de requisitos.
  - Lista que referencia cada requisito con los puntos o párrafos de los documentos de especificación de los usuarios; además, para cada requisito propone un método de validación (test, verificación visual, inspección de código...).
- Plan de pruebas (opcional).

### ***3.4 Validación de requisitos de software.***

En la actividad de Validación de requisitos de software se debe confirmar que los requisitos, una vez analizados y resueltos los posibles conflictos, corresponden realmente a las necesidades de clientes y usuarios. [Pohl 1997, Sawyer y Kontoya 1999].

El objetivo principal de esta actividad y sus entradas y salidas son los siguientes:

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

Asegurarse de que los requisitos describen el producto deseado: aunque las actividades de elicitación y análisis se hayan realizado correctamente, siempre es necesario confirmar que los requisitos obtenidos se corresponden realmente con los que los clientes y usuarios desean, de forma que se evite la situación en la que el producto final, que puede ser técnicamente correcto, no es satisfactorio.

Las actividades de Validación conllevan generalmente la elicitación de nuevos requisitos debido a que, a medida que el nuevo sistema se va perfilando, suelen ir apareciendo nuevas necesidades que hasta entonces estaban ocultas, sobre todo mediante la utilización de prototipos (Davis 1995).

### Entradas y salidas

Las entradas de esta actividad son los requisitos, los requisitos, en el caso de que los clientes y usuarios tengan conocimientos suficientes para comprender las notaciones utilizadas, el prototipo y la información de Validación proveniente de clientes y usuarios.

La salida de esta actividad son las versiones validadas, total o parcialmente, de los requisitos, en el caso de que los clientes y usuarios tengan conocimientos suficientes para comprender las notaciones utilizadas, y del prototipo. En el caso de que se detecten conflictos o se eliciten nuevos requisitos se repite el ciclo completo de actividades.

La Validación de requisitos examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedades, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto.

### ***3.5 Gestión de requisitos de software.***

La gestión de requisitos es el proceso de comprender y controlar los cambios en los requerimientos del sistema. Es necesario mantenerse al tanto de los requisitos particulares y mantener vínculos entre los requisitos dependientes de forma que se pueda evaluar el impacto de los cambios en los requisitos. Hay que establecer un proceso formal para implementar las propuestas de cambio y vincular estos a los requisitos del sistema. El

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

proceso de gestión de requisitos debería empezar en cuanto esté disponible una versión preliminar del documento de requisitos, pero se debería empezar a planificar cómo gestionar los requisitos que cambian durante el proceso de obtención de los requisitos.

La información de rastreo de los requisitos vincula los requerimientos dependientes en el documento de requisitos. Esta información se utiliza para evaluar cómo es probable que muchos requisitos se vean afectados por un cambio propuesto y la magnitud de los cambios consecuentes en los requisitos.

Un proceso de gestión de cambios es el conjunto de actividades que evalúa el impacto y coste de los cambios.

La gestión de cambio de los requisitos se debe aplicar a todos los cambios propuestos en los requisitos. La ventaja de utilizar un proceso formal para gestionar el cambio es que todos los cambios propuestos son tratados de forma consistente y que los cambios en el documento de requisitos se hacen de forma controlada.

El proceso del ciclo de vida de la gestión de requisitos, debería ser flexible y adaptable para reunir las necesidades del proyecto. Las características del alcance e implementación del proceso del ciclo de vida de la gestión de requisitos en un proyecto, variará dependiendo de algunos factores claves.

- Tamaño y complejidad del proyecto.
- Experiencia del personal del proyecto.
- Experiencia de los clientes del proyecto.
- Dominio de la aplicación.
- El propósito y uso de esta aplicación.

La gestión de requisitos es una parte esencial para controlar la complejidad, riesgo, alcance del proyecto, y definir los roles y criterios para un software o un proyecto de negocio exitoso. La administración de requisitos es el proceso de establecer las capacidades básicas de un proyecto. Esto es más que una manifestación del propósito para una aplicación: Es fundamentalmente acerca de administración de cambios y el mantenimiento de los requisitos en un estado que refleje exactamente el software que será, o que debe ser, construido.

Incluye la administración del cambio, cualidades de los requisitos y rastreo de los requisitos. (Caro)

### **1.3 Validación de requisitos de software. Particularidades.**

La validación es la actividad de la ingeniería de requisitos que permite demostrar que los requerimientos definidos en el sistema son los que realmente quiere el cliente; además revisa que no se haya omitido ninguno, que no sean ambiguos, inconsistentes o redundantes. Tiene como misión demostrar que la definición de requisitos define el sistema que quiere el usuario. El objetivo de la Validación de requisitos es descubrir problemas en el documento de requisitos antes de comprometer recursos a su implementación.

Como resultado de esta validación se produce una línea-base (baseline). El documento debe revisarse para descubrir omisiones, conflictos, ambigüedades, y comprobar la calidad del documento y su grado de adhesión a estándares.

#### **1.3.1 Diferencia entre verificación y Validación de requisitos de software.**

En el contexto del presente trabajo se hace necesario destacar la diferencia entre los conceptos de verificación y Validación de requisitos de software con el objetivo de eliminar confusiones que puedan surgir.

La ECSS-E-40 plantea:

##### **Verificación**

- Confirmación mediante examen y evidencias objetivas que un elemento cumple con los requisitos específicos.
- El proceso de verificación debe asegurar que para cada actividad existen las entradas y especificaciones adecuadas, y que las salidas de esa actividad son correctas y consistentes con dichas entradas y especificaciones.

¿Estamos construyendo el producto correctamente?

---

### Validación

- Confirmación mediante examen y evidencias objetivas que un software cumple con los requisitos particulares para un uso específico.
- El proceso de validación asegurar que en el producto final software las funciones baseline y los rendimientos requeridos están correcta y completamente implementados.
- ¿Estamos construyendo el producto correcto?

Diferencia entre Validación & Verificación y pruebas

- La segunda es una técnica de Validación & Verificación.
- Objetivo final: tener confianza en que el software va a funcionar. El nivel de confianza requerido depende de:
  - Propósito del sistema: cuan crítico es el software para la organización
  - Expectativas del usuario.  
En cuanto el software no puede fallar. La moral del usuario a este respecto es baja y pierde confianza.

### 1.4 Métodos de Validación de requisitos de software.

Se hace alusión a continuación a varios de los métodos de Validación de requisitos de software, los cuales han sido utilizados durante el diseño del procedimiento para la Validación de requisitos de software.

Los métodos para validación y verificación pueden clasificarse en:

**Métodos estáticos:** se centran en el análisis y comprobación de la representación del sistema, incluyendo documentos, diagramas y código.

**Métodos dinámicos:** implican ejecutar algún tipo de implementación del sistema. Podría parecer que sólo con los métodos estáticas es suficiente pero esto no tiene sentido puesto que los métodos estáticas se orientan más bien a la verificación, no pueden demostrar que el sistema satisface las expectativas del usuario (validación). (Escalona, 2002)

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

En este trabajo de diploma el interés es utilizar ambos métodos con el fin de poder detectar y corregir errores en los requisitos.

A continuación se detallan las características de las técnicas de Validación de requisitos seleccionadas para hacer uso de ellas en el procedimiento para la Validación de requisitos de software.

### 1.4.1 Revisiones.

Una revisión del software es una evaluación de un elemento del software para determinar diferencias de los resultados planeados y recomendar mejoras, los procedimientos para las revisiones del software están basados en ANSI/IEEE Std 1028-1988.

Pueden usarse tres tipos de revisiones para la comprobación del software:

- La revisión técnica.
- El Walkthroughs.
- La inspección del software.

Los tres tipos de revisiones son todas las revisiones formales en el sentido que todos tienen objetivos específicos y procedimientos. Todos los tipos de revisión buscan identificar los defectos y diferencias del software contra las especificaciones, planes y normas.

El Reviews es la fórmula más empleada para validación. Un grupo de personas (incluyendo usuarios) se ocupan de revisar el documento de requisitos el cual consta de tres fases:

1. Búsqueda de problemas.
2. Reunión.
3. Acuerdos.

Como guía para identificar problemas habituales, se pueden utilizar listas de comprobación (checklists). Existen checklists adaptadas a distintos tipos de sistemas. Una lista de chequeo (Ver Anexo 2 Lista de chequeo.) debería girar alrededor de los 24 atributos de calidad que debería poseer una Especificación de requisitos de software.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

Esta técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante.

### **1.4.2 Prototipos orientados a clientes y usuarios.**

Permite descubrir con rapidez si el usuario se encuentra satisfecho, o no, con los requisitos (Uso de escenarios/casos de uso).

Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario (Olsina, 1999). Esta técnica tiene el problema de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

### **1.4.3 Auditorias.**

La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una checklist predefinida o definida a comienzos del proceso, es decir sólo una muestra es revisada. (Garzón)

### **1.4.4 Matrices de trazabilidad.**

Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo (Durán, Bernáldez, Ruíz & Toro, 1999). Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.

Pocas son las propuestas existentes que ofrecen técnicas para la realización de la Validación de requisitos y muchas de ellas consisten en revisar los modelos obtenidos en la definición de requisitos con el usuario para detectar errores o inconsistencias, pero aun así, existen algunas técnicas como las planteadas anteriormente que pueden aplicarse hoy en día.

---

### 1.5 Calidad de software. Definiciones.

Se ve la necesidad de destacar algunas definiciones de calidad, producto a que durante el desarrollo del presente trabajo se definió tanto en el problema científico, como en el objetivo principal de esta investigación la necesidad de mejorar la calidad de los requisitos que le son entregados al cliente.

La calidad se ha visto como la capacidad para identificar las necesidades y expectativas de los clientes y demás partes interesadas para satisfacerlas, cumpliendo los requisitos del producto o servicio ofrecido, esto adquiere cada vez más importancia en la gestión de las empresas, de ahí que los gerentes reconozcan que se pueden obtener ventajas competitivas sustanciales mediante el desarrollo de sistemas de gestión de calidad. La calidad se ha convertido en el mundo globalizado de hoy, en una necesidad ineludible para permanecer en el mercado. Por ello los sistemas de gestión de la calidad basados en las normas ISO 9001, que reflejan el consenso internacional en este tema, han cobrado una gran popularidad, y muchas organizaciones se han decidido a tomar el camino de documentarlo e implementarlo. (Miranda, y otros, 2008)

Es difícil en ocasiones dar un concepto apropiado de lo que sería calidad, pero desde el punto de vista de un ingeniero informático, la calidad de un producto de software según los autores de esta investigación no es más que la capacidad de satisfacer los deseos y necesidades de un cliente.

Algunos autores reconocidos en el mundo de la industria de software han dado su criterio de lo que significa para ellos calidad en esta ocasión le mostraremos algunos de estos conceptos que les permitan enriquecer sus conocimientos

Definición de la norma ISO 9000:

“Calidad: grado en el que un conjunto de características inherentes que cumple con los requisitos”. Philip Crosby: “Calidad es cumplimiento de requisitos” William Edwards Deming.

“Calidad es satisfacción del cliente”.

“Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

implícitas que se espera de todo software desarrollado profesionalmente” R. S. Pressman (2002).

“El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas” ISO 8402 (UNE 66-001-92)

De manera que de aquí en lo adelante se puede tratar calidad de software como la forma en que se logra una correspondencia entre las funcionalidades del mismo y la forma en la que se satisfacen las necesidades del cliente de acuerdo a estas funcionalidades y en concordancia con los requisitos establecidos, cumpliendo con las normas y estándares de calidad.

Anteriormente se había planteado que la falta de concordancia con los requisitos se traduce en una falta de calidad, según estudios realizados se demuestra que el 2004 el 71% de los proyectos que no terminan con éxito lo hacen por una mala gestión de requisitos (Ver Anexo 1 Fuentes Standish Group Inc.). La UCI, como centro de desarrollo de software no queda excluida de estos problemas. El conocimiento de términos asociados a este proceso es de vital importancia para lograr una mejor capacitación profesional.

### **1.6 Notación de modelado.**

Con el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC's), han surgido numerosas herramientas que centralizan su atención en el soporte al desarrollo, fomentando de esta forma el intercambio de información y la sistematización de esfuerzos de especialistas de un equipo de trabajo incluso estando distantes geográficamente.

Para el desarrollo del procedimiento para la Validación de requisitos de software se propició el estudio de dos de estas herramientas de modelado: Business Process Modeling Notation (BPMN) y Unified Modelling Language (UML), seleccionándose BPMN debido a que es un estándar muy utilizado en los últimos tiempos en la construcción de procesos de software producidos en la UCI, con el principal propósito de modelar el negocio como modelo de procesos. Esta notación, tiene como objetivo proporcionar una notación fácilmente comprensible por todos los usuarios del negocio, define la notación y semántica de un

Diagrama de Procesos de Negocio (Business Process Diagram- BPD), para ser usado por las personas que diseñan y administran procesos de negocio. Diferenciándose de UML que es un lenguaje estándar, fácil de aprender y permite una comunicación fluida entre los desarrolladores de software, además de ofrecer una amplia variedad de diagramas para visualizar el sistema desde varias perspectivas, conservando un control eficaz.

### 1.6.1 BPMN

Business Process Modeling Notation o BPMN (en español Notación para el Modelado de Procesos de Negocio) es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (workflow). BPMN fue inicialmente desarrollada por la organización Business Process Management Initiative (BPMI), y es actualmente mantenida por el OMG (Object Management Group), luego de la fusión de las dos organizaciones en el año 2005. Su versión actual es la 1.1 y hay una versión futura propuesta, la 2.0.

BPMN es un nuevo estándar de modelado de procesos de negocio, donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes. BPMN está dirigido a gerentes, directores, dueños de empresas, ingenieros de procesos, analistas de negocios, analistas de sistemas, administradores de proyectos, responsables de calidad y todo aquel que necesita definir, documentar y hacer más eficientes sus procesos de negocio con el estándar más avanzado y aceptado a nivel internacional. BPMN define un único diagrama: el de procesos del negocio. En la especificación del mismo se plantean dos objetivos, el primero: ofrecer una notación sencilla de entender por todos los involucrados en el modelado del negocio y el segundo, no menos importante: asegurar que los lenguajes como BPEL4WS puedan visualizarse a través de esta notación. Es un lenguaje gráfico para representar el negocio, facilitando a las organizaciones la habilidad para comunicar esos procedimientos de una manera estándar.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

BPMN está planeado para dar soporte únicamente a aquellos procesos que sean aplicables a procesos de negocios. Esto significa que cualquier otro tipo de modelado realizado por una organización con fines distintos a los del negocio no estará en el ámbito de BPMN.

Proporciona a los negocios la capacidad de entender sus procedimientos internos en una notación gráfica.

Como principales objetivos tiene:

- Proveer una notación que sea fácilmente entendida por todos los usuarios, desde el analista de negocio, el desarrollador técnico y hasta la propia gente del negocio.
- Crear un puente estandarizado para el vacío existente entre el diseño del proceso de negocio y su implementación.
- Asegurar que los lenguajes para la ejecución de los procesos de negocio puedan ser visualizados con una notación común.
- Es usado para comunicar una amplia variedad de información a una amplia variedad de audiencias.

Actualmente hay una amplia variedad de lenguajes, herramientas y metodologías para el modelado de procesos de negocio. La cada vez mayor adopción de la notación BPMN como estándar ayudará a unificar la expresión de conceptos básicos de procesos de negocio así como conceptos avanzados de modelado (por ejemplo manejo de excepciones, compensación de transacciones, entre otros).

El modelado en BPMN se realiza mediante diagramas muy simples con un conjunto muy pequeño de elementos gráficos. Con esto se busca que para los usuarios del negocio y los desarrolladores técnicos sea fácil entender el flujo y el proceso.

Las cuatro categorías básicas de elementos son:

- Objetos de flujo: Eventos, Actividades, Rombos de control de flujo (Gateways.)
- Objetos de conexión: Flujo de Secuencia, Flujo de Mensaje, Asociación.
- Swimlanes (Carriles de piscina): Pool, Lane.
- Artefactos: Objetos de Datos, Grupo, Anotación.

## CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.

---

Estas cuatro categorías de elementos nos dan la oportunidad de realizar un diagrama simple de procesos de negocio (en inglés Business Process Diagram o BPD). En un BPD se permite definir un tipo personalizado de Objeto de Flujo o un Artefacto, si con ello se hace el diagrama más comprensible.

### Ventajas

Considera un único diagrama para la representación de los procesos (BPD, Business Process Diagram).

Pensando para ser asignado con naturalidad a lenguajes de ejecución.

Fácil de entender.

### 1.6.2 UML

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y una regla para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- Especificar: UML permite especificar cuáles son las características de un sistema antes de su construcción.
- Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura re-visión.

---

### 1.7 Herramienta.

Para el desarrollo del procedimiento para Validación de requisitos de software se compararon dos alternativas de desarrollo: desde, utilizar Rational Rose, herramienta CASE que propone entre otras cosas la utilización de cuatro tipos de modelo para realizar un diseño del sistema utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico permitiendo crear y refinar estas vistas creando de esta forma un modelo completo y complejo que representa el dominio del problema y el sistema de software, hasta Visual Paradigm herramienta con un diseño centrado en casos de uso y enfocado al negocio que genera un software de calidad, tiene la particularidad de ser un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación. Posee capacidades de Ingeniería directa (versión profesional) e inversa, modelo y código que permanece sincronizado en todo el ciclo de desarrollo, disponibilidad de múltiples versiones, disponibilidad en múltiples plataformas. Pero si dejar de centrarse en que para darle solución al objetivo fundamental será necesario el diseño de un procedimiento para la Validación de requisitos de software que garantice la calidad de los requisitos que le son entregados al cliente.

#### 1.7.1 Visual Paradigm

Existen varias versiones de Visual Paradigm para UML. La gratuita no permite realizar ingeniería inversa, pero permite crear diagramas y generar código a partir de ellos. Admite la importación y exportación de XML e imágenes, la administración de requerimientos, la creación de esquemas de clases a partir de una base de datos y viceversa, y cuenta con un soporte que facilita el trabajo simultáneo sobre un mismo diagrama entre dos desarrolladores en un tiempo real.

Visual Paradigm para UML es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar 13 tipos de

diagramas diferentes a través de un intuitivo modelado visual. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos.

### **1.7.2 Rational Rose.**

Es una de las herramientas CASE más potente que existe para el modelado, se basa principalmente en la integración que tiene con el resto de las herramientas que lo acompañan.

Posibilita que haya varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo utiliza un proceso de desarrollo iterativo controlado logrando como resultado en cada secuencia de iteraciones una aproximación al análisis, diseño e implementación del sistema. Proporciona mecanismos para realizar la Ingeniería inversa en una buena cantidad de lenguajes de programación que viene predefinido.

Luego de hacer un análisis comparativo de las herramientas antes mencionadas, en el que se tuvieron en cuenta agentes como: validación en tiempo real del modelo, interfaz de usuario configurable, copias de diagramas como imágenes, se obtuvo como resultado la selección de Visual Paradigm como herramienta para el diseño del procedimiento para validar requisitos de software.

### **Conclusiones.**

En este capítulo se han identificado características propias de la Ingeniería de requisitos y sus etapas, demostrándose principalmente que los problemas en la Validación de los requisitos son una de las causas de que los proyectos de software no cumplan los cronogramas, sobrepasen los costos y de que se obtengan sistemas que no satisfacen las expectativas de los clientes y usuarios finales.

Se estudiaron métodos y técnicas importantes sobre Validación de requisitos de software, logrando seleccionar, de acuerdo a sus características, las que serán utilizadas en desarrollo de procedimiento. El estudio evidenció además, las características debido a las cuales se definieron tanto la notación de modelado, como la herramienta que se utilizará en el procedimiento. Todo esto demostró la importancia que posee contar con un procedimiento para la Validación de requisitos de software que permita mejorar la calidad de los requisitos que le son entregados al cliente.

