

# **Universidad de las Ciencias Informáticas**

## **Facultad 15**



**Título: Validación de Requisitos con Especialistas Funcionales para Proyectos de Desarrollo de Software en la UCI.**

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

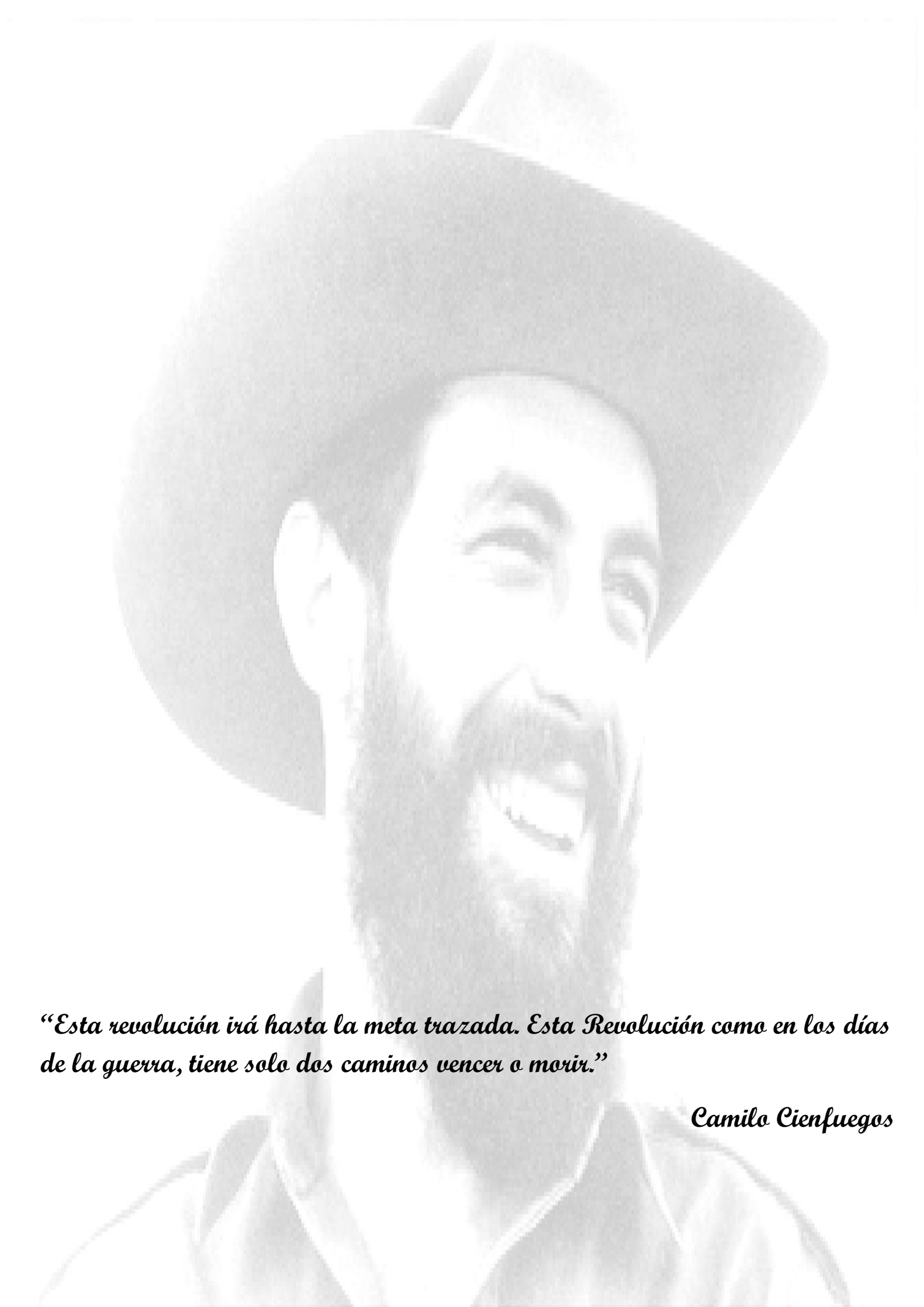
**Autor:**

Arelis Nelsa Zamora Abreu

**Tutor:**

Ing. Ana Marys García Rodríguez

CIUDAD DE LA HABANA, JUNIO 2010.



*“Esta revolución irá hasta la meta trazada. Esta Revolución como en los días de la guerra, tiene solo dos caminos vencer o morir.”*

*Camilo Cienfuegos*

## DECLARACIÓN DE AUTORÍA

### **DECLARACIÓN DE AUTORÍA:**

Declaro ser la única autora del trabajo titulado:

Validación de Requisitos con Especialistas Funcionales para Proyectos de Desarrollo de Software en la UCI.

Autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

Ing. Ana Marys García Rodríguez

\_\_\_\_\_

Arelis Nelsa Zamora Abreu

## DATOS DE CONTACTO

### **DATOS DE CONTACTO:**

**Nombre y Apellidos:** Ing. Ana Marys García Rodríguez.

**País:** Cuba.

**Ciudadanía:** cubana.

**Correo:** agarcia@uci.cu

**Institución:** Universidad de las Ciencias Informáticas (UCI)

**Dirección:** Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.

### **Currículo:**

Graduada de Ingeniera en Ciencias Informáticas en Julio del año 2007 en la Universidad de Ciencias Informáticas (UCI). Ha recibido cursos de postgrado: DIU, Negociación y Gestión de Contratación, Técnicas de Dirección, CMMI, Inglés Básico, Curso Básico de Gestión de Proyectos. Presentó dos trabajos en el evento UCIENCIA 2008: Autora y ponente del trabajo "Estrategia para la Gestión de Configuración en Organizaciones Productivas" que se encuentra publicado en las memorias del evento, coautora del trabajo "Apuntes sobre la Disciplina Ambiente" que se encuentra publicado en la Serie Científica de la UCI. Se encuentra matriculada en la maestría de Gestión de Proyectos. Se ha desempeñado como revisora en dos revisiones organizadas por la Dirección de Calidad UCI. Se desempeñó como gestora de cambios y analista del proyecto SIGEP, cumpliendo misión en Venezuela en dos ocasiones para la captura de requerimientos de este proyecto. Actualmente es analista del proyecto Aduana. Se ha desempeñado durante dos años como profesora de Ingeniería de Software en la facultad 4, y actualmente asume el cargo de Jefa de Asignatura.

AGRADECIMIENTOS

*Agradezco a Fidel, a la Revolución y a la Universidad de las Ciencias Informáticas por darme la oportunidad de formarme y hacer de mis sueños una realidad.*

*A mis padres y a mi abuelita, por su apoyo moral y espiritual en todo momento.*

*A mis hermanas por ser las niñas que impulsan mi vida en todo momento para serles un buen ejemplo de superación.*

*A mi tutora, que con su seriedad científica debidamente matizada de criticidad y humor, hicieron de esta experiencia académica algo muy satisfactorio, pero más que mi tutora por ser mi amiga.*

*A los profesores que han compartido sus experiencias durante estos años. Siempre les estaremos agradecidos.*

*A todas mis amistades, que de una forma u otra han hecho de mi vida en la UCI una etapa inolvidable en mi vida.*

*A mi amigo Jorge Luis Beatón Torres por estar desde el día que lo conocí siempre a mi lado, en la buenas, malas y peores.*

*A todos los que de una forma u otra, me han ayudado durante la realización de esta investigación y en el cursar de toda mi carrera pues contribuyeron a que este momento sea realidad: amigos, compañeros, en fin gracias a todos.*

DEDICATORIA

*Les dedico este trabajo de diploma a mis padres Belkis Abreu Morffi, Aldo Fernández Gutiérrez y a mi abuela Nelsa González Rodríguez, pues gracias a ellos he logrado llegar hasta el día de hoy. Por haberme sabido guiar, apoyar y darme ánimos en los momentos en los que he creído que no puedo seguir adelante. Por darme la estabilidad necesaria para concentrar mis pensamientos en mis estudios. Por haberme dado todo su amor, su amistad, su cariño, su comprensión, su tolerancia, por aceptarme como soy con todos mis defectos y virtudes, por siempre estar a mi lado y nunca abandonarme, hasta en los peores momentos, siempre he contado con su ayuda. Y a mi hermana, para que siga mi ejemplo y pronto continuando sus estudios, logre cursar una carrera profesional, para doble orgullo de mis padres.*

### **RESUMEN**

La presente investigación trata una de las áreas más polémicas dentro del desarrollo de software: la Ingeniería de Requisitos (IR) y específicamente dentro de esta, el proceso de validación de requisitos, haciendo énfasis en la importancia de este proceso para lograr un producto final con calidad y reducir los costos y tiempos de corrección de defectos en etapas posteriores a la validación de requisitos. Se estudiaron las características que debe cumplir un documento de Especificación de Requisitos de Software (ERS) para lograr que el software propuesto a construir cumpla las expectativas y necesidades del cliente.

Para la obtención de la propuesta se desarrolló un estudio de las técnicas utilizadas a escala mundial y en la Universidad de Ciencias Informáticas (UCI) para la validación de requisitos. Se realizó además un análisis de los atributos de calidad más importantes que se deben tener en cuenta para la