### Capítulo 2: Propuesta de solución.

#### Introducción

En este capítulo se presenta la propuesta de solución del problema planteado, que radica en la creación de un procedimiento para la Validación de requisitos de software (PVRs) que mejore la calidad de los requisitos que le son entregados al cliente, destacándose el propósito y alcance del mismo, así como sus características. Se precisarán elementos tales como: actividades, responsables y productos de entrada y salida del flujo de trabajo que se definirá en el procedimiento. Se mostrará una descripción general y detallada de las fases por las que está compuesto el procedimiento, además de las tareas propuestas para darle cumplimiento a los objetivos propuestos.

#### 2.1 Alcance de la propuesta de solución.

Este procedimiento es aplicable a todos los proyectos de la Universidad de las Ciencias Informáticas que utilicen metodología RUP, debido a que RUP “es un marco de trabajo genérico que puede especializarse para una gran variedad de Sistemas Software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto” (Jacobson, 2000) Se caracteriza por ser iterativo e incremental, guiado por casos de uso y centrado en la arquitectura. “Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto, las iteraciones y construcciones proporcionan ventajas: tareas pequeñas, grupos de trabajo pequeños, una unión con la gestión de riesgos, controles frecuentes, y realimentaciones” (Jacobson, 2000). Los casos de uso expresan lo que los usuarios futuros necesitan y desean, y a partir de ellos se van desarrollando cada una de las funcionalidades hasta lograr el producto final (Jacobson, 2000).

Es además, aplicado a proyectos que utilicen metodología RUP, porque el procedimiento para la Validación de requisitos de software pretende mejorar la calidad de los requisitos de software que le son entregados a los clientes, y es en proyectos grandes donde se utiliza mayormente esta metodología, en los que se ven reflejados la mayor cantidad de

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

inconformidades por parte de los clientes debido a la gran cantidad de requisitos que son capturados, además se propone la aplicación del procedimiento antes planteado a la metodología RUP, porque en él se validan documentos que aportan gran flujo de información referente a los requisitos de software, los cuales son:

- Documento de especificación de requisitos de software (Ver Anexo 3 Estructura del documento de especificación de requisitos de software.
- Modelo de casos de uso del sistema (Ver Anexo 4 Estructura del modelo de casos de uso del sistema.)

### 2.1.1 Propósito de la propuesta de solución.

El procedimiento para validar requisitos de software tiene como propósito aplicarse a todos los proyectos de la Universidad de las Ciencias Informáticas que utilicen metodología RUP, lo que contribuirá a mejorar la calidad de los requisitos que le son entregados al cliente y a minimizar posibles cambios en el desarrollo del software.

### 2.2 Objetivos.

Los objetivos definidos en el procedimiento son los que se muestran a continuación.

- Definir una secuencia de actividades que permita guiar la Validación de requisitos de manera organizada, con eficiencia, dentro del plazo y costos previstos.
- Lograr un alto grado de satisfacción del cliente.
- Definir las plantillas predefinidas por el departamento de calidad de software de la Universidad de las Ciencias Informáticas (Calisoft), necesarias para formalizar y documentar toda la información que se genera durante el proceso de Validación de requisitos.
- Especificar los roles y las responsabilidades de los trabajadores que intervienen en cada una de las actividades propuestas en el procedimiento.

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

- Establecer un conjunto de técnicas que den soporte a las actividades propuestas por el procedimiento.

### **2.3 Características.**

El procedimiento está conformado por un conjunto de actividades organizadas lógicamente y secuencialmente de acuerdo al elemento que se esté analizando, estas actividades servirán como guía en el proceso de Validación de requisitos.

Es un proceso iterativo, debido a que se hace necesario realizar varias iteraciones mientras los documentos a validar no cumplan con las características establecidas en el procedimiento para detectar y corregir errores, por lo que permiten a los desarrolladores realizar versiones cada vez más completas de los dichos documentos. Es incremental ya que aplica varias secuencias de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia produce un incremento del software.

### **2.4 Estructura.**

Para lograr los objetivos del procedimiento propuesto, este se ha dividido en tres fases: fase I, validar los requisitos de software, fase II, validar el documento de especificación de requisitos de software y fase III, validar el modelo de casos de uso del sistema; esta estructura facilitará una mejor organización en las actividades. Estas fases serán explicadas con mayor precisión en el desarrollo del capítulo.

### **2.5 Responsables y productos de entradas y salidas.**

Para la realización de este procedimiento se definieron un conjunto de responsables y productos los cuales se describen a continuación:

| Nombre                   | Descripción                                                                         |
|--------------------------|-------------------------------------------------------------------------------------|
| Asegurador de la calidad | Responsable por parte del equipo de desarrollo de asegurar la calidad del producto. |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

---

|             |                                                                                                                                                       |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| Funcionario | Responsable de atender las necesidades de los desarrolladores, es la persona que posee un conocimiento profundo de todas las actividades del negocio. |
| Analista    | Responsable de corregir los errores detectados.                                                                                                       |

**Tabla 1 Responsables y Productos**

### **Producto de entrada.**

Documento de requisitos: Documento generado por el asegurador de la calidad que contiene un listado de los requisitos, los que fueron seleccionados del documento de especificación de requisitos de software. Este documento se genera con el objetivo de validar los requisitos de software, como entes separados a sus especificaciones. A continuación se muestra su estructura.

---

---

**Listado de Requisitos de software**

<Nombre del proyecto>

<Nombre del producto>

<Versión>

**Control de versiones**

| Fecha | versión | Descripción | Autor |
|-------|---------|-------------|-------|
|-------|---------|-------------|-------|

**Reglas de confidencialidad.**

**Introducción.**

**Definiciones, Acrónimos y Abreviaturas.**

**Referencia**

**Funcionalidades.**

**Ilustración 2 Listado de requisitos de software**

**Documento de Especificación de Requisitos de Software (ERS) (Ver Anexo 3 Estructura del documento de especificación de requisitos de software.**

): Documento que contiene los requisitos funcionales y no funcionales del sistema especificados por escrito. A continuación se muestra su estructura.

---

---

## Especificación de Requisitos

<Nombre del proyecto>

<Nombre del producto>

<Versión>

### Control de versiones

| Fecha | versión | Descripción | Autor |
|-------|---------|-------------|-------|
|-------|---------|-------------|-------|

Reglas de confidencialidad.

Introducción.

Definiciones, Acrónimos y Abreviaturas.

Referencia

Funcionalidades.

**Ilustración 3** Especificación de requisitos de software.

Modelo de Casos de Uso del Sistema (MCUS) (Ver Anexo 4 Estructura del modelo de casos de uso del sistema.): Documento en el que se muestran los casos de uso, sus especificaciones y además los actores de casos de uso del sistema. A continuación se muestra su estructura.

---

---

## Modelo de Casos de Uso del Sistema

<Nombre del proyecto>

<Nombre del producto>

<Versión>

### Control de versiones

| Fecha | versión | Descripción | Autor |
|-------|---------|-------------|-------|
|-------|---------|-------------|-------|

Reglas de confidencialidad.

Introducción.

Definiciones, Acrónimos y Abreviaturas.

Referencia

Funcionalidades.

**Ilustración 4 Modelo de casos de uso del sistema.**

Productos de salida

Documento de No-Conformidades (DNC): Documento emitido por el asegurador de la calidad en el cual quedan plasmados todas las no conformidades encontradas durante la revisión de un documento.

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

Durante la aplicación de este procedimiento se generan varios documentos de no conformidades, con el objetivo de hacerle mas organizado el trabajo a los analistas en el momento de corregir las mismas.

En la siguiente tabla se muestra parte de la estructura del documento de no conformidades.

|  <span style="float: right;">Documento de No-Conformidades</span>                                                                     |        |                               |                                        |                                      |                       |                             |                       |                  |                                             |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-------------------------------|----------------------------------------|--------------------------------------|-----------------------|-----------------------------|-----------------------|------------------|---------------------------------------------|
| <b>Descripción General</b><br>[Descripción de Aspectos Generales a tener en cuenta a la hora de realizar el diseño de las pruebas, incidencias en el momento de su desarrollo y otros aspectos relevantes.]            |        |                               |                                        |                                      |                       |                             |                       |                  |                                             |
| <b>Elementos probados</b><br>[Descripción general o lista de los Elementos Probados, y otros aspectos importantes a tener en cuenta a la hora analizar las No Conformidades Detectadas.]                               |        |                               |                                        |                                      |                       |                             |                       |                  |                                             |
| <b>Elementos no probados y causas</b><br>[Descripción de Aspectos Generales a tener en cuenta a la hora de realizar el diseño de las pruebas, incidencias en el momento de su desarrollo y otros aspectos relevantes.] |        |                               |                                        |                                      |                       |                             |                       |                  |                                             |
| Ele<br>men<br>to                                                                                                                                                                                                       | N<br>o | No<br>confo<br>rmi<br>da<br>d | Aspect<br>o<br>corresp<br>ondient<br>e | Eta<br>pa<br>de<br>det<br>ecc<br>ión | Sign<br>ificat<br>iva | No<br>Sign<br>ificat<br>iva | Reco<br>menda<br>ción | Esta<br>do<br>NC | Respu<br>esta<br>del<br>Equip<br>o<br>Desar |

**CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.**

|                       |     |                                    |                                           |                                 |     |     |     |                                                                                                                               |                                                                                                                                                       |
|-----------------------|-----|------------------------------------|-------------------------------------------|---------------------------------|-----|-----|-----|-------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       |     |                                    |                                           |                                 |     |     |     |                                                                                                                               | rollo                                                                                                                                                 |
| <Nombre del Elemento> | <1> | <Descripción de la No Conformidad> | <Descripción del Aspecto correspondiente> | <Etapas de detección del error> | <X> | <X> | <X> | [Se coloca el estado de la NC y la fecha, cada vez que se revise se deja el estado anterior y se coloca el nuevo con la fecha | [Esta columna comienza a llenar a partir de la 2da iteración, y es responsable del equipo de desarrollo, quien especifica la conformidad con lo encon |

**PROCEDIMIENTO PARA LA VALIDACIÓN DE REQUISITOS DE SOFTWARE.**

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

|  |  |  |  |  |  |  |  |                                                                                                            |                                                                                                                |
|--|--|--|--|--|--|--|--|------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
|  |  |  |  |  |  |  |  | en<br>que<br>se<br>revis<br>ó.]<br>RA:<br>Resu<br>elta<br>PD:P<br>endie<br>nte<br>NP:N<br>o<br>Proc<br>ede | trado<br>o no y<br>en<br>caso<br>de no<br>proce<br>der la<br>no<br>confor<br>midad<br>explic<br>a por<br>qué.] |
|--|--|--|--|--|--|--|--|------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|

**Tabla 2 Estructura del documento de no conformidades**

### 2.6 Descripción del procedimiento para la Validación de requisitos de software.

En la siguiente figura se muestra una imagen de las fases en las que se encuentra dividido el procedimiento, destacándose los diferentes productos de entrada y salida, que permitirá al usuario tener una mejor visión del procedimiento en general.

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

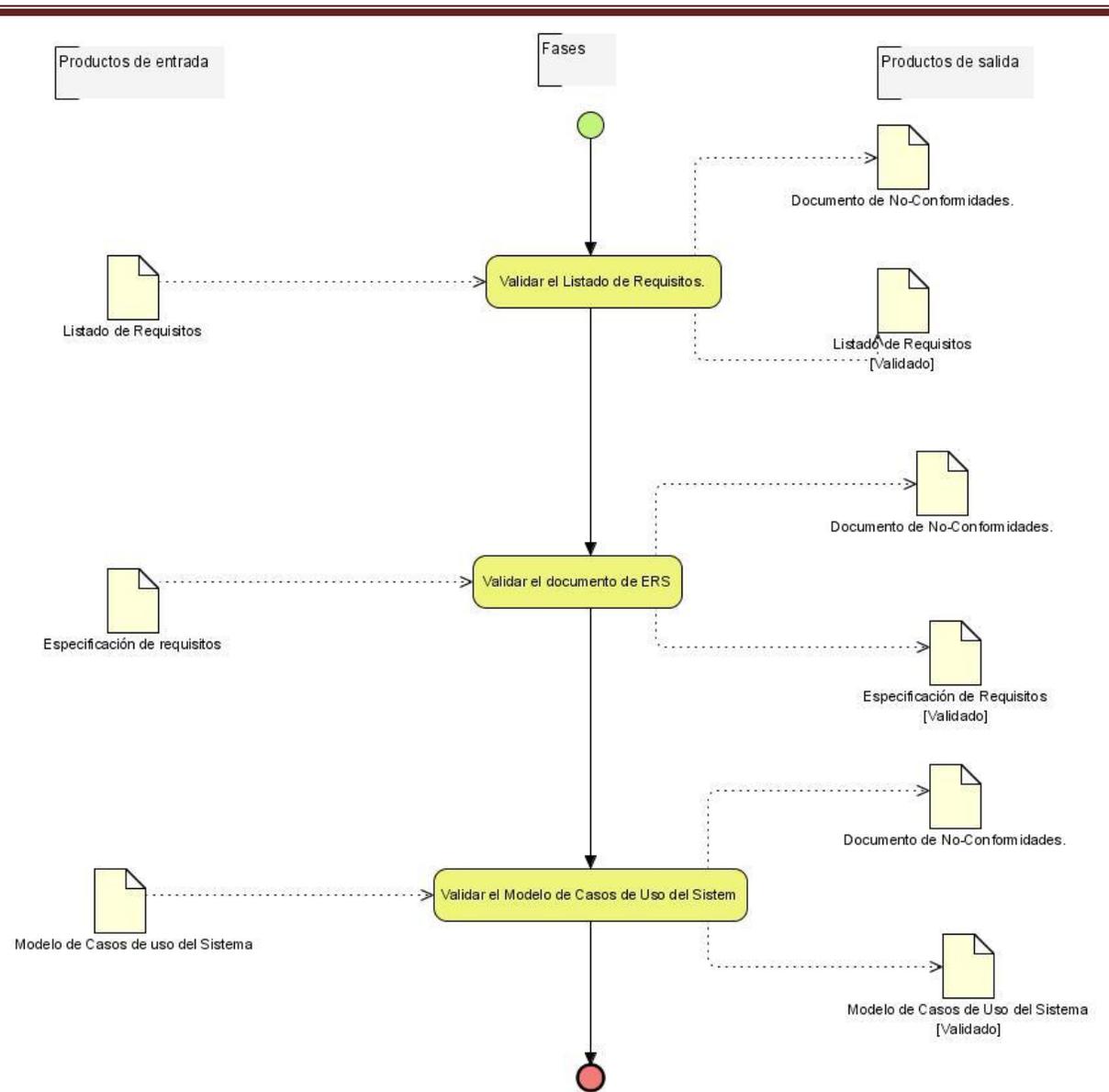


Ilustración 5 Descripción general del procedimiento.

### 2.7 Fases del procedimiento.

A continuación se describe de forma detallada cada una de las fases del procedimiento. Se definen los objetivos, responsables, productos de entradas, salidas, así como las tareas y

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

técnicas a utilizar, para garantizar que los documentos a validar en cada una de las fases cumplan con las características establecidas en el marco de dicho procedimiento.

### 2.8 Fase I. Validación del listado de requisitos de software.

Esta fase tiene como objetivo hacer un análisis detallado de cada requisito tanto funcional como no funcional, contenido en el listado de requisitos, de manera que permita obtener las no conformidades detectadas a partir de los problemas encontrados en los requisitos y reflejadas en el documento de no conformidades para que sean analizadas y corregidas.

| Fase I.              |                                                                                                                                                                                                                                                                                                                                |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre               | Validación de requisitos.                                                                                                                                                                                                                                                                                                      |
| Objetivo             | Validar por separado todo los requisitos que sean contemplados en el listado de requisitos. Para lograr que todos cumplan con las características requeridas.                                                                                                                                                                  |
| Responsable          | Asegurador de la calidad.                                                                                                                                                                                                                                                                                                      |
| Otros Participantes  | Analista, Funcionario.                                                                                                                                                                                                                                                                                                         |
| Productos de entrada | Listado de requisitos.                                                                                                                                                                                                                                                                                                         |
| Productos de salida  | Listado de requisitos (Validados).<br>Documento de no conformidades.                                                                                                                                                                                                                                                           |
| Tareas               | Verificar que el requisito sea no ambiguo.<br>Verificar que el requisito sea especificado por escrito.<br>Verificar que el requisito sea posible de probar o verificar.<br>Verificar que el requisito sea descrito como una característica del sistema.<br>Verificar que el requisito sea lo más abstracto y conciso posibles. |
| Técnica a utilizar   | Revisiones.                                                                                                                                                                                                                                                                                                                    |

**Tabla 3 Fase I. Validación del listado de requisitos de software.**



## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

---

| Fase II              |                                                                                                                                                                                                                                              |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre               | Validación del documento de especificación de requisitos de software.                                                                                                                                                                        |
| Objetivo             | Validar el documento de especificación de requisitos de software detalladamente, para lograr que los requisitos describan en un lenguaje entendible lo que debe hacer el software.                                                           |
| Responsable          | Asegurador de la calidad.                                                                                                                                                                                                                    |
| Otros Participantes  | Analista, Funcionario.                                                                                                                                                                                                                       |
| Productos de entrada | Documento de especificación de requisitos de software.                                                                                                                                                                                       |
| Productos de salida  | Documento de especificación de requisitos de software (validados).<br>Documento de no conformidades.                                                                                                                                         |
| Tareas               | Verificar que la ERS sea correcta.<br>Verificar que la ERS sea No –ambigua.<br>Verificar que la ERS sea completa.<br>Verificar que la ERS sea consistente.<br>Verificar que la ERS sea verificable.<br>Verificar que la ERS sea modificable. |
| Técnica a utilizar   | Auditorías.                                                                                                                                                                                                                                  |

**Tabla 4 Fase II. Validación del documento de especificación de requisitos de software**

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

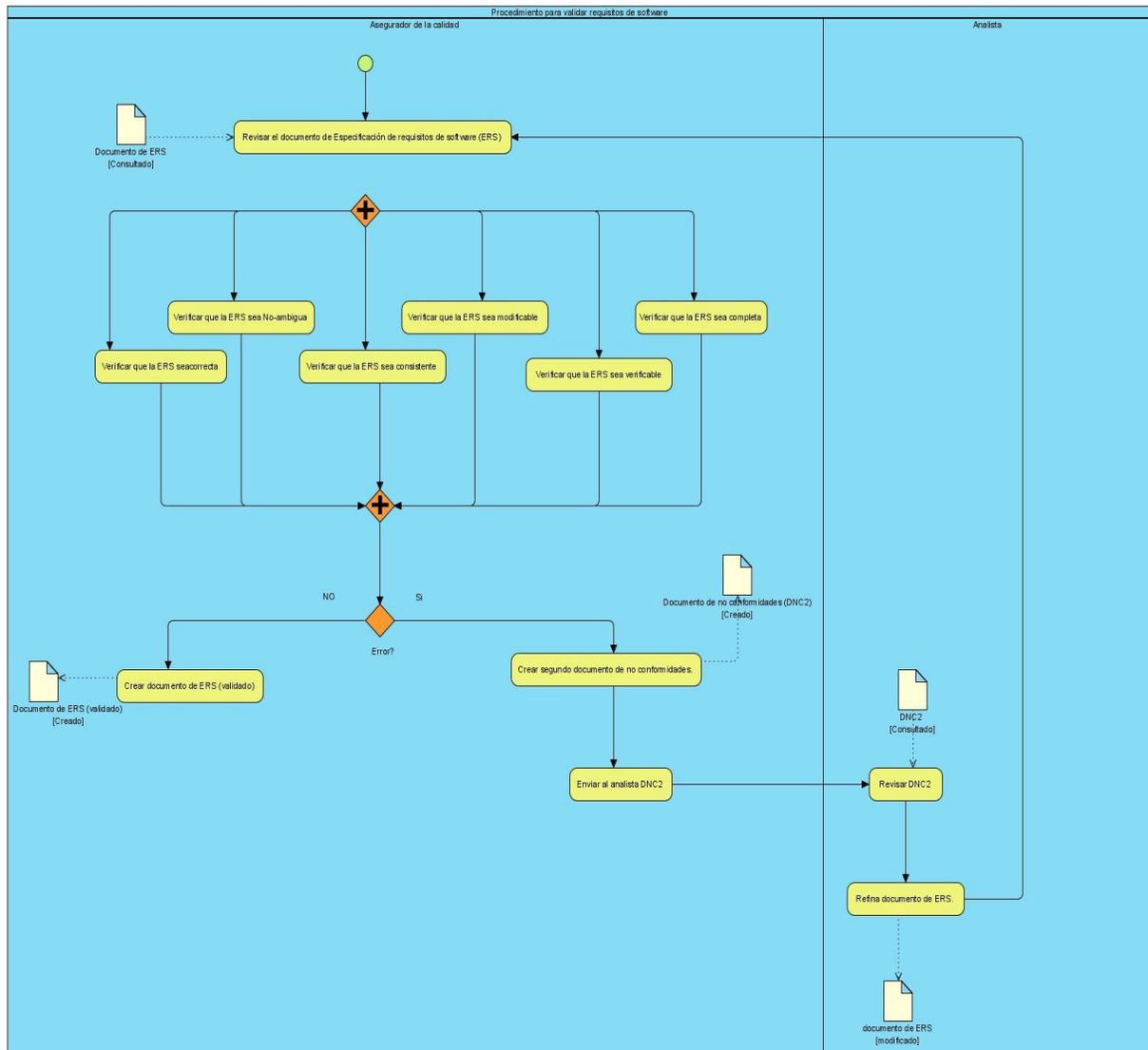


Ilustración 7 Actividades de la fase II.

### 2.10 Fase III. Validación del modelo de casos de uso del Sistema.

El objetivo de esta fase es encontrar mediante métricas definidas, los errores que puedan tener el documento de casos de uso, haciendo mayor énfasis en la descripción de los mismos, ya que es lo que aporta mayor cantidad de detalles sobre los casos de uso.

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

| Fase III             |                                                                                                                                                                                                                                                                                                                                                             |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre               | Validación del modelo de casos de uso del sistema.                                                                                                                                                                                                                                                                                                          |
| Objetivo             | Validar la descripción de Casos de Uso del sistema basado en métricas para lograr que los requisitos se vean reflejados en los casos de uso y para corregir los posibles errores que puedan traer consigo.                                                                                                                                                  |
| Responsable          | Asegurador de la calidad.                                                                                                                                                                                                                                                                                                                                   |
| Otros Participantes  | Analista, Funcionario.                                                                                                                                                                                                                                                                                                                                      |
| Productos de entrada | Modelo de Casos de Uso del Sistema.                                                                                                                                                                                                                                                                                                                         |
| Productos de salida  | Modelo de Casos de Uso del sistema (validado).<br>Documento de no conformidades.                                                                                                                                                                                                                                                                            |
| Tareas               | Verificar que cada requisito se vea reflejado en al menos un caso de uso.<br>Verificar completitud de los casos de uso.<br>Verificar correctitud de los casos de uso.<br>Verificar consistencia de los casos de uso.<br>Verificar complejidad de los casos de uso.<br>Verificar que cada caso de uso se vea reflejado en al menos un prototipo de interfaz. |
| Técnica a utilizar   | Matriz de trazabilidad.<br>Prototipo de interfaz de usuario.                                                                                                                                                                                                                                                                                                |

**Tabla 5 Fase III. Validación del modelo de casos de uso del sistema.**

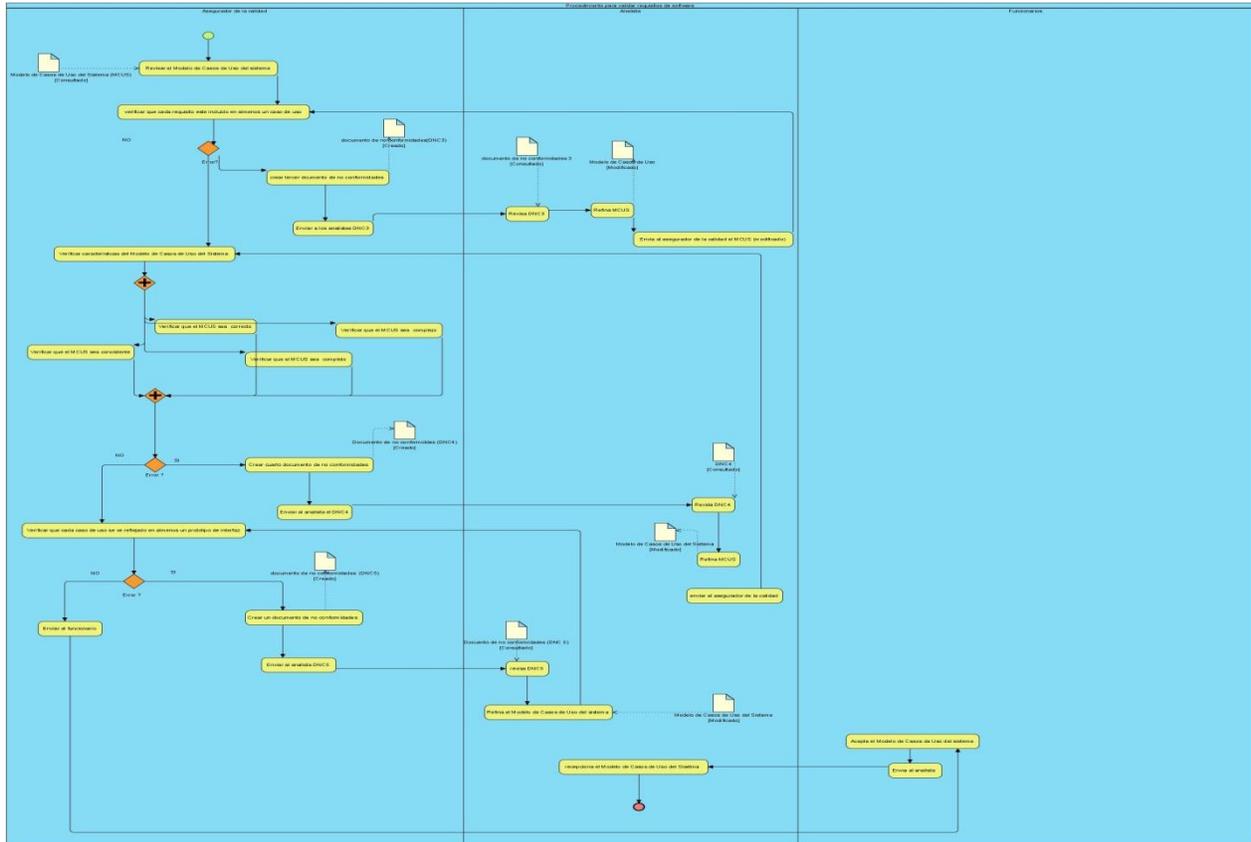


Ilustración 8 Actividades de la fase III.

**2.11 Generalidades del procedimiento para validar requisitos de software.**

En la siguiente figura se representa una vista externa del procedimiento para validar requisitos de software la cual ha sido diseñada en la herramienta seleccionada (Visual Paradigm).



## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

### 2.12 Fase I. Validación del listado de requisitos de software. Particularidades.

A continuación se describen las actividades de la fase I del procedimiento. Se define por cada actividad, los responsables, demás participantes que intervienen en la ejecución de las actividades, productos de entrada, productos de salida, el conjunto de tareas que se deben realizar para cumplir el objetivo de la actividad en cuestión, así como las técnicas que permiten ejecutarlas. Se muestra además, aquellos conceptos que podrían resultar de difícil interpretación para los encargados de la aplicación de estas actividades.

#### Verificar que los requisitos sean especificados por escrito:

Esta tarea consiste en validar que todos los requisitos levantados estén escritos en un documento tangible para los clientes y desarrolladores del software, esto es posible de comprobar cuando pasamos a la fase de validar el listado de requisitos y aplicando la técnica de Validación de requisitos conocida como Revisiones se le solicita al cliente el documento donde todos los requisitos estén especificados por escrito. Para lograr esta tarea se propone en el procedimiento la actividad de Reunión con los clientes.

|                      |                                                                                                                                                                        |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre               | Verificar que los requisitos sean especificados por escrito.                                                                                                           |
| Objetivo             | Verificar que todos los requisitos levantados han sido plasmados en un documento tangible que especifique cada uno de ellos.                                           |
| Responsable          | Asegurador de la calidad.                                                                                                                                              |
| Otros Participantes  | Analista, Funcionario.                                                                                                                                                 |
| Productos de entrada | Listado de requisitos de software.                                                                                                                                     |
| Productos de salida  | Listado de requisitos de software parcialmente validado.                                                                                                               |
| Actividad            | Solicitar al analista listado de requisitos.<br>Verificar que todos los requisitos se encuentren contenidos en el documento recibido.<br>Reunión con los funcionarios. |
| Técnica a utilizar   | Revisiones.                                                                                                                                                            |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

---

**Tabla 6 Primera tarea de la fase I.**

**Verificar que los requisitos sea posible de probar o verificar:**

Esta tarea consiste en enunciar una prueba específica que establezca que se satisface. Un requisito es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes aspectos de verificación: inspección, análisis, demostración o pruebas.

|                      |                                                                                                                                                                                                                             |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre               | Verificar que los requisitos sean posibles de probar o verificar.                                                                                                                                                           |
| Objetivo             | Verificar que todos los requisitos puedan ser probados en una serie finita de pasos. Con el objetivo de saber si el requisito podrá ser o no una funcionalidad capaz de satisfacer las expectativas del futuro del sistema. |
| Responsable          | Asegurador de calidad.                                                                                                                                                                                                      |
| Otros Participantes  | Analista, Funcionario.                                                                                                                                                                                                      |
| Productos de entrada | Listado de requisitos de software                                                                                                                                                                                           |
| Productos de salida  | Listado de requisitos de software parcialmente validados.                                                                                                                                                                   |
| Actividades          | Verificar que los requisitos pueden ser probados.                                                                                                                                                                           |
| Técnica a utilizar   | Revisiones.                                                                                                                                                                                                                 |

**Tabla 7 Segunda tarea de la fase I.**

**Verificar que los requisitos sean descritos como una característica del sistema:**

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

Esta actividad garantiza que los requisitos describan qué debe hacer el sistema y no cómo debe de hacerlo. En la siguiente tabla se recogen los aspectos fundamentales que caracterizan la misma.

|                      |                                                                                                                                                                                          |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre               | Verificar que los requisitos estén escritos como una característica del sistema.                                                                                                         |
| Objetivo             | Verificar que cada requisito diga qué debe hacer y no cómo lo debe hacer. Con el propósito de lograr que cada requisito describa las necesidades objetivas que debe cumplir el software. |
| Responsable          | Asegurador de la calidad.                                                                                                                                                                |
| Otros Participantes  | Analista, Funcionario                                                                                                                                                                    |
| Productos de entrada | Listado de requisitos de software.                                                                                                                                                       |
| Productos de salida  | Listado de requisitos de software parcialmente validados.                                                                                                                                |
| Actividad            | Verificar que cada requisito describa de forma concisa que es lo que debe hacer el sistema.                                                                                              |
| Técnica a utilizar   | Revisiones.                                                                                                                                                                              |

**Tabla 8 Tercera tarea de la fase I.**

### **Verificar que los requisitos sean los más abstractos y concisos posibles:**

Esta actividad describe como deben permitir tanto a los funcionarios como a los desarrolladores abstraerse lo más detalladamente posible del futuro sistema con la menor cantidad de información posible.

|          |                                                         |
|----------|---------------------------------------------------------|
| Nombre   | Verificar abstracción y consistencia en los requisitos. |
| Objetivo | Verificar que cada requisito describa lo más            |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

---

|                      |                                                                                                                                                           |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | detalladamente el producto en la menor cantidad de palabras posible.                                                                                      |
| Responsable          | Asegurador de la calidad.                                                                                                                                 |
| Otros Participantes  | Analista, Funcionario                                                                                                                                     |
| Productos de entrada | Listado de requisitos de software                                                                                                                         |
| Productos de salida  | Listado de requisitos de software parcialmente validados.                                                                                                 |
| Actividad            | Definir con los analistas y funcionarios que es lo que realmente debe hacer el software de forma concreta y precisa.<br>Refinar el listado de requisitos. |
| Técnica a utilizar   | Revisiones.                                                                                                                                               |

**Tabla 9 Cuarta tarea de la fase I**

### **2.13 Fase II. Validación del documento de especificación de requisitos de software. Particularidades.**

A continuación se describen las actividades de la fase II del procedimiento. Se define, por cada actividad, los responsables, demás participantes que intervienen en la ejecución de las actividades, productos de entrada, productos de salida, el conjunto de tareas que se deben realizar para cumplir el objetivo de la actividad en cuestión, así como las técnicas que permitan ejecutarlas. Se muestra además aquellos conceptos que podrían resultar de difícil interpretación para los encargados de la aplicación de estas actividades. Se describen además las tareas para validar el listado de los requisitos.

#### **Verificar que la ERS sea correcta**

La ERS es correcta sí, y sólo sí, contiene todos los requisitos que el software debe satisfacer. No hay una herramienta o procedimiento que garantice esta característica, sin embargo, limitarse a obtener de una vez lo que “quiere” el cliente pone en riesgo el logro de esta característica. Realizar un proceso iterativo, donde se involucren los interesados, seguir

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

la traza de los requisitos y validarlos con los interesados contribuyen a lograr una ERS correcta. (IEEE '98a; LEISHMAN y COOK '02).

Una definición clásica de corrección indica que “un conjunto de requisitos software es correcto sólo si todos los requisitos contenidos representan algo que es requerido para la construcción del sistema y no hay errores que afecten al diseño.”

Un aspecto fundamental a tener en cuenta acerca de la corrección es que depende del usuario final del sistema, Un problema habitual en este ámbito es la omisión (usuarios) y adición (analistas) de información.

|                      |                                                                                                                                                        |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre               | Verificar que la ERS sea correcta.                                                                                                                     |
| Objetivo             | Verificar que la especificación de requisitos sea la que realmente necesita el cliente, para verificar que estemos validando los requisitos correctos. |
| Responsable          | Asegurador de la calidad.                                                                                                                              |
| Otros Participantes  | Analista, Funcionario.                                                                                                                                 |
| Productos de entrada | Documento de especificación de requisitos de software.                                                                                                 |
| Productos de salida  | Documento de especificación de requisitos de software parcialmente validado.                                                                           |
| Actividad            | Analizar con los funcionarios si todos los requisitos contenidos representan algo que es requerido para la construcción del sistema.                   |
| Técnica a utilizar   | Auditorías.                                                                                                                                            |

**Tabla 10 Primera tarea de la fase II.**

**Verificar que la ERS sea No –ambigua**

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

Se considera que un requisito individual, como ítem de información, es ambiguo cuando puede ser interpretado de formas diversas por diferentes personas. Así, el estándar IEEE 830 establece que un requisito es no-ambiguo “si, y sólo si, puede estar sujeto a una única interpretación.”

Asegurar que una ERS está libre de ambigüedades es difícil, pues las personas que intervienen en este proceso provienen de diferentes culturas y tienen sus propios enfoques de percibir los conceptos.

Para elaborar una ERS no ambigua es necesario mantener y documentar un acuerdo entre el equipo de desarrollo y los interesados respecto a los requisitos. Cuando se usa el lenguaje natural para documentar los requisitos pueden introducirse ambigüedades, sin embargo, este es fácilmente comprendido por los interesados, a diferencia de los lenguajes formales, que requieren conocimientos específicos pero permiten reducir la ambigüedad. Anteriormente se sugirió usar una combinación de representaciones a lo cual habría que añadir que si en dicha combinación se hace uso del lenguaje natural debe mantenerse un glosario de términos para disminuir la ambigüedad. (IEEE '98a).

| Nombre               | Verificar que la ERS sea No -ambigua.                                                                                                                                                                                                  |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Objetivo             | Verificar que la ERS tenga una única interpretación, para lograr que los usuarios y desarrolladores tengan un entendimiento común y de esta forma la ERS sea más entendible para cualquiera que necesite hacer uso de dicho documento. |
| Responsable          | Asegurador de la calidad.                                                                                                                                                                                                              |
| Otros Participantes  | Analista, Funcionario.                                                                                                                                                                                                                 |
| Productos de entrada | Documento de especificación de requisitos de software.                                                                                                                                                                                 |
| Productos de salida  | Documento de especificación de requisitos de software parcialmente validado.                                                                                                                                                           |
| Actividad            | 1. Analizar las respuestas de las ERS a las                                                                                                                                                                                            |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

|                    |                                                                                                                                                                                     |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | preguntas contenidas en las listas de chequeo.<br>2. Hacer un análisis cuantitativo y cualitativo de aquellas especificaciones que son ambiguas.<br>3. Refinar el documento de ERS. |
| Técnica a utilizar | Auditorías.                                                                                                                                                                         |

**Tabla 11 Segunda tarea de la fase II.**

### **Verificar que la ERS sea completa.**

El estándar establece que una especificación es completa “si, y sólo si, describe todos los requisitos relevantes para el usuario, incluyendo requisitos asociados con funcionalidad, actuación, restricciones de diseño, atributos o interfaces externas.”

La persona a cargo de este proceso debe tener toda la información necesaria para especificar hasta el más mínimo detalle del sistema. Un problema en este sentido es la falta de un mecanismo que permita detectar la falta de requisitos necesarios para completar el sistema.

La ERS es completa sí, y sólo sí, incluye los siguientes elementos:

Todos los requisitos funcionales y no funcionales son conocidos y documentados en la ERS. Están definidas todas las responsabilidades del sistema respecto a los datos de entrada, tanto válidos como no válidos y respecto a los datos de salida.

En los documentos de la ERS todas las figuras, tablas, diagramas y definiciones de términos están nombrados y referenciados.

Desarrollar de forma iterativa la ERS de software, validarla y usar múltiples representaciones contribuyen a lograr la completitud. (IEEE '98a; LEISHMAN y COOK '02; PRESSMAN '05).

|          |                                                                                                         |
|----------|---------------------------------------------------------------------------------------------------------|
| Nombre   | Verificar que la ERS sea completa.                                                                      |
| Objetivo | Verificar que todos los requisitos funcionales y no funcionales son conocidos y documentados en la ERS. |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

---

|                      |                                                                                                           |
|----------------------|-----------------------------------------------------------------------------------------------------------|
| Responsable          | Asegurador de la calidad.                                                                                 |
| Otros Participantes  | Analista, Funcionario.                                                                                    |
| Productos de entrada | Documento de especificación de requisitos de software.                                                    |
| Productos de salida  | Documento de especificación de requisitos de software parcialmente validados.                             |
| Actividad            | Responder todas las preguntas contenidas en las listas de chequeo definidas para validación en esta fase. |
| Técnica a utilizar   | Auditorias.                                                                                               |

**Tabla 12 Tercera tarea de la fase II.**

### **Verificar que la ERS sea consistente.**

Se considera que una especificación es consistente “si, y sólo si, no hay ningún subconjunto de requisitos descrito dentro de ella que esté en conflicto con cualquier otro.”

El mayor problema en este ámbito es la repetición de información en los requisitos (requisitos repetitivos y redundantes). A veces se usa la misma palabra para designar conceptos diferentes (incoherencias).

La ERS es consistente sí, y sólo sí, ningún subconjunto de la misma entra en contradicción con otro subconjunto (IEEE '98a). Las revisiones técnicas y el uso de diferentes representaciones para los requisitos contribuyen a reconocer la inconsistencia.

|                      |                                                                                                                                                |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre               | Verificar que la ERS sea consistente.                                                                                                          |
| Objetivo             | Verificar que la ERS sea consistente para evitar que no contenga dentro un subconjunto de requisitos que este en conflicto con cualquier otro. |
| Responsable          | Asegurador de la calidad.                                                                                                                      |
| Otros Participantes  | Analista, Funcionario.                                                                                                                         |
| Productos de entrada | Documento de especificación de requisitos de software.                                                                                         |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

---

|                     |                                                                                                                                                            |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Productos de salida | Documento de especificación de requisitos de software parcialmente validados.                                                                              |
| Actividad           | Verificar que las especificaciones no sean redundantes.<br>Atendiendo a las preguntas formuladas en las listas de chequeo.<br>Refinar el documento de ERS. |
| Técnica a utilizar  | Auditorias.                                                                                                                                                |

**Tabla 13 Cuarta tarea de la fase II.**

### **Verificar que la ERS sea verificable.**

Se considera que una especificación es verificable “si lo son cada uno de los requisitos constituyentes”. A su vez, se considera que un requisito individual es verificable “si existe un proceso acotado (en plazo y presupuesto) que permita determinar que el sistema construido satisface lo descrito en el propio requisito”.

La descripción detallada y prueba de los requisitos una vez implementados ayudan considerablemente a su verificación.

|                      |                                                                                                                                                                                                             |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre               | Verificar la ERS sea verificable.                                                                                                                                                                           |
| Objetivo             | Verificar que la ERS pueda ser probada en una serie finita de pasos. Con el objetivo de saber si el requisito podrá ser o no una funcionalidad capaz de satisfacer las expectativas del futuro del sistema. |
| Responsable          | Asegurador de la calidad.                                                                                                                                                                                   |
| Otros Participantes  | Analista, Funcionario.                                                                                                                                                                                      |
| Productos de entrada | Documento de especificación de requisitos de software.                                                                                                                                                      |
| Productos de salida  | Documento de especificación de requisitos de software                                                                                                                                                       |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

---

|                    |                                                                                                                                                                                     |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | parcialmente validados.                                                                                                                                                             |
| Actividad          | Analizar la ERS.<br>Verificar que existe un proceso acotado (en plazo y presupuesto) que permita determinar que el sistema construido satisface lo descrito en el propio requisito. |
| Técnica a utilizar | Auditorias.                                                                                                                                                                         |

**Tabla 14 Quinta tarea de la fase II.**

### **Verificar que la ERS sea modificable.**

Se considera que una especificación es modificable “si su estructura es tal que permite realizar cambios sobre los requisitos que contiene de forma sencilla, completa y consistente, manteniendo la estructura inicial del conjunto”.

La gestión de configuración juega un papel fundamental en este sentido; además de una buena organización de la información y lograr un bajo acoplamiento entre requisitos.

La ERS es modificable sí, y sólo sí, puede ser modificada fácilmente manteniendo la completitud y consistencia. Mantener la traza de los requisitos, organizarlos adecuadamente y usar referencias cruzadas contribuye a hacerla modificable. (IEEE '98a).

Clasificada por importancia. La ERS cumple con esta característica si cada requisito puede ser identificado de manera única y tiene un atributo que indica su importancia desde el punto de vista de los interesados. Todos los requisitos no son igualmente importantes, algunos son críticos para el sistema y otros son deseables, conocer estos atributos es útil para planificar las iteraciones. (IEEE '98a).

|          |                                                                                                            |
|----------|------------------------------------------------------------------------------------------------------------|
| Nombre   | Verificar que la ERS sea modificable.                                                                      |
| Objetivo | Demostrar que la estructura de la ERS es tal que permite hacer cambios sencillos en los requisitos sin que |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

---

|                      |                                                                                                            |
|----------------------|------------------------------------------------------------------------------------------------------------|
|                      | sean afectados ningún otro requisito relacionado.                                                          |
| Responsable          | Asegurador de la calidad.                                                                                  |
| Otros Participantes  | Analista, Funcionario.                                                                                     |
| Productos de entrada | Documento de especificación de requisitos de software.                                                     |
| Productos de salida  | Documento de especificación de requisitos de software parcialmente validado.                               |
| Actividad            | 4. Aplicar las listas de chequeo.<br>5. Analizar los resultados de las preguntas de las listas de chequeo. |
| Técnica a utilizar   | Auditorias.                                                                                                |

**Tabla 15 Sexta tarea de la fase II.**

### **2.14 Validación del modelo de casos de uso del sistema. Particularidades.**

A continuación se describen las particularidades de la fase III del procedimiento. Se definen quienes serán los responsables, así como los demás participantes que intervienen en la aplicación del procedimiento, se podrá observar cuales son los productos de salida y cuáles son las actividades que se llevan a cabo cumplir con el objetivo propuesto. Se muestra además cuales serán las métricas definidas por el equipo de Calisoft, las cuales serán utilizadas para validar completitud, complejidad, correctitud y consistencia.

Completitud Nivel Conceptual: Grado en que se ha logrado definir de forma clara y concisa todos los casos de uso del negocio.

Nivel Especificación: Grado en que se ha logrado detallar todos los casos de uso relevantes.

Consistencia Nivel Conceptual: Grado en que los casos de uso del negocio representan en forma única y no contradictoria los requerimientos funcionales.

Nivel Especificación: Grado en que los casos de uso del sistema describen las interacciones adecuadas entre el usuario y el sistema.

Correctitud Nivel Conceptual: Grado en que los casos de uso del negocio son entendidos y aceptados por el usuario.

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

Nivel Especificación: Grado en que las interacciones actor / sistema soportan adecuadamente el proceso del negocio.

Complejidad: Grado de claridad en la presentación de los elementos que describen el contexto y funcionalidad del sistema.

### Definición de Métricas por Factores.

| Factores de Completitud                                                                                                       | Métricas Asociadas                                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Factor 1. ¿Han sido involucradas todas las áreas funcionales relevantes a las cuales apoyará el sistema?                      | Métrica 1: Número de áreas funcionales relevantes omitidas<br>Umbral: < 10%<br>Acciones sugeridas: Revisar los requerimientos del negocio                                                                  |
| Factor 2. ¿Han sido involucradas todas las áreas funcionales secundarias a las cuales apoyará el sistema?                     | Métrica 2 : Número de áreas funcionales secundarias omitidas<br>Umbral: < 10%<br>Acciones sugeridas: Revisar los requerimientos del negocio                                                                |
| Factor 3. ¿Han sido definidos todos los roles relevantes de usuario encargados de generar/modificar o consultar información?  | Métrica 3: Número de roles relevantes omitidos<br>Umbral: < 10%<br>Acciones sugerida: Revisar el alcance del sistema e Involucrar tipos de usuarios representativos de cada una de las áreas funcionales   |
| Factor 4. ¿Han sido definidos todos los roles secundarios de usuario encargados de generar/modificar o consultar información? | Métrica 4 : Número de roles secundarios omitidos<br>Umbral: < 10%<br>Acciones sugerida: Revisar el alcance del sistema e Involucrar tipos de usuarios representativos de cada una de las áreas funcionales |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

|                                                                                                                                  |                                                                                                                                                                                                                                                                                             |
|----------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Factor 5. ¿Han sido considerados todos los sistemas externos con los cuáles interactuará el sistema?</p>                      | <p>Métrica 5: Número de sistemas externos omitidos<br/>Umbral: &lt; 10%<br/>Acción sugerida: Revisar los sistemas actuales en operación y analizar su interacción con el sistema a desarrollar</p>                                                                                          |
| <p>Factor 6. ¿Se presenta una descripción resumida (descripción de alto nivel) de todos los casos de uso del negocio?</p>        | <p>Métrica 6: Número de casos de uso que no tiene descripción resumida<br/>Umbral: &lt; 10%<br/>Acción sugerida: Completar la descripción resumida del caso de uso</p>                                                                                                                      |
| <p>Factor 7. ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?</p>                          | <p>Métrica 7: Número de requisitos omitidos por caso de uso<br/>Umbral &lt; 10%<br/>Métrica 8: Número de casos de uso que tienen requisitos omitidos<br/>Umbral &lt; 10%<br/>Acción sugerida: Revisar la lista de requisitos para determinar cuáles serán apoyados por cada caso de uso</p> |
| <p>Factor 8. ¿Existen requisitos que no han sido considerados en algún caso de uso?</p>                                          | <p>Métrica 9: Número de requisitos que no son considerados en ningún caso de uso.</p>                                                                                                                                                                                                       |
| <p>Factor 9. ¿Han sido definidos todos los roles de usuario encargados de actividades de soporte/ mantenimiento / auditoría?</p> | <p>Métrica 10: Número de roles de soporte / mantenimiento / auditoria omitidos.<br/>Umbral: &lt; 20%<br/>Acción sugerida: Analizar las funciones de soporte / mantenimiento / auditoria y definir los usuarios que interactuarán con el sistema</p>                                         |
| <p>Factor 10. ¿Se presenta una</p>                                                                                               | <p>Métrica 11: Número de casos de uso que no poseen</p>                                                                                                                                                                                                                                     |

### PROCEDIMIENTO PARA LA VALIDACIÓN DE REQUISITOS DE SOFTWARE.

**CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.**

|                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>descripción detallada (descripción extendida esencial) de todos los casos de uso del negocio?</p>                                       | <p>una descripción extendida.<br/>Umbral &lt; 20%<br/>Acción sugerida: Interactuar con el usuario para realizar la definición extendida del caso de uso que sea consistente con la definición a alto nivel</p>                                                                                                                                                                                      |
| <p>Factor 11. ¿Están todas las acciones del flujo de eventos redactadas en función del responsable?</p>                                    | <p>Métrica 12: Número de acciones del flujo de eventos que no están redactadas en función del responsable.<br/>Umbral &lt; 20%<br/>Acción sugerida: Revisar las responsabilidades tanto del actor (actores) como del sistema<br/>Métrica 13: Número de casos de uso que tienen acciones del flujo de eventos no redactados en función del responsable.<br/>Umbral &lt; 10%<br/>Acción sugerida:</p> |
| <p>Factor 12. ¿Se describen las condiciones de excepción relevantes que debe contemplar cada flujo de eventos?</p>                         | <p>Métrica 14: Número de casos de uso que no describen condiciones de excepción relevantes<br/>Umbral &lt; 20%<br/>Acción sugerida: Revisar las excepciones presentadas en el flujo de eventos que producen un mensaje de error al usuario</p>                                                                                                                                                      |
| <p>Factor 13. ¿Todos los casos de uso del negocio han sido clasificados de acuerdo a su relevancia (primario / secundario / opcional)?</p> | <p>Métrica 15: Número de casos de uso que no han sido clasificados<br/>Umbral: &lt; 10%<br/>Acción sugerida: Hacer reuniones con los usuarios para analizar y priorizar los requisitos de acuerdo a su relevancia.</p>                                                                                                                                                                              |

**Tabla 16 Factores de complejidad**

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

| Factores de Consistencia                                                                                                                      | Métricas Asociadas                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Factor 14. ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario? | <p>Métrica 16: Número de casos de uso que tienen un nombre incorrecto</p> <p>Umbral &lt; 20%</p> <p>Acción sugerida: Modifique el nombre del caso de uso de tal manera que signifique una acción desde el punto de vista del usuario</p>                                                                                                                                                                                             |
| Factor 15. ¿Representa el caso de uso una interacción observable por un actor?                                                                | <p>Métrica 17: Número de casos de uso que no representan una interacción observable por un actor</p> <p>Umbral &lt; 5%</p> <p>Acción sugerida: Elimine el caso de uso e incorpore su funcionalidad como una responsabilidad del sistema dentro de otro caso de uso</p>                                                                                                                                                               |
| Factor 16. ¿No existe solapamiento en la funcionalidad que representan los diferentes casos de uso?                                           | <p>Métrica 18: Número de casos de uso que se solapan</p> <p>Umbral &lt; 10%</p> <p>Acción sugerida: Integrar en uno solo aquellos casos de uso que se solapan. Revisar las acciones repetidas y asignarlas a un solo caso de uso</p>                                                                                                                                                                                                 |
| Factor 17. ¿Existen acciones en el flujo de eventos asignadas a un responsable que no le corresponde?                                         | <p>Métrica 19: Número de acciones del flujo de eventos que no se corresponde la definición de las con el responsable</p> <p>Umbral &lt; 20%</p> <p>Acción sugerida: Revisar las responsabilidades tanto del actor (actores) como del sistema</p> <p>Métrica 20: Número de casos de uso que tienen acciones del flujo de eventos asignados a un responsable que no le corresponde.</p> <p>Umbral &lt; 10%</p> <p>Acción sugerida:</p> |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

|                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Factor 18. ¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos?</p>                                                                                                  | <p>Métrica 21: Grado de adecuación de la descripción del flujo de eventos para un caso de uso<br/>           Acción sugerida: Revise la descripción para que sea definida en el lenguaje del usuario. Asegúrese de definir el responsable de la acción. Establezca claramente las acciones de inicio y fin del caso de uso<br/>           Métrica 22: Número de casos de uso no aceptados<br/>           Umbral &lt; 20%<br/>           Acción sugerida: Realice reuniones con el usuario para mejorar la descripción de los casos de uso</p> |
| <p>Factor 19. ¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o por una condición interna del sistema claramente identificable?</p> | <p>Métrica 23: Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema<br/>           Umbral: &lt; 10%<br/>           Acción sugerida: Complete la definición del caso de uso incluyendo la acción fuera del sistema que da inicio al caso de uso o la condición interna que el sistema tiene controlar para dar inicio al caso de uso</p>                                                                                                                          |
| <p>Factor 20. Si en el caso de uso interviene más de un actor, ¿existe claridad en cuál de ellos es el actor iniciador?</p>                                                                        | <p>Métrica 24: Número de casos de uso con más de un actor que no describe cuál es el actor iniciador<br/>           Umbral: &lt; 20%<br/>           Acción sugerida: Revise los puntos de inicio del caso de uso y asigne el actor que inicia la acción</p>                                                                                                                                                                                                                                                                                   |
| <p>Factor 21. ¿Existe una adecuada separación entre el flujo básico de eventos y los flujos alternos y/o flujos subordinados?</p>                                                                  | <p>Métrica 25: Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos<br/>           Umbral: &lt; 20%<br/>           Acción sugerida: Estructure el caso de uso de manera que separe su funcionalidad básica (caso de uso base) de la funcionalidad repetitiva o alternativa. Si hay pasos</p>                                                                                                                                                                                                       |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

|                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                 | repetitivos forme un caso de uso que lo incluye y los pasos alternativos formen un caso de uso que lo extienda.                                                                                                                                                                                                                                                                                                              |
| Factores de Correctitud                                                                                         | Métricas Asociadas                                                                                                                                                                                                                                                                                                                                                                                                           |
| Factor 22. ¿Existe para cada caso de uso de negocio por lo menos un usuario responsable?                        | <p>Métrica 26: Número de casos de uso que no tienen un usuario responsable</p> <p>Umbral: &lt; 10%</p> <p>Acción sugerida: Analice la responsabilidad que representa el caso de uso y acuerde con los usuarios cuál es el responsable directo de éste</p>                                                                                                                                                                    |
| Factor 23. ¿Representa el caso de uso requisitos comprensibles por el usuario?                                  | <p>Métrica 27: Grado en que los requisitos representados por el caso de uso son comprensibles por el usuario</p> <p>Métrica 28: Número de casos de uso en que los requisitos representados no son comprensibles por el usuario</p> <p>Umbral: &lt; 5%</p> <p>Acción sugerida: Discuta con el usuario la interacción que describe el caso de uso y ajuste dicha descripción de manera que sea comprensible por el usuario</p> |
| Factor 24. ¿Se ajusta la representación del diagrama del caso de uso de acuerdo a lo normado en la metodología? | <p>Métrica 29: Grado en que se ajusta el diagrama del caso de uso a la metodología.</p> <p>Umbral:</p> <p>Acciones Sugeridas:</p>                                                                                                                                                                                                                                                                                            |
| Factor 25. ¿Las interacciones definidas describen la funcionalidad requerida del sistema?                       | <p>Métrica 30: Grado en que las interacciones definidas describen la funcionalidad solicitada por el usuario</p> <p>Métrica 31: Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema</p> <p>Umbral: &lt; 10%</p> <p>Acción sugerida:</p>                                                                                                                                          |
| Factor 26. ¿Las                                                                                                 | Métrica 32: Número de casos de uso que deben ser                                                                                                                                                                                                                                                                                                                                                                             |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

|                                                                                                                       |                                                                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| interacciones definidas introducen mejoras al proceso actual?                                                         | <p>modificados para mejorar el proceso actual</p> <p>Umbral: &lt; 20%</p> <p>Acción sugerida: Analice la situación descrita en la interacción y estudie la manera de mejorar el proceso con el uso de la tecnología informática</p>                               |
| Factor 27. ¿Se ajusta la representación del diagrama del caso de uso de acuerdo a lo normado en la metodología?       | <p>Métrica 33: Grado en que se ajusta el diagrama del caso de uso a la metodología.</p> <p>Umbral:</p> <p>Acciones Sugeridas:</p>                                                                                                                                 |
| Factores de Complejidad                                                                                               | Métricas Asociadas                                                                                                                                                                                                                                                |
| Factor 28. ¿En sistemas relativamente grandes se ha realizado una agrupación de los casos de uso en paquetes?         | <p>Métrica 34: Se hizo partición por paquetes</p> <p>Umbral:</p> <p>Acción sugerida:</p> <p>Métrica 35: Grado en que es adecuada la partición por paquetes</p> <p>Umbral:</p> <p>Acción sugerida:</p>                                                             |
| Factor 29. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación? | <p>Métrica 36: Número de elementos del diagrama que requieren reubicación</p> <p>Umbral: &lt; 30%</p> <p>Acción sugerida: Modifique la ubicación de los elementos del diagrama de manera que los elementos relacionados se encuentren lo más cercano posible.</p> |

**Tabla 17 Factores de consistencia**

A continuación se muestran las actividades necesarias para dar cumplimiento a las actividades previstas en la fase III, las cuales permitirán dar cumplimiento a las características propuestas en la validación del modelo de casos de uso del sistema.

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

|                      |                                                                                                                                          |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre               | Verificar completitud del modelo de casos de uso del sistema.                                                                            |
| Objetivo             | Verificar que todas las partes han sido involucradas en el proceso.                                                                      |
| Responsable          | Asegurador de la calidad.                                                                                                                |
| Otros Participantes  | Analista, Funcionario.                                                                                                                   |
| Productos de entrada | Modelo de Casos de Uso del sistema.                                                                                                      |
| Productos de salida  | Modelo de Casos de Uso del sistema parcialmente validado.                                                                                |
| Actividad            | Aplicar métricas definidas por Calisoft.<br>Realizar prototipo de interfaz de usuario.<br>Refinar documento de casos de uso del sistema. |
| Técnica a utilizar   | Prototipos.                                                                                                                              |

**Tabla 18 Primera tarea de la fase III.**

|                      |                                                                                      |
|----------------------|--------------------------------------------------------------------------------------|
| Nombre               | Verificar Correctitud del modelo de casos de uso del sistema.                        |
| Objetivo             | Verificar que exista para cada caso de uso al menos 1 responsable.                   |
| Responsable          | Asegurador de la calidad                                                             |
| Otros Participantes  | Analista, Funcionario.                                                               |
| Productos de entrada | Modelo de Casos de Uso del sistema.                                                  |
| Productos de salida  | Modelo de Casos de Uso del sistema parcialmente validado.                            |
| Actividad            | 6. Realizar un análisis teniendo en cuenta los resultados de las métricas aplicadas. |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

|                    |                                                                                                               |
|--------------------|---------------------------------------------------------------------------------------------------------------|
|                    | <p>7. Realizar prototipo de interfaz de usuario.</p> <p>8. Refinar documento de casos de uso del sistema.</p> |
| Técnica a utilizar | <p>Matriz de trazabilidad.</p> <p>Prototipo.</p>                                                              |

**Tabla 19 Segunda tarea de la fase III.**

|                      |                                                                                                                                                                                                         |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nombre               | Verificar Consistencia del modelo de casos de uso del sistema.                                                                                                                                          |
| Objetivo             | Verificar que cada requisito esté involucrado con al menos 1 casos de uso                                                                                                                               |
| Responsable          | Asegurador de la calidad                                                                                                                                                                                |
| Otros Participantes  | Analista, Funcionario.                                                                                                                                                                                  |
| Productos de entrada | Modelo de Casos de Uso del sistema.                                                                                                                                                                     |
| Productos de salida  | Modelo de Casos de Uso del sistema parcialmente validado.                                                                                                                                               |
| Actividad            | <p>Crear matriz de trazabilidad teniendo en cuenta los resultados de las meticas aplicadas.</p> <p>Realizar prototipo de interfaz de usuario.</p> <p>Refinar documento de casos de uso del sistema.</p> |
| Técnica a utilizar   | <p>Matriz de trazabilidad.</p> <p>Prototipo.</p>                                                                                                                                                        |

**Tabla 20 Tercera tarea de la fase III.**

|          |                                                                                                                  |
|----------|------------------------------------------------------------------------------------------------------------------|
| Nombre   | Verificar Complejidad del modelo de casos de uso del sistema.                                                    |
| Objetivo | Verificar el grado de claridad en la presentación de los elementos que describen el contexto y funcionalidad del |

## CAPÍTULO 2. PROPUESTA DE SOLUCIÓN.

---

---

|                      |                                                                                                                                                                                                          |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | sistema                                                                                                                                                                                                  |
| Responsable          | Asegurador de la calidad                                                                                                                                                                                 |
| Otros Participantes  | Analista, Funcionarios.                                                                                                                                                                                  |
| Productos de entrada | Documento de Casos de Uso del sistema.                                                                                                                                                                   |
| Productos de salida  | Documento de Casos de Uso del sistema parcialmente validado.                                                                                                                                             |
| Actividad            | 9. Verificar el cumplimiento de las métricas definidas.<br>10. Analizar los resultados arrojados.<br>11. Realizar prototipo de interfaz de usuario<br>12. Refinar documento de casos de uso del sistema. |
| Técnica a utilizar   | Prototipo.                                                                                                                                                                                               |

**Tabla 21 Cuarta tarea de la fase III.**

### **Conclusiones**

En el presente capítulo se describió el procedimiento propuesto para la Validación de requisitos de software. Fueron definidas tres fases para guiar el proceso, un total de catorce tareas para darle cumplimiento a los objetivos del procedimiento, tres productos de entradas y seis productos de salida, generados en cada una de las fases planteadas. Se derivan de lo antes planteado un procedimiento para la Validación de requisitos de software.

---

## Capítulo 3: Aplicación del procedimiento. Análisis de resultados.

### Introducción

En este capítulo se realiza la aplicación del procedimiento propuesto en el capítulo anterior, al módulo generador de reportes del proyecto MENPET perteneciente al Centro de Tecnologías de Almacenamiento y Análisis de Datos de la UCI. Se aborda lo referente al análisis de los resultados, donde se hace un resumen de los principales aportes y ventajas que proporciona la aplicación del procedimiento antes definido en cualquiera de los proyectos de desarrollo de software que utilicen metodología RUP de la Universidad de las Ciencias Informáticas.

### 3.1 Planificación del procedimiento.

| No | Nombre                                                                           | Responsable                            | Tiempo ejecución | Margen de riesgo |
|----|----------------------------------------------------------------------------------|----------------------------------------|------------------|------------------|
| 1  | Recopilar los artefactos definidos a los cuales se les aplicará el procedimiento | Asegurador de la calidad               | 24 Horas.        | 12 Horas.        |
| 2  | Elaborar listado de requisitos de software                                       | Asegurador de la calidad               | 12 Horas.        | 6 Horas          |
| 3  | Aplicar la fase I del procedimiento.                                             | Asegurador de la calidad.<br>Analista. | 24 Horas         | 12 Horas         |
| 4  | Corregir los errores detectados en la                                            | Analista.<br>Funcionario.              | 48 Horas         | 24 Horas         |

### CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.

|           | fase I.                                         |                                        |                    |          |
|-----------|-------------------------------------------------|----------------------------------------|--------------------|----------|
| 5         | Aplicar la fase II del procedimiento.           | Asegurador de la calidad.<br>Analista. | 48 Horas           | 12 Horas |
| 6         | Corregir los errores detectados en la fase II.  | Analista.<br>Funcionario.              | 48 Horas           | 12 Horas |
| 7         | Aplicar la fase III del procedimiento.          | Asegurador de la calidad.<br>Analista. | 72 Horas           | 48 Horas |
| 8         | Corregir los errores detectados en la fase III. | Analista.<br>Funcionario.              | 48 Horas           | 24 Horas |
| Tota<br>I |                                                 |                                        | 324 Horas<br>Días. |          |

**Tabla 22 Planificación del procedimiento.**

#### **3.2 Aplicación de la fase I. Resultados.**

Durante la captura de requisitos se generan diferentes productos que complementan la ERS, entre ellos se encuentran el listado de requisitos de software; primer documento, al cual según el procedimiento propuesto, se le aplicará la validación.

Luego de un análisis detallado por parte del asegurador de la calidad y su equipo de trabajo del documento de requisitos, se detectó en una primera iteración que el 85% de los requisitos se encontraban especificados por escritos, en la segunda iteración el 90% contaban con esa característica, y por último el 100% de los requisitos, lograban encontrarse en un documento real, tangible, tanto para los funcionarios como para los desarrolladores.

Otros de los errores detectados durante esta fase del procedimiento es que en la primera iteración solo el 12.98% de los requisitos son posibles de probar; un requisito es probable si existe una serie de pasos finitos con los cuales verificar que el producto satisface el software, en la segunda iteración el 61% cumplen esta característica y finalmente se logra que el 99% sean posibles de probar y verificar. Continuando con la aplicación del PVRS se detectó que solamente el 7.02 % de los requisitos contemplados en el documento son abstractos y concisos en la primera iteración, en la segunda iteración solo el 45% permitía abstraerse y ser lo más conciso posible, y en la última iteración ya el 98% eran capaces de brindar la mayor cantidad de información que permitía abstraerse lo más posible de lo que puede ser el futuro sistema con la menor cantidad de palabras.

Como ejemplo de un requisito abstracto y conciso tenemos:

RF # 2.5. Modelos semánticos: Un requisito que no permite hacer una abstracción detallada de cómo sería el futuro sistema.

Otra característica que deberían cumplir los requisitos es que deben describirse como una característica del sistema, esto se sustenta fundamentalmente en que los requisitos deben decir qué es lo que el sistema debe hacer y no cómo lo debe hacer. Inicialmente el 23% se encontraba descrito como una característica del sistema, así como en la 2da y tercera iteración cumplían esta característica en un 40% y 100% respectivamente.

Ejemplo de ello

RF # 1. Tipos de Reportes (Enlace y Buscador). Nos dice que debemos de hacer y cómo debe ser el tipo de reporte.

Según el procedimiento propuesto en este trabajo, todos los errores detectados se encuentran en un documento de No-Conformidades el cual le es enviado a los analistas, estos a su vez lo revisan y aplicando nuevamente la técnica de Revisiones se reúnen con los clientes y desarrolladores para corregir los mismos.

Como parte de las actividades a seguir en el PVRS, el asegurador de la calidad envía a los analistas el listado de no conformidades, estos a su vez revisan dicho documento y teniendo en cuenta las necesidades de los clientes, refinan el listado de requisitos,

eliminando de esta manera los errores detectados, generando a su vez como producto de salida, el documento que contiene el listado de requisitos validados.

Ejemplo de requisitos refinados donde se ha eliminado la ambigüedad.

Requisito inicial: RF # 2.4. Explorando reportes

Requisito corregido: RF# 2.4. Mostrar definición de reportes.

Fue necesario realizar un total de 3 iteraciones del procedimiento para que de esta forma quedara totalmente validado el listado de requisitos. Dichos resultados se muestran a continuación en la siguiente figura.

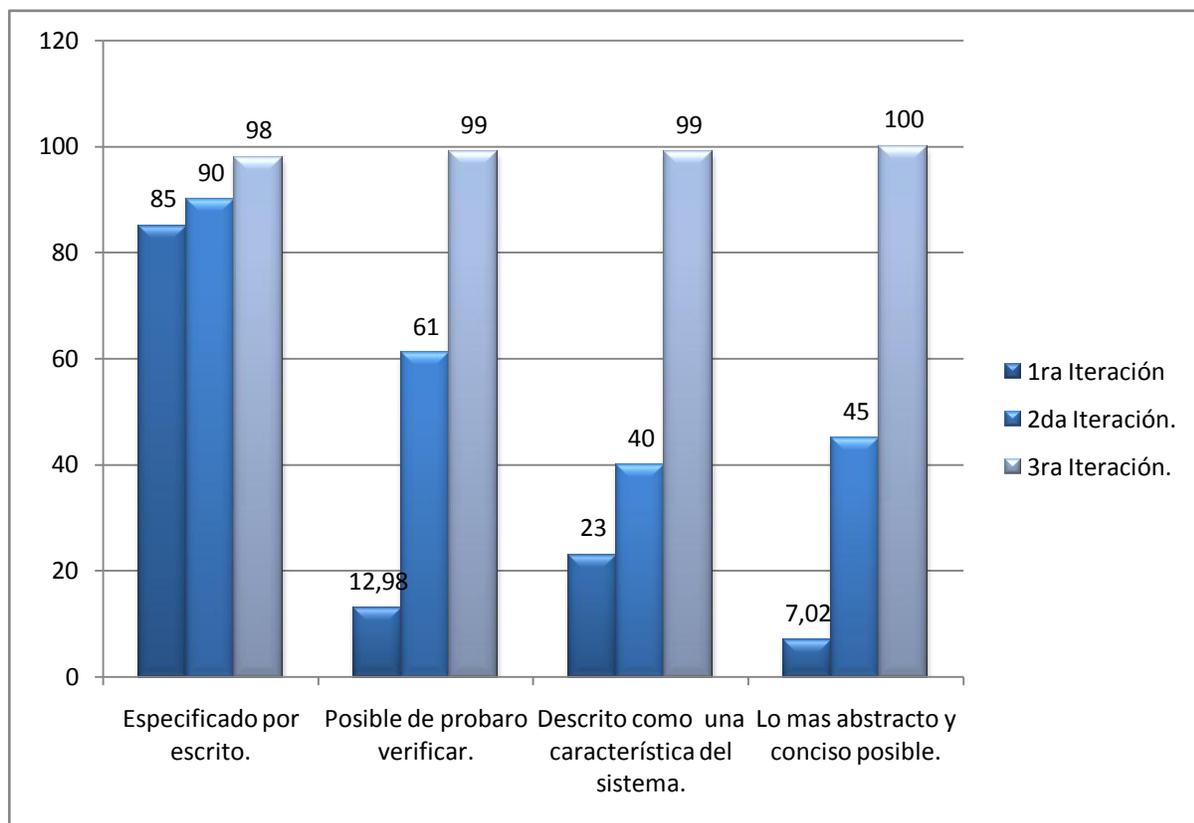


Ilustración 10 Resultados de la fase I.

### 3.3 Aplicación de la fase II. Resultados.

### CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.

Para darle continuidad a la aplicación de PVRS en el módulo generador de reportes del proyecto MENPET, es analizado el documento de ERS para el cual se tiene como entradas los documentos de salida generados en la fase I.

Ejemplo de la aplicación de las listas de chequeo al documento de ERS.

| Elementos definidos por la metodología RUP.                                                                        |      |      |                                 |                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------|------|------|---------------------------------|----------------------------------------------------------------------------------------------------|
| Indicadores a Evaluar                                                                                              | Eval | (NP) | Cantidad de elementos afectados | Comentarios                                                                                        |
| ¿Están todos los requisitos redactados de forma simple y clara para aquellos que vayan a consultarlo en un futuro? | 0    |      |                                 |                                                                                                    |
| ¿Debería especificarse algún requisito con más detalle?                                                            | 1    |      | 57                              | Los requisitos deberían ser aun más especificados con el objetivo de poder entenderlos claramente. |
| ¿Debería especificarse algún requisito con menos detalles?                                                         | 0    |      |                                 |                                                                                                    |

**CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.**

|                                                                                                                   |   |  |   |  |
|-------------------------------------------------------------------------------------------------------------------|---|--|---|--|
| ¿Todos los requisitos identificados se centran en lo que el sistema debe hacer y no como el sistema debe hacerlo? | 1 |  | 1 |  |
| ¿Han sido abordadas e identificadas los valores de entradas y salidas?                                            | 0 |  |   |  |
| ¿Han sido incluidos las respuestas válidas y no válidas de los valores de entrada?                                | 0 |  |   |  |

**Tabla 23 Elementos definidos en la lista de chequeo por la metodología RUP.**

| Semántica del documento                                            |      |      |                                                        |             |
|--------------------------------------------------------------------|------|------|--------------------------------------------------------|-------------|
| Indicadores a Evaluar                                              | Eval | (NP) | Cantidad de elementos afectados                        | Comentarios |
| ¿Ha identificado errores ortográficos?                             | 0    |      |                                                        |             |
| ¿Se entiende claramente lo que se ha especificado en el documento? | 0    |      | Hay especificaciones que tiene varias interpretaciones |             |

**CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.**

|                                                                                                                              |   |  |  |  |
|------------------------------------------------------------------------------------------------------------------------------|---|--|--|--|
|                                                                                                                              |   |  |  |  |
| ¿El número de página que aparece en el índice coincide con el contenido que se refleja realmente en dicha página?            | 1 |  |  |  |
| ¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento? | 1 |  |  |  |

**Tabla 24 Semántica del documento.**

Luego de analizado todo el documento de ERS y formulado las preguntas correspondientes a las listas de chequeo, es conformado el documento de no conformidades, el cual es verificado por los analistas y estos posteriormente proceden al refinamiento del documento de ERS.

En el documento de ERS se detectaron que el 86% de un total de 55 requisitos, presentan problemas de ambigüedad, el 3% no son correctos, así como el 22% no son completos, y además el 73%, el 87% y el 82 % no son modificables, verificables y concisos respectivamente.

Todos estos resultados fueron posibles luego de aplicada la técnica de Validación de requisitos (Listas de chequeo) correspondiente a esta fase del PVRS.

Para lograr que el documento de ERS cumpliera con todas las características requeridas en el PVRS se vio la necesidad de iterar en 2 ocasiones el procedimiento, obteniéndose como resultado los datos siguientes.

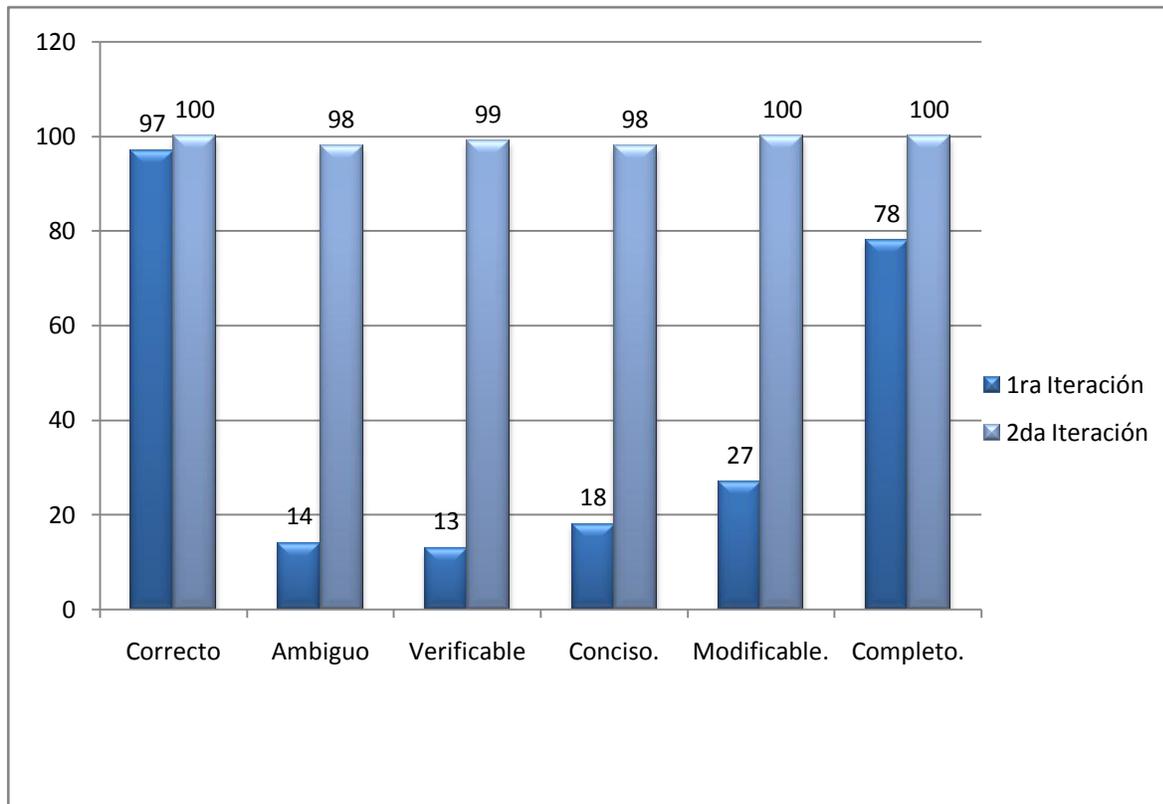


Ilustración 11 Resultados de la fase II.

### 3.4 Aplicación de la fase III. Resultados.

El documento de casos de uso del sistema se validó teniendo en cuenta las actividades propuestas en el PVRS. Primeramente se realizó por parte del asegurador de la calidad una revisión minuciosa de dicho documento, posteriormente se construyó una matriz para verificar si cada requisito estaba contenido en al menos un CU.

Para lograr que todos los requisitos se encontraran en al menos un caso de uso fue necesario aplicar la técnica de Validación de requisitos llamada matriz de trazabilidad, y además fue necesario una revisión por parte de los analistas de los documentos de requisitos validados anteriormente para consolidar cuales eran realmente las necesidades planteadas por los clientes y de esta manera obtener una matriz de trazabilidad donde todos los requisitos de software se encuentran reflejados en al menos un caso de uso.

**CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.**

A continuación se presenta la matriz de trazabilidad de los requisitos de software, pertenecientes al módulo generador de reportes del proyecto MENPET.

| RF    | CU1 | CU2 | CU3 | CU4 | CU5 | CU6 | CU7 | CU8 | CU9 | CU10 | CU11 | CU12 | CU13 | CU14 | CU15 | CU16 | CU17 | CU18 | CU19 | CU20 | CU21 | CU22 |   |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|------|------|------|------|------|---|
| RF_1  |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      | x    |      |      |      |   |
| RF_2  |     |     |     |     |     |     |     |     |     | x    |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_3  |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_4  |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_5  |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_6  |     |     |     |     |     |     |     |     |     |      |      |      |      |      | x    |      |      |      |      |      |      |      |   |
| RF_7  |     |     |     |     |     |     |     |     |     |      |      |      |      |      | x    |      |      |      |      |      |      |      |   |
| RF_8  |     |     |     |     |     |     |     |     |     |      |      |      |      |      | x    |      |      |      |      |      |      |      |   |
| RF_9  |     |     |     |     |     |     |     |     |     |      |      |      |      |      | x    |      |      |      |      |      |      |      |   |
| RF_10 |     |     |     |     |     |     |     |     |     |      |      |      |      |      | x    |      |      |      |      |      |      |      |   |
| RF_11 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_12 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_13 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      | x    |      |      |      |   |
| RF_14 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_15 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_16 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_17 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_18 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_19 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_20 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_21 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_22 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_23 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_24 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_25 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_26 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_27 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_28 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_29 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_30 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_31 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      | x    |   |
| RF_32 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      | x |
| RF_33 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      | x    |      |      |   |
| RF_34 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_35 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_36 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      |   |
| RF_37 |     |     |     |     |     |     |     |     |     |      |      |      |      |      |      |      |      |      |      |      |      |      | x |

**Ilustración 12 Matriz de trazabilidad.**

Como se puede apreciar en la matriz de trazabilidad anteriormente expuesta, se demuestra que el 64.86% de los requisitos no están incluidos en al menos un caso de uso y solamente el 35.14 %de los requisitos están reflejados en los casos de uso. Producto a esto los

### CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.

requisitos en su totalidad no se ven reflejados en los casos de uso, lo que provocaría que no todas las funcionalidades del software queden totalmente desarrolladas, por lo que se hizo necesario la creación de un documento de no conformidades el cual se le hizo llegar a los analistas, estos a su vez lo analizaron y refinaron de modo que se incluyeran los requisitos en los casos de uso.

Una vez resuelto los problemas de inclusión de los requisitos en los casos de uso, se procedió a darle continuidad a las actividades propuestas por el PVRS el cual plantea que en la siguiente etapa, los casos de uso deberán cumplir con un conjunto de factores los cuales son evaluados mediante métricas definidas por Calisoft, utilizadas para verificar completitud, correctitud, complejidad y consistencia de los requisitos de software.

A continuación se muestran ejemplos de las métricas que deben cumplir los requisitos de software del módulo generador de reportes del proyecto MENPET, al cual se le ha venido aplicando el procedimiento para validar requisitos de software propuesto por las autoras.

| Factores de Completitud                                                                                   | Métricas Asociadas                                                                                                                          | Valor |
|-----------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|-------|
| Factor 1. ¿Han sido involucradas todas las áreas funcionales relevantes a las cuales apoyará el sistema?  | Métrica 1: Número de áreas funcionales relevantes omitidas.<br>Umbral: < 10%<br>Acciones sugeridas: Revisar los requerimientos del negocio  | 0     |
| Factor 2. ¿Han sido involucradas todas las áreas funcionales secundarias a las cuales apoyará el sistema? | Métrica 2 : Número de áreas funcionales secundarias omitidas<br>Umbral: < 10%<br>Acciones sugeridas: Revisar los requerimientos del negocio | 0     |
| Factor 3. ¿Han sido                                                                                       | Métrica 3: Número de roles relevantes omitidos                                                                                              | 0     |

### CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.

|                                                                                                                               |                                                                                                                                                                                                                                                                         |    |
|-------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| definidos todos los roles relevantes de usuario encargados de generar/modificar o consultar información?                      | Umbral: < 10%<br>Acciones sugerida: Revisar el alcance del sistema e Involucrar tipos de usuarios representativos de cada una de las áreas funcionales.                                                                                                                 |    |
| Factor 4. ¿Han sido definidos todos los roles secundarios de usuario encargados de generar/modificar o consultar información? | Métrica 4 : Número de roles secundarios omitidos<br>Umbral: < 10%<br>Acciones sugerida: Revisar el alcance del sistema e Involucrar tipos de usuarios representativos de cada una de las áreas funcionales                                                              | 0  |
| Factor 5. ¿Han sido considerados todos los sistemas externos con los cuáles interactuará el sistema?                          | Métrica 5: Número de sistemas externos omitidos<br>Umbral: < 10% 2<br>Acción sugerida: Revisar los sistemas actuales en operación y analizar su interacción con el sistema a desarrollar                                                                                | 0  |
| Factor 6. ¿Se presenta una descripción resumida (descripción de alto nivel) de todos los casos de uso del negocio?            | Métrica 6: Número de casos de uso que no tiene descripción resumida<br>Umbral: < 10%<br>Acción sugerida: Completar la descripción resumida del caso de uso                                                                                                              | 2  |
| Factor 7. ¿Están definidos todos los requisitos que justifican la funcionalidad del caso de uso?                              | Métrica 7: Número de requisitos omitidos por caso de uso<br>Umbral < 10%<br>Métrica 8: Número de casos de uso que tienen requisitos omitidos<br>Umbral < 10% Acción sugerida: Revisar la lista de requisitos para determinar cuáles serán apoyados por cada caso de uso | 25 |

### CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.

|                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                                                   |    |
|---------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Factor 8. ¿Existen requisitos que no han sido considerados en algún caso de uso?                                          | Métrica 9: Número de requisitos que no son considerados en ningún caso de uso.                                                                                                                                                                                                                                                                                                    | 25 |
| Factor 9. ¿Han sido definidos todos los roles de usuario encargados de actividades de soporte/ mantenimiento / auditoría? | Métrica 10: Número de roles de soporte / mantenimiento / auditoria omitidos.<br>Umbral: < 20%<br>Acción sugerida: Analizar las funciones de soporte / mantenimiento / auditoria y definir los usuarios que interactuarán con el sistema                                                                                                                                           | 0  |
| Factor 10. ¿Se presenta una descripción detallada (descripción extendida esencial) de todos los casos de uso del negocio? | Métrica 11: Número de casos de uso que no poseen una descripción extendida.<br>Umbral < 20%<br>Acción sugerida: Interactuar con el usuario para realizar la definición extendida del caso de uso que sea consistente con la definición a alto nivel                                                                                                                               | 10 |
| Factor 11. ¿Están todas las acciones del flujo de eventos redactadas en función del responsable?                          | Métrica 12: Número de acciones del flujo de eventos que no están redactadas en función del responsable.<br>Umbral < 20%<br>Acción sugerida: Revisar las responsabilidades tanto del actor (actores) como del sistema<br>Métrica 13: Número de casos de uso que tienen acciones del flujo de eventos no redactados en función del responsable.<br>Umbral < 10%<br>Acción sugerida: | 0  |
| Factor 12. ¿Se describen las condiciones de                                                                               | Métrica 14: Número de casos de uso que no describen condiciones de excepción relevantes                                                                                                                                                                                                                                                                                           | 5  |

**CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.**

|                                                                                                                                     |                                                                                                                                                                                                            |   |
|-------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| excepción relevantes que debe contemplar cada flujo de eventos?                                                                     | Umbral < 20%<br>Acción sugerida: Revisar las excepciones presentadas en el flujo de eventos que producen un mensaje de error al usuario                                                                    |   |
| Factor 13. ¿Todos los casos de uso del negocio han sido clasificados de acuerdo a su relevancia (primario / secundario / opcional)? | Métrica 15: Número de casos de uso que no han sido clasificados<br>Umbral: < 10%<br>Acción sugerida: Hacer reuniones con los usuarios para analizar y priorizar los requisitos de acuerdo a su relevancia. | 0 |

**Tabla 25 Factor de completitud.**

| Factores de Consistencia                                                                                                                      | Métricas Asociadas                                                                                                                                                                                                                                   | Valor |
|-----------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| Factor 14. ¿El nombre dado a los casos de uso es una expresión verbal que describe alguna funcionalidad relevante en el contexto del usuario? | Métrica 16: Número de casos de uso que tienen un nombre incorrecto<br>Umbral < 20%<br>Acción sugerida: Modifique el nombre del caso de uso de tal manera que signifique una acción desde el punto de vista del usuario                               | 0     |
| Factor 15. ¿Representa el caso de uso una interacción observable por un actor?                                                                | Métrica 17: Número de casos de uso que no representan una interacción observable por un actor<br>Umbral < 5%<br>Acción sugerida: Elimine el caso de uso e incorpore su funcionalidad como una responsabilidad del sistema dentro de otro caso de uso | 0     |
| Factor 16. ¿No existe solapamiento en la                                                                                                      | Métrica 18: Número de casos de uso que se solapan                                                                                                                                                                                                    | 0     |

**CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.**

|                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                      |                   |
|--------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| <p>funcionalidad que representan los diferentes casos de uso?</p>                                            | <p>Umbral &lt; 10%</p> <p>Acción sugerida: Integrar en uno solo aquellos casos de uso que se solapan. Revisar las acciones repetidas y asignarlas a un solo caso de uso</p>                                                                                                                                                                                                                                                          |                   |
| <p>Factor 17. ¿Existen acciones en el flujo de eventos asignadas a un responsable que no le corresponde?</p> | <p>Métrica 19: Número de acciones del flujo de eventos que no se corresponde la definición de las con el responsable</p> <p>Umbral &lt; 20%</p> <p>Acción sugerida: Revisar las responsabilidades tanto del actor (actores) como del sistema</p> <p>Métrica 20: Número de casos de uso que tienen acciones del flujo de eventos asignados a un responsable que no le corresponde.</p> <p>Umbral &lt; 10%</p> <p>Acción sugerida:</p> | <p>0</p> <p>0</p> |
| <p>Factor 18. ¿Está adecuadamente redactado (en el lenguaje del usuario) el flujo de eventos?</p>            | <p>Métrica 21: Grado de adecuación de la descripción del flujo de eventos para un caso de uso</p> <p>Acción sugerida: Revise la descripción para que sea definida en el lenguaje del usuario. Asegúrese de definir el responsable de la acción. Establezca claramente las acciones de</p>                                                                                                                                            | <p>0</p>          |

**CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.**

|                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                      |   |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
|                                                                                                                                                                                             | <p>inicio y fin del caso de uso</p> <p>Métrica 22: Número de casos de uso no aceptados</p> <p>Umbral &lt; 20%</p> <p>Acción sugerida: Realice reuniones con el usuario para mejorar la descripción de los casos de uso</p>                                                                                                                                                                           | 0 |
| Factor 19. ¿La descripción del flujo de eventos se inicia con la descripción de una acción externa originada por un actor o por una condición interna del sistema claramente identificable? | <p>Métrica 23: Número de casos de uso cuya descripción extendida no inicia con una acción externa o con una condición monitoreada por el sistema</p> <p>Umbral: &lt; 10%</p> <p>Acción sugerida: Complete la definición del caso de uso incluyendo la acción fuera del sistema que da inicio al caso de uso o la condición interna que el sistema tiene controlar para dar inicio al caso de uso</p> | 0 |
| Factor 20. Si en el caso de uso interviene más de un actor, ¿existe claridad en cuál de ellos es el actor iniciador?                                                                        | <p>Métrica 24: Número de casos de uso con más de un actor que no describe cuál es el actor iniciador</p> <p>Umbral: &lt; 20%</p> <p>Acción sugerida: Revise los puntos de inicio del caso de uso y asigne el actor que inicia la acción</p>                                                                                                                                                          | 0 |
| Factor 21. ¿Existe una adecuada separación entre el flujo básico de eventos y                                                                                                               | <p>Métrica 25: Número de casos de uso complejos que no tienen separación del flujo básico y de flujos alternos</p>                                                                                                                                                                                                                                                                                   | 0 |

**CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.**

|                                                     |                                                                                                                                                                                                                                                                                                                      |  |
|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <p>los flujos alternos y/o flujos subordinados?</p> | <p>Umbral: &lt; 20%</p> <p>Acción sugerida: Estructure el caso de uso de manera que separe su funcionalidad básica (caso de uso base) de la funcionalidad repetitiva o alternativa. Si hay pasos repetitivos forme un caso de uso que lo incluye y los pasos alternativos formen un caso de uso que lo extienda.</p> |  |
|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

**Tabla 26 Factor de consistencia.**

| Factores de Correctitud                                                                             | Métricas Asociadas                                                                                                                                                                                                                                                                                                                                                                                                           | valor      |
|-----------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| <p>Factor 22.<br/>¿Existe para cada caso de uso de negocio por lo menos un usuario responsable?</p> | <p>Métrica 26: Número de casos de uso que no tienen un usuario responsable</p> <p>Umbral: &lt; 10%</p> <p>Acción sugerida: Analice la responsabilidad que representa el caso de uso y acuerde con los usuarios cuál es el responsable directo de éste</p>                                                                                                                                                                    | <p>0</p>   |
| <p>Factor 23.<br/>¿Representa el caso de uso requisitos comprensibles por el usuario?</p>           | <p>Métrica 27: Grado en que los requisitos representados por el caso de uso son comprensibles por el usuario</p> <p>Métrica 28: Número de casos de uso en que los requisitos representados no son comprensibles por el usuario</p> <p>Umbral: &lt; 5%</p> <p>Acción sugerida: Discuta con el usuario la interacción que describe el caso de uso y ajuste dicha descripción de manera que sea comprensible por el usuario</p> | <p>20%</p> |

**CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.**

|                                                                                                                            |                                                                                                                                                                                                                                                                                |              |
|----------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| <p>Factor 24.<br/>¿Se ajusta la representación del diagrama del caso de uso de acuerdo a lo normado en la metodología?</p> | <p>Métrica 29: Grado en que se ajusta el diagrama del caso de uso a la metodología.<br/>Umbral:<br/>Acciones Sugeridas:</p>                                                                                                                                                    | <p>100%</p>  |
| <p>Factor 25.<br/>¿Las interacciones definidas describen la funcionalidad requerida del sistema?</p>                       | <p>Métrica 30: Grado en que las interacciones definidas describen la funcionalidad solicitada por el usuario<br/>Métrica 31: Número de casos de uso que deben ser modificados para adecuarlos a la funcionalidad del sistema<br/>Umbral: &lt; 10%<br/>Acción sugerida:</p>     | <p>todas</p> |
| <p>Factor 26.<br/>¿Las interacciones definidas introducen mejoras al proceso actual?</p>                                   | <p>Métrica 32: Número de casos de uso que deben ser modificados para mejorar el proceso actual<br/>Umbral: &lt; 20%<br/>Acción sugerida: Analice la situación descrita en la interacción y estudie la manera de mejorar el proceso con el uso de la tecnología informática</p> | <p>22</p>    |
| <p>Factor 27.<br/>¿Se ajusta la representación del diagrama</p>                                                            | <p>Métrica 33: Grado en que se ajusta el diagrama del caso de uso a la metodología.<br/>Umbral:<br/>Acciones Sugeridas:</p>                                                                                                                                                    |              |

**CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.**

|                                                            |  |  |
|------------------------------------------------------------|--|--|
| del caso de uso de acuerdo a lo normado en la metodología? |  |  |
|------------------------------------------------------------|--|--|

**Tabla 27 Factor de correctitud.**

| Factores de Complejidad                                                                                               | Métricas Asociadas                                                                                                                                                                                                                             | valor                                           |
|-----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------|
| Factor 28. ¿En sistemas relativamente grandes se ha realizado una agrupación de los casos de uso en paquetes?         | Métrica 34: Se hizo partición por paquetes<br>Umbral:<br>Acción sugerida:<br>Métrica 35: Grado en que es adecuada la partición por paquetes<br>Umbral:<br>Acción sugerida:                                                                     | No<br><br><br><br><br><br><br><br><br><br><br>0 |
| Factor 29. ¿Los elementos dentro del diagrama están adecuadamente ubicados de manera que facilitan su interpretación? | Métrica 36: Número de elementos del diagrama que requieren reubicación<br>Umbral: < 30%<br>Acción sugerida: Modifique la ubicación de los elementos del diagrama de manera que los elementos relacionados se encuentren lo más cercano posible |                                                 |

**Tabla 28 Factor de complejidad.**

En cuanto a la completitud, al aplicar las métricas, quedó demostrado que fueron involucradas todas las áreas relevantes y secundarias que apoyarán al sistema, se definieron todos los roles relevante de usuario pero dos casos de uso no tenían descripción resumida y un total de 25 requisitos omitidos por caso de uso. En cuanto a la consistencia quedó demostrado que todos los casos de unos tenían un nombre correcto y que representaban una interacción observable para el actor. En cuanto a la métrica de correctitud se observó que solo el 20% de los requisitos eran comprensibles para el usuario y que debían ser modificados para mejorar el proceso actual. Estas métricas permitieron al asegurador de la calidad identificar un grupo de inconsistencias, estas a su vez fueron recogidas en un documento de no conformidades que posteriormente le permitió a los analistas corregir los errores detectados.

Para validar las funcionalidades del sistema fueron utilizados los prototipos de interfaz de usuario, los cuales permitieron tanto a los clientes como desarrolladores a validar la autenticidad de los requisitos levantados. A continuación se muestran algunos de los prototipos que fueron utilizados.

Ejemplo #1 RF # 6 Exportar la información a pdf, el cual se ve reflejado en el caso de uso #15, Exportar reporte, y se muestra en su flujo normal de eventos en el siguiente prototipo de interfaz de usuario.

## CAPÍTULO 3. APLICACIÓN DEL PROCEDIMIENTO. ANÁLISIS DE RESULTADO.

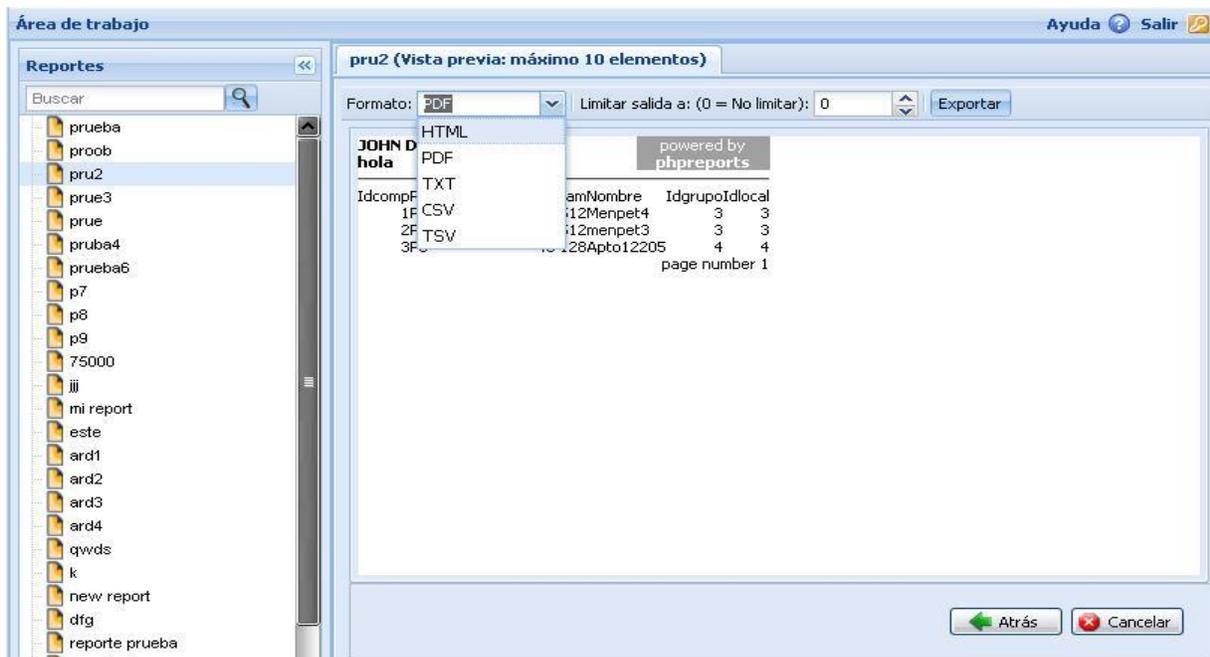


Ilustración 13 Exportar información.

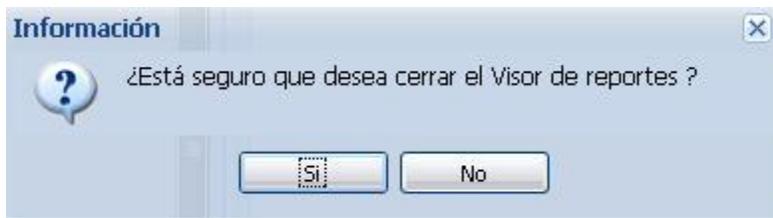


Ilustración 14 Confirmación de la exploración.

Ejemplo # 2 RF #33, Buscar Modelo, el cual se ve reflejado en el caso de uso # 20, Buscar Modelo, y se visualiza en su flujo normal de eventos el siguiente prototipo.



Ilustración 15 Buscar modelo.

Mientras en el flujo alterno se visualiza de la siguiente manera.

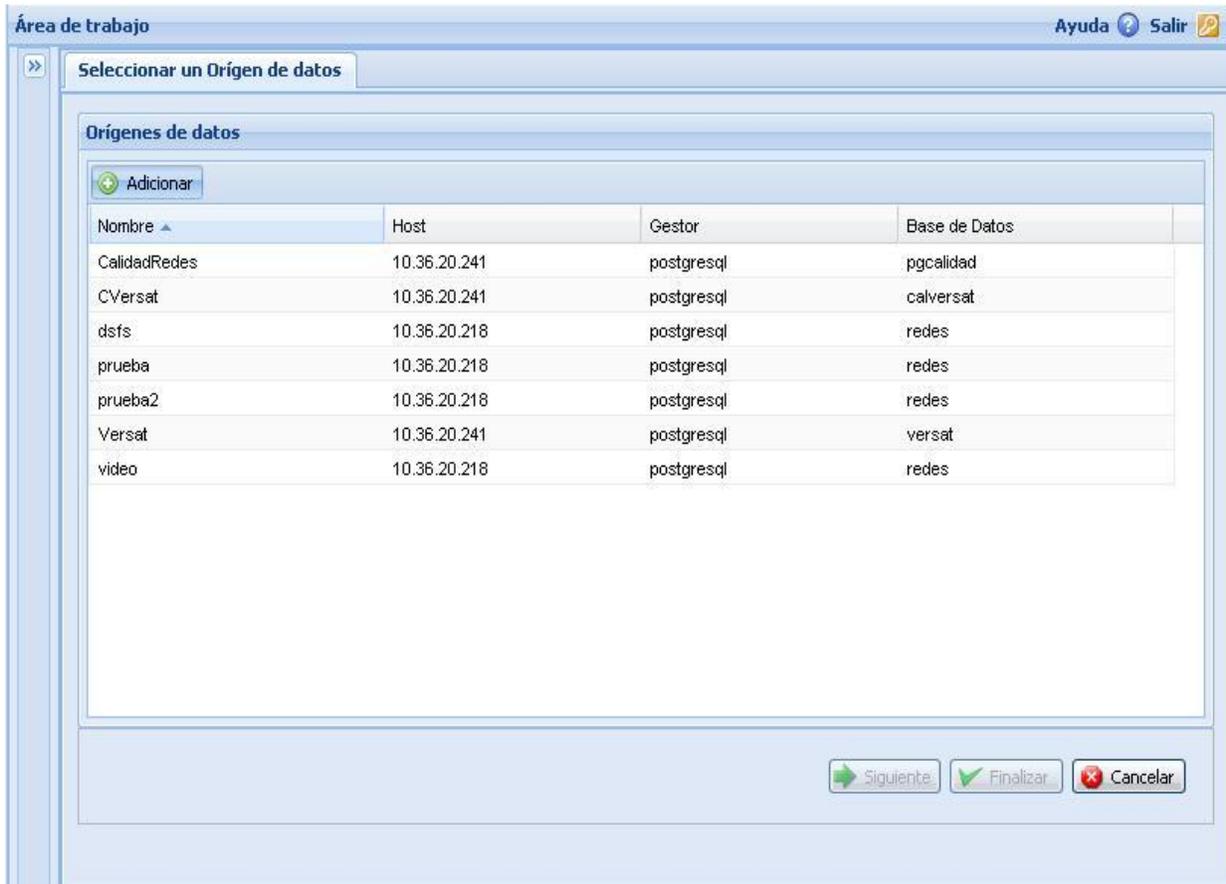


Ilustración 16 Buscar modelo.

Ejemplo # 3: RF# 31 Buscar Reporte, relacionado con el caso de uso #21 Buscar Reporte y con el siguiente prototipo.



**Ilustración 17** Buscar reportes.

### **Conclusiones**

En este capítulo se aplica el Procedimiento para validar requisitos de software al módulo Generador de reportes del proyecto MENPET y se analizaron los resultados haciéndose necesario realizar varias iteraciones para validar el listado de requisitos, el documento de especificación de requisitos y el modelo de casos del sistema, quedando finalmente validados y con calidad los requisitos de software que se le entregan a los clientes. Se demuestran además como con la aplicación del procedimiento propuesto se logran detectar un conjunto de errores los que pueden llegar a ser muy costosos durante todo el desarrollo de software del futuro sistema, además se obtuvieron resultados que demuestran la efectividad de la propuesta.

# RECOMENDACIONES

---

## Conclusiones

Como resultado de la investigación se arribaron a las siguientes conclusiones.

- Se realizó el marco teórico de la investigación, incluyendo además el marco conceptual, analizándose dentro del mismo temáticas como el estudio de la Ingeniería de requisitos, el cual arrojó como resultados que la Validación de requisitos de software contribuye al control, seguimiento y mejora de la calidad del proceso de desarrollo de software.
- Se demostró que los problemas de Validación de requisitos de software existentes en la UCI y a nivel mundial, y las tendencias del auge en la producción de software en el país, propició la necesidad de crear un procedimiento para la Validación de requisitos de software.
- El procedimiento desarrollado es capaz de mejorar la calidad de los requisitos que le son entregados al cliente, siendo preciso en la propuesta definir un conjunto de actividades, responsables y productos de entada y salida que potenciaron un mejor entendimiento del procedimiento.
- Se validó el procedimiento aplicándolo al módulo generador de reportes del proyecto MENPET perteneciente al Centro de Tecnologías de Almacenamiento y Análisis de Datos de la universidad comprobándose que la mayoría de los requisitos, especificaciones y casos de uso del sistema, presentan problemas de ambigüedad, inconsistencia, verificabilidad, correctitud entre otras características importantes y definidas en el procedimiento.
- Se cumplió el objetivo general trazado en este trabajo de diploma: Desarrollar un procedimiento para la validación de requisitos de software que mejorara la calidad de los mismos.

# RECOMENDACIONES

---

## Recomendaciones

- Aplicar el procedimiento propuesto a todos los proyectos que utilicen metodología RUP.
- Refinar el procedimiento teniendo en cuenta que se posible de aplicar a cualquier proyecto independientemente de la metodología de desarrollo que este utilice.
- Permitir que cada proyecto que utilice este procedimiento cree su propia lista de chequeo, en dependencia de las características propias de cada proyecto.
- Impartir un curso de Validación de requisitos. Tomándose este trabajo de diploma como guía de aprendizaje.

---

## Bibliografía

1. **Booch, G., Rumbaugh, J., & Jacobson.** *IEEE Software Requirement Engineering.* Second Edition, 1997.
2. **Caro, Bárbara A. Mcdonald Landazuri Edmundo Tovar.** *Definición de Perfiles en Herramientas de Gestión de Requisitos.*
3. **Chaves, Michael Arias.** *La Ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software.* 1997. S/N.
4. **Durán, A and Bernárdez, B.** *Metodología para la Elicitación de Requisitos de Sistemas Software.* Universidad de Sevilla : s.n., 2000. LSI-2000-10.
5. David. 1995.
6. **Escalona, María José & Koch, Nora.** *Ingeniería de Requisitos en Aplicaciones para la Web, un estudio comparativo.* Sevilla, España : s.n., 2002.
7. **FONT, Y. C. y VÁZQUEZ, M. C.** *Gestión de requisitos en el desarrollo del software, Tesis de Grado, Universidad de las Ciencias Informáticas.* La Habana : s.n., 2007.
8. **Gabriela N. Aranda, Aurora Vizcaíno, Alejandra Cechich.** *Mejora del Proceso de Elicitación de Requisitos en Proyectos GSD.*
9. **Garzón, Darwin Jiménez.** *Ingeniería de Software II.*
10. **González, Guillermo.** *Ingeniería de Requisitos.* 1995.
11. **Grady Booch, James Rumbaugh e Ivar Jacobson.** *El Lenguaje Unificado de Modelado.* 8478290761.
12. **IEEE, S.** *Calidad de Software.* Marzo 2006.
13. **Jacobson, Ivar, Booch, Grady and Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* 2000. Vols. Capítulos 7, 8 páginas 125-163, 187-202.
14. **JACOBSON, I.** *El Proceso unificado de desarrollo de software.* 2004.
15. **Leishman, T.** *Requirements risks can drown software projects.* 2002.

- 
16. **Loucopoulos, P and Karakostas.** *System Requirements Engineering McGraw-Hi.* 1995.
  17. —. *System Requirements Engineering McGraw-Hill.* 1995.
  18. **M. Griselda Báez, Silvia I. Barba Brunner.** *Metodología DoRCU para la Ingeniería de Requisitos.* 2001.
  19. **M.J.Escalona, N.Koch.** *Ingeniería de requisitos en aplicaciones para la web. Un estudio comparativo.*
  20. **Mitecnologico.** Proceso de Ingeniería de requisitos. [Online] [Cited: 12 5, 2008.] [www.mitecnologico.com/Main/ProcesosDeLaIngenieriaDeRequerimientos](http://www.mitecnologico.com/Main/ProcesosDeLaIngenieriaDeRequerimientos).
  21. **Pressman, R.** *Ingeniería de software: Un enfoque práctico.* La Habana : Editorial Félix Varela, 2005.
  22. **Pressman, Roger.** *Ingeniería del Software Un enfoque práctico.*
  23. **PRESSMAN, R. S.** *Ingeniería del Software. Un enfoque práctico.* Madrid : s.n., 2005. 97-010-5473-3.
  24. **Ralph, R. Young.** *The Requirements Engineering Handbook.* s.l. : Artech House, 2004.
  25. **RUP.** *Ayuda extendida de RUP.*
  26. **SÁNCHEZ, M. A. M.** *Metodologías De Desarrollo De Software.* 2004.
  27. **Somerville, Ian.** *Ingeniería del Software.* Sexta Edición.
  28. **Somerville I, Addison-Wesley.** *Software Engineering.* 1995.
  29. **Sommerville, I. and Sawyer P.** *Requirements' Engineering: A Good Practice Guide.* 1997. S/N.
  30. **Sommerville, Ian.** *Ingeniería de software.* Séptima Edición 2008. N/S.
  31. **Torres, José Luis.** *Especificación de requisitos en Ingeniería de requisitos.*
  32. **Wieggers,** 2006), citando a (Somerville 1997).

---

## Glosario de Términos.

**ANSI:** American National Standards Institute. Se trata del organismo estandarizado norteamericano, pero sus decisiones y normas de estandarización tienen un importante peso específico sobre la industria informática mundial. Incluye el IM (Institute of Electrical and Electronics Engineers) y la VA (Electronic Industries Association)

**BPMN:** Business Process Modeling Notation. Es un estándar de modelado de procesos de negocio, en donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes.

**Calisoft:** Departamento de calidad de software de la Universidad de las Ciencias Informáticas.

**Checklist:** lista de chequeo que posee un listado de preguntas, en forma de cuestionario que sirve para verificar el grado de cumplimiento de determinadas reglas establecidas a priori con un fin determinado.

**CU:** Caso de uso. Proporciona una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas.

**DNC:** Documento de no conformidades. Documento en el que se encuentran las no conformidades o errores encontrados.

**ERS:** Especificación de requisitos de software. Documento que contiene y describe los requisitos del sistema.

**IEEE:** Institute of Electrical and Electronics Engineers. Importante asociación de técnicos y profesionales, con sede en los Estados Unidos. Fue fundada en 1884 y en 1998 tenía aproximadamente 320.000 miembros en 147 países. Favorece la investigación en campos diversos, como la tecnología aeroespacial, la computación, las comunicaciones y la tecnología biomédica. Promueve la estandarización de normas.

---

**IR:** Ingeniería de Requisitos. Conjunto de actividades en las cuales, utilizando técnicas y herramientas, se analiza un problema y se concluye con la especificación de una solución.

**ISO:** Organización Internacional para la Normalización. Es la entidad responsable para la normalización a escala mundial con una agrupación hasta la fecha de 91 países. Está formado por distintos comités técnicos, cada uno de los cuales es responsable de la normalización para cada área de especialidad. El propósito de ISO es promover el desarrollo de la normalización para fomentar a nivel internacional el intercambio de bienes y servicios y para el desarrollo de la cooperación en actividades económicas, intelectuales, científicas y tecnológicas. El resultado del trabajo técnico dentro de ISO se publica en forma final como normas internacionales.

**MCUS:** Modelo de casos de uso del sistema. Documento que contiene los casos de uso del sistema y sus especificaciones.

**MENPET:** Ministerio de Energía y el Petróleo de la República Bolivariana de Venezuela. Proyecto productivo desarrollado en la universidad de las ciencias informática.

**Procedimiento:** es el modo de ejecutar determinadas acciones que suelen realizarse de la misma forma, con una serie común de pasos claramente definidos, que permiten realizar una ocupación o trabajo correctamente.

**RF:** Requisitos funcionales. Alguna actividad o servicio que debe realizar el software y que puede ser comprobada.

**RNF:** Requisitos no funcionales. Limitación de hardware o software bajo la cual el sistema debe operar.

**RUP:** El Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software. Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

**Stakeholders:** Implicados con el sistema; dígase usuarios, beneficiados del sistema o producto final, desarrolladores y los encargados de mantener el Software posteriormente.

---

**TIC:** Tecnologías de la información y la comunicación. Son un conjunto de servicios, redes, software y dispositivos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario. Las Tecnologías de la información y la comunicación, son un solo concepto en dos vertientes diferentes como principal premisa de estudio en las ciencias sociales donde tales tecnologías afectan la forma de vivir de las sociedades. Su uso y abuso exhaustivo para denotar modernidad ha llevado a visiones totalmente erróneas del origen del término. La ciencia informática se encarga del estudio, desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de hardware y software como recursos de los sistemas informáticos. Más de lo anterior no se encargan las tecnologías como tal.

**UdeD:** Universo de Discurso. Es el contexto general en el cual el software deberá ser desarrollado y deberá operar. Incluye todas las fuentes de información y todas las personas relacionadas con el software. Esas personas son conocidas también como actores de ese universo. El UdeD es la realidad circunstanciada por el conjunto de objetivos definidos por quienes demandaron el software. A partir de la extracción y análisis de información en su ámbito se obtienen todas las especificaciones necesarias y tipos de requisitos para el futuro producto software.

**UML:** Lenguaje para modelamiento unificado. Es un lenguaje para la especificación, visualización, construcción y documentación de los artefactos de un proceso de sistema intensivo. Fue originalmente concebido por la Corporación Rational Software y tres de los más prominentes metodólogos en la industria de la tecnología y sistemas de información: Grady Booch, James Rumbaugh, e Ivar Jacobson ("The Three Amigos"). El lenguaje ha ganado un significativo soporte de la industria de varias organizaciones vía el consorcio de socios de UML y ha sido presentado al Object Management Group (OMG) y aprobado por éste como un estándar

---

## Anexos

Anexo 1. Tabla publicada por Standish Group.

| Culminación Proyectos | 1994 | 1996 | 1998 | 2000 | 2002 | 2004 |
|-----------------------|------|------|------|------|------|------|
| Exitosos              | 16%  | 27%  | 26%  | 28%  | 15%  | 29%  |
| Comprometidos         | 53%  | 33%  | 46%  | 49%  | 51%  | 53%  |
| Fallaron              | 31%  | 40%  | 28%  | 23%  | 34%  | 18%  |
| No exitosos           | 84%  | 73%  | 74%  | 72%  | 84%  | 71%  |

Anexo 1 Fuentes Standish Group Inc.

Lista de Chequeo Especificación de Requisitos

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

[Documento para la confección de Listas de Chequeo]

Introducción

Esta lista de chequeo cuenta con varios puntos, los cuales serán clasificados antes de ser aplicados, teniendo en cuenta su pertinencia, además de tener un punto donde se

---

---

referencia las observaciones en cuanto al aspecto a evaluar.

#### Propósito y objetivos

El objetivo general de la lista de chequeo es evaluar las especificaciones del [nombre del artefacto] de los proyectos de la universidad.

Esta plantilla ha sido confeccionada para guiar a desarrolladores, especialistas o expertos técnicos en la verificación y evaluación de las especificaciones del [nombre del artefacto].

Los aspectos definidos en esta lista de chequeo podrán ser referenciados en otras actividades de chequeo, en dependencia de lo que se necesite verificar.

Esta plantilla permitirá recoger los puntos eficientes y los ineficientes que tienen los elementos chequeados.

#### Alcance

Esta plantilla es aplicable a cada una de las Revisiones de especificaciones del [nombre del artefacto] que se desarrollen.

#### Resumen

La plantilla tiene un carácter flexible para el especialista de calidad o el desarrollador teniendo en cuenta que pueden surgir modificaciones e inclusiones.

#### Forma de Uso:

**Peso:** Define si el indicador a evaluar es crítico o no.

**Evaluación (Eval):** Es la forma de evaluar el indicador en cuestión. El mismo se evalúa de 1 en caso de mal y 0 en caso que elemento revisado no presente errores.

**Cantidad de elementos afectados:** Especifica la cantidad de errores encontrados sobre el mismo indicador.

**Comentario:** Especifica los señalamientos o sugerencias que quiera incluir la persona que aplica la lista de chequeo.

**Estructura del Documento:** Abarca todos los aspectos definidos por el expediente de

proyecto o el formato establecido por el proyecto.

Elementos definidos por la metodología: Abarca todos los indicadores a evaluar según la metodología.

Semántica del documento: Contempla todos los indicadores a evaluar respecto a la ortografía, redacción y demás.

N.P. (No Procede): Se usa para especificar que el indicador a evaluar no se puede aplicar en ese caso.

#### Control de versiones

| Fecha       | Versión | Descripción                              | Autor                       |
|-------------|---------|------------------------------------------|-----------------------------|
| <dd/mmm/yy> | <x.x>   | <descripción de la actividad a realizar> | <nombre del que la elabora> |
| <dd/mmm/yy> | <x.x>   | <descripción de la actividad a realizar> | <nombre del que la aplica>  |

#### Reglas de Confidencialidad

Clasificación: <<Clasificación>>

Este documento contiene información propietaria de **ALBET Ingeniería y Sistemas** y/o "**<<Empresa Cliente>>**", y es emitido confidencialmente para un propósito específico.

El que recibe el documento asume la custodia y control, comprometiéndose a no reproducir, divulgar, difundir o de cualquier manera hacer de conocimientos público su contenido, excepto para cumplir el propósito para el cual se ha generado.

Estas reglas son aplicables a las **11** páginas de este documento.

Estructura de la lista de chequeo

| <b>Estructura del documento</b>               |                                                                                                                       |             |             |                                     |                   |
|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|-------------|-------------|-------------------------------------|-------------------|
| <b>Peso</b>                                   | <b>Indicadores a Evaluar</b>                                                                                          | <b>Eval</b> | <b>(NP)</b> | <b>Cantidad elementos afectados</b> | <b>de Comenta</b> |
| crítico                                       | 1. ¿Está el documento acorde con a la plantilla estándar del proyecto o del expediente de proyecto?                   |             |             |                                     |                   |
| crítico                                       | 2. ¿Contiene las secciones obligatorias definidas en el expediente? (Ver Expediente de Proyecto)                      |             |             |                                     |                   |
| <b>Elementos definidos por la metodología</b> |                                                                                                                       |             |             |                                     |                   |
| <b>Peso</b>                                   | <b>Indicadores a Evaluar</b>                                                                                          | <b>Eval</b> | <b>(NP)</b> | <b>Cantidad elementos afectados</b> | <b>de Comenta</b> |
| crítico                                       | 1. ¿Están todos los requisitos redactados de forma simple y clara para aquellos que vayan a consultarlo en un futuro? |             |             |                                     |                   |

|         |                                                                                                                      |  |  |  |  |  |
|---------|----------------------------------------------------------------------------------------------------------------------|--|--|--|--|--|
|         |                                                                                                                      |  |  |  |  |  |
|         | 2. ¿Debería especificarse algún requisito con más detalle?                                                           |  |  |  |  |  |
|         | 3. ¿Debería especificarse algún requisito con menos detalles?                                                        |  |  |  |  |  |
|         | 4. ¿Todos los requisitos identificados se centran en lo que el sistema debe hacer y no como el sistema debe hacerlo? |  |  |  |  |  |
| crítico | 5. ¿Han sido abordadas e identificadas los valores de entradas y salidas?                                            |  |  |  |  |  |
|         | 6. ¿Han sido incluidos las respuestas válidas y no válidas de los valores de entrada?                                |  |  |  |  |  |
|         | 7. ¿Se han identificado los requerimientos de software y de hardware?                                                |  |  |  |  |  |
|         | 8. ¿Han sido identificadas las restricciones de diseño e implementación?                                             |  |  |  |  |  |

|  |                                                                                                                                                                                                                                                                                   |  |  |  |  |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|--|--|
|  | 9. ¿Han sido identificadas las restricciones de interfaz externa?                                                                                                                                                                                                                 |  |  |  |  |
|  | 10. ¿Los requerimientos de soporte y usabilidad se han identificados?                                                                                                                                                                                                             |  |  |  |  |
|  | 11. ¿Se han identificado los requerimientos de seguridad (confidencialidad, integridad, disponibilidad)?                                                                                                                                                                          |  |  |  |  |
|  | 12. ¿Se puede verificar cada requisito? (Un requisito se dice que es verificable si existe algún proceso no excesivamente costoso por el cual una persona o una máquina pueda chequear que el software satisface dicho requerimiento, ejemplo la especificación del caso de uso). |  |  |  |  |
|  | 13. ¿Se han enumerado los requisitos incluso los que se derivan de otros requisitos?                                                                                                                                                                                              |  |  |  |  |
|  | 14. ¿Se puede trazar                                                                                                                                                                                                                                                              |  |  |  |  |

|                                |                                                                                                                                           |             |             |                                        |           |                    |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------------|----------------------------------------|-----------|--------------------|
|                                | cada requisito al origen en el entorno del problema, (caso de uso del negocio)?                                                           |             |             |                                        |           |                    |
|                                | 15. ¿Se han especificado todos los posibles cambios en los requisitos, incluyendo la probabilidad de cambio?                              |             |             |                                        |           |                    |
|                                | 16. ¿No aparece un mismo requisito en más de un lugar del documento de especificación?                                                    |             |             |                                        |           |                    |
| crítico                        | 17. ¿No existe contradicción entre lo especificado por un requisito y lo especificado por otro?                                           |             |             |                                        |           |                    |
|                                | 18. ¿Existe correspondencia entre el modelo de caso de uso, las Especificaciones Suplementarias y las especificaciones de requerimientos? |             |             |                                        |           |                    |
| <b>Semántica del documento</b> |                                                                                                                                           |             |             |                                        |           |                    |
| <b>Peso</b>                    | <b>Indicadores a Evaluar</b>                                                                                                              | <b>Eval</b> | <b>(NP)</b> | <b>Cantidad de elementos afectados</b> | <b>de</b> | <b>Comentarios</b> |

|         |                                                                                                                                 |  |  |  |  |
|---------|---------------------------------------------------------------------------------------------------------------------------------|--|--|--|--|
| Crítico | 1. ¿Ha identificado errores ortográficos?                                                                                       |  |  |  |  |
| Crítico | 2. ¿Se entiende claramente lo que se ha especificado en el documento?                                                           |  |  |  |  |
|         | 3. ¿El número de página que aparece en el índice coincide con el contenido que se refleja realmente en dicha página?            |  |  |  |  |
|         | 4. ¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento? |  |  |  |  |

Registro de defectos y dificultades detectados

| Elemento              | No  | No conformidad                     | Aspecto correspondiente                   | Etapas de detección             | Significativa | No Significativa | Recomendación | Estado NC                                                  | Respuesta del Equipo Desarrollador                                    |
|-----------------------|-----|------------------------------------|-------------------------------------------|---------------------------------|---------------|------------------|---------------|------------------------------------------------------------|-----------------------------------------------------------------------|
| <Nombre del Elemento> | <1> | <Descripción de la No Conformidad> | <Descripción del Aspecto correspondiente> | <Etapas de detección del error> | <X>           | <X>              | <X>           | [Se coloca el estado de la NC y la fecha, que se revise se | [Esta columna se comienza a llenar a partir de la 2da iteración, y es |

|  |  |  |  |  |  |  |  |                                                                                                                                                             |                                                                                                                                                                      |
|--|--|--|--|--|--|--|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  |  |  |  |  |  |  |  | <i>deja el estado anterior y se coloca el nuevo con la fecha en que se revisó.]</i><br><i>RA: Resuelta</i><br><i>PD: Pendiente</i><br><i>NP: No Procede</i> | <i>responsabilidad del equipo de desarrollo, quien especifica la conformidad con lo encontrado o no y en caso de no proceder la no conformidad explica por qué.]</i> |
|--|--|--|--|--|--|--|--|-------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Evaluación del Artefacto**

Se aborta la revisión del artefacto revisado si:

El promedio de las No Conformidades críticas por casos de uso es superior a uno.

Este criterio es empleado únicamente para los casos de uso. En caso de otro artefacto la revisión se aborta si:

Existen al menos dos indicadores críticos evaluados de mal.

Existe más de una falta de ortografía por página o pantalla en caso de ser una interfaz.

Incumple con más del 50 % de los indicadores a evaluar de la sección Estructura del Documento que posee la lista de chequeo.

Se mantienen las No Conformidades de una revisión a otra.

Se evalúa de regular la calidad del artefacto revisado si el artefacto no cumple los criterios

---

---

para ser abortado y:

Existe una No Conformidad crítica.

La cantidad de elementos afectados de un indicador evaluado de mal es superior a tres.

Estos criterios se cumplen para todas las secciones que tiene la lista de chequeo.

El artefacto es evaluado de bien si no cumple ninguno de los criterios anteriores y:

No existe ninguna No Conformidad relacionada con indicadores con peso crítico.

Si la cantidad de elementos afectados de un indicador que no sea crítico no es mayor que dos.

Evaluación: \_\_\_\_\_

Nombre y Apellido del Evaluador: \_\_\_\_\_

**Anexo 2 Lista de chequeo.**

Especificación de Requisitos

<Nombre del Proyecto>

<Nombre del Producto>

<Versión>

Control de versiones

| Fecha | Versión | Descripción | Autor |
|-------|---------|-------------|-------|
|-------|---------|-------------|-------|

| <dd/mmm/yy>                | <x.x>                                                                      | <detalles> | <nombre> |
|----------------------------|----------------------------------------------------------------------------|------------|----------|
| <b>Tabla de Contenidos</b> |                                                                            |            |          |
| 1.                         | Introducción                                                               | 120        |          |
| 1.1                        | Alcance                                                                    | 122        |          |
| 1.2                        | Definiciones, Acrónimos y Abreviaturas                                     | 122        |          |
| 1.3                        | Referencias                                                                | 122        |          |
| 2.                         | Funcionalidad                                                              | 122        |          |
| 2.1                        | <Requisito Funcional 1..n>                                                 | 122        |          |
| 3.                         | Usabilidad                                                                 | 122        |          |
| 3.1                        | < Requisito de Usabilidad 1..n>                                            | 123        |          |
| 4.                         | Fiabilidad                                                                 | 123        |          |
| 4.1                        | < Requisito de Fiabilidad 1..n>                                            | 124        |          |
| 5.                         | Eficiencia                                                                 | 124        |          |
| 5.1                        | < Requisito de Eficiencia 1..n>                                            | 120        |          |
| 6.                         | Soporte                                                                    | 123        |          |
| 6.1                        | < Requisito de Soporte 1..n>                                               | 123        |          |
| 7.                         | Restricciones de diseño                                                    | 123        |          |
| 7.1                        | < Requisito de Soporte 1..n>                                               | 123        |          |
| 8.                         | Requisitos para la documentación de usuarios en línea y ayuda del sistema. | 123        |          |
| 9.                         | Componentes Comprados                                                      | 123        |          |
| 10.                        | Interfaz                                                                   | 123        |          |
| 10.1                       | Interfazces de usuario                                                     | 125        |          |
| 10.2                       | Interfaces Hardware                                                        | 125        |          |
| 10.3                       | Interfaces Software                                                        | 125        |          |
| 10.4                       | Interfaces de Comunicación                                                 | 125        |          |
| 11.                        | Requisitos de Licencia                                                     | 125        |          |
| 12.                        | Requisitos Legales, de Copyright y otros.                                  | 125        |          |
| 13.                        | Estándares Aplicables                                                      | 125        |          |

---

Introducción

*Alcance*

*[Proyectos con los que se involucra la Especificación]*

*Definiciones, Acrónimos y Abreviaturas*

*Referencias*

*[Lista de documentos a los que se hace referencia en la Especificación]*

| Código | Título                                           |
|--------|--------------------------------------------------|
| [1]    | Documento 1                                      |
| [2]    | Documento 2                                      |
| [3]    | Modelo de Diseño - Módulo de Administración v0.0 |

Funcionalidad

[Esta sección describe los requisitos funcionales del sistema expresados en lenguaje natural. Típicamente se organiza por características pero también son apropiados métodos alternativos de organización como por ejemplo por usuario o subsistema.]

<Requisito Funcional 1..n>

[Descripción del requisito]

Usabilidad

[Esta sección incluye todos los requisitos que afectan la usabilidad. Ejemplos:

*Especificar el tiempo de entrenamiento requerido para que usuarios normales y avanzados sean productivos operando el sistema.*

---

---

< Requisito de Eficiencia 1..n>

[Descripción del requisito]

Soporte

[Esta sección indica cualquier requisito que refuerce el soporte o mantenimiento del sistema a construir, incluyendo normas de codificación, convenciones para nombrado, bibliotecas de clase, el acceso y utilidades de mantenimiento.]

< Requisito de Soporte 1..n>

[Descripción del requisito]

Restricciones de diseño

[Esta sección debe indicar cualquier restricción de diseño en el sistema a construir. Las restricciones representan decisiones de diseño que se han tomado y a las cuales es necesario adherirse. (Ej. lenguajes de programación, requisitos de proceso de software, el uso prescrito de herramientas de desarrollo, restricciones de arquitectura y diseño, componentes comprados, las bibliotecas de la clase, etc.)]

< Requisito de Soporte 1..n>

[Descripción del requisito]

Requisitos para la documentación de usuarios en línea y ayuda del sistema.

[Describe los requisitos para la documentación de usuarios en línea, la ayuda del sistema, ayuda relacionada con avisos, etc.].

Componentes Comprados

[Esta sección describe cualquier componente comprado y a ser usado en el sistema, cualquier licencia aplicable o restricciones del uso, y cualquier compatibilidad/interoperabilidad asociada o estándares de interfaz.]

*Especificar requisitos acordes con estándares de usabilidad establecidos]*

< Requisito de Usabilidad 1..n>

[Descripción del requisito]

Fiabilidad

[En esta sección se especifican los requisitos relacionados con la Fiabilidad.

Ejemplos:

---

---

Disponibilidad – especificar por ciento de tiempo disponible (xx.xx%), horas de uso, acceso para mantenimiento, modo de funcionamiento degradado etc.

Tiempo medio entre fallos – usualmente se especifica en horas pero puede también especificarse en términos de días, meses o años.

Tiempo medio de reparación – Cuanto tiempo está permitido que el sistema quede fuera de operación luego de haber fallado?

Exactitud – especificar la precisión y exactitud requerida en las salidas del sistema.

Máximo de errores – usualmente es expresado en términos de errores/MLC (miles de líneas de código) o errores/puntos de función.

Errores – categorizar los errores en términos de menores, significativos y críticos: los requisitos deben definir que se entiende por error crítico (ej. Pérdida total de los datos o inhabilitadas para el uso ciertas partes del funcionamiento del sistema).]

< Requisito de Fiabilidad 1..n >

[Descripción del requisito]

Eficiencia

[Deben perfilarse en esta sección las características de la eficiencia del sistema. Incluir los tiempos de respuesta específicos. Donde sea aplicable, hacer referencia a los Casos de Uso por el nombre.

Tiempo de respuesta por transacción (promedio, máximo).

Rendimiento (ej. transacciones por segundo, cantidad de datos que pueden ser transferidos en un segundo).

Capacidad (ej. número de clientes o transacciones que el sistema puede alojar).

Modos de degradación (cual es el modo de operación aceptable cuando el sistema de alguna forma ha sido degradado).

Utilización de recursos (memoria, disco, comunicaciones, etc.)

Interfaz

[Esta sección define las interfaces que deben soportadas por la aplicación. Debe contener la especificidad adecuada, protocolos, puertos y direcciones lógicas, etc.,

---

---

para que el software pueda desarrollarse y verificarse contra los requisitos de la interfaz.]

Interfaces de usuario

[Describe las interfaces de usuario que deben ser implementadas por el software.]

Interfaces Hardware

[Esta sección define cualquier interfaz del hardware que será soportada por el software, incluyendo la estructura lógica, direcciones físicas, el comportamiento esperado, etc.]

Interfaces Software

[Esta sección describe las interfaces del software a otros componentes del sistema del software. Éstos pueden ser componentes comprados, componentes reutilizados de otra aplicación o componentes que se desarrollan para subsistemas fuera del alcance de este documento, pero con esta aplicación debe actuar recíprocamente.]

Interfaces de Comunicación

[Describe cualquier interfaz de comunicaciones a otros sistemas o dispositivos como las redes de área locales, los dispositivos remotos, etc.]

Requisitos de Licencia

[Define cualquier requisito de licencia o restricción de uso que serán seguidos por el software.]

Requisitos Legales, de Derecho de Autor y otros.

[Esta sección describe cualquier denegación legal necesaria, garantías, notificaciones de derecho de autor, patentes, marca comercial o complacencia con logotipo para el software.]

Estándares Aplicables

[Esta sección describe por referencia cualquier norma o estándar aplicable y las secciones específicas que aplicadas al sistema. Por ejemplo, podría incluir estándares legales, de calidad, regulatorios, normas de la industria para la usabilidad, el interoperabilidad, internacionalización, integración con el sistema

---

---

operativo, etc.]

**Anexo 3 Estructura del documento de especificación de requisitos de software.**

Introducción

Propósito

[Resumen del propósito de este documento]

Alcance

[Breve descripción del alcance del modelo del negocio]

Definiciones, Acrónimos y Abreviaturas

Referencias

[Lista de documentos a los que se hace referencia]

Actores del Sistema

[Se especifican todos los actores del negocio y se le asocia una descripción simple de cada uno de ellos]

| Actor | Descripción |
|-------|-------------|
|       |             |

Diagrama de Casos de Uso del Sistema

**Modelo de Casos de Uso del sistema**

Rector

---

<Nombre del Proyecto>

<Nombre del producto>

<Versión>

Control de versiones

| Fecha       | Versión | Descripción | Autor    |
|-------------|---------|-------------|----------|
| <dd/mmm/yy> | <x.x>   | <detalles>  | <nombre> |

Tabla de contenidos

**AGRADECIMIENTOS** .....

**DEDICATORIA** .....

**ÍNDICE DE TABLAS**.....

**INTRODUCCIÓN** .....

**CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**.....

INTRODUCCIÓN .....

1.1 REQUISITOS DE SOFTWARE. DEFINICIONES Y CONCEPTOS. ....

    1.1.1 *Requisitos funcionales y requisitos no funcionales.* .....

1.2 INGENIERÍA DE REQUISITOS DE SOFTWARE. ....

    1.2.1 *Características de la Ingeniería de requisitos.*.....

    1.2.2 *Importancia de la Ingeniería de requisitos de software.*.....

    1.2.3 *Etapas de la Ingeniería de requisitos de software.*.....

1.3 VALIDACIÓN DE REQUISITOS DE SOFTWARE.....

|                                                    |                                                                                                             |
|----------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| 1.3.1                                              | <i>Diferencia entre verificación y Validación de requisitos de software.....</i>                            |
| 1.4                                                | MÉTODOS DE VALIDACIÓN DE REQUISITOS DE SOFTWARE.....                                                        |
| 1.4.1                                              | <i>Revisiones.....</i>                                                                                      |
| 1.4.2                                              | <i>Prototipos orientados a clientes y usuarios.....</i>                                                     |
| 1.4.3                                              | <i>Auditorias.....</i>                                                                                      |
| 1.4.4                                              | <i>Matrices de trazabilidad.....</i>                                                                        |
| 1.5                                                | CALIDAD DE SOFTWARE. DEFINICIONES.....                                                                      |
| 1.6                                                | NOTACIÓN DE MODELADO.....                                                                                   |
| 1.6.1                                              | <i>BPMN.....</i>                                                                                            |
| 1.6.2                                              | <i>UML.....</i>                                                                                             |
| 1.7                                                | HERRAMIENTA.....                                                                                            |
| 1.7.1                                              | <i>Visual Paradigm.....</i>                                                                                 |
| 1.7.2                                              | <i>Rational Rose.....</i>                                                                                   |
|                                                    | CONCLUSIONES.....                                                                                           |
| <b>CAPÍTULO 2: REFINAMIENTO DE REQUISITOS.....</b> |                                                                                                             |
|                                                    | INTRODUCCIÓN.....                                                                                           |
| 2.1                                                | ALCANCE.....                                                                                                |
| 2.1.1                                              | <i>Propósito.....</i>                                                                                       |
| 2.2                                                | OBJETIVOS.....                                                                                              |
| 2.3                                                | CARACTERÍSTICAS.....                                                                                        |
| 2.4                                                | <i>Estructura.....</i>                                                                                      |
| 2.5                                                | <i>Responsables y productos de entradas y salidas.....</i>                                                  |
| 2.6                                                | <i>Descripción del procedimiento para la validación de requisitos de software.....</i>                      |
| 2.7                                                | <i>Fases del procedimiento.....</i>                                                                         |
| 2.8                                                | <i>Fase I. Validación del listado de requisitos de software.....</i>                                        |
| 2.9                                                | <i>Fase II. Validación del documento de especificación de requisitos.....</i>                               |
| 2.10                                               | <i>Fase III. Validación del modelo de casos de uso del Sistema.....</i>                                     |
| 2.11                                               | <i>Generalidades del procedimiento para validar requisitos de software.....</i>                             |
| 2.12                                               | <i>Fase I. Validación del listado de requisitos de software. Particularidades.....</i>                      |
| 2.13                                               | <i>Fase II. Validación del documento de especificación de requisitos de software. Particularidades.....</i> |
| 2.14                                               | <i>Validación del modelo de casos de uso del sistema. Particularidades.....</i>                             |

CONCLUSIONES .....

**CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL NVP .....**

INTRODUCCIÓN .....

3.1 PLANIFICACIÓN DEL PROCEDIMIENTO .....

3.2 APLICACIÓN DE LA FASE I. RESULTADOS .....

3.3 APLICACIÓN DE LA FASE II. RESULTADOS .....

3.4 APLICACIÓN DE LA FASE III. RESULTADOS .....

CONCLUSIONES .....

**CONCLUSIONES.....**

**RECOMENDACIONES.....**

**BIBLIOGRAFÍA.....**

**GLOSARIO DE TÉRMINOS.....**

ALCANCE .....

DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS .....

REFERENCIAS .....

[Figura que ilustre del modelo de casos de uso del sistema]

Especificación de los Casos de Uso

<Primer Caso de Uso del Sistema>

*Descripción de Casos de Uso*

[Se describe en la tabla los detalles del caso de uso en fusión de acción del actor y respuesta del sistema.]

Si se decide tener un documento independiente para la definición de cada Caso de Uso, en esta sección se haría referencia a ese documento]

Caso de Uso:

|                              |                       |
|------------------------------|-----------------------|
| Actores:                     |                       |
| Resumen:                     |                       |
| Precondiciones:              |                       |
| Referencias                  |                       |
| Prioridad                    |                       |
| Flujo Normal de Eventos      |                       |
| Sección ""                   |                       |
| Acción del Actor             | Respuesta del Negocio |
|                              |                       |
| <i>Prototipo de Interfaz</i> |                       |
| Flujos Alternos              |                       |
| Acción del Actor             | Respuesta del Negocio |
|                              |                       |
| <i>Prototipo de Interfaz</i> |                       |
| Poscondiciones               |                       |

**Anexo 4 Estructura del modelo de casos de uso del sistema.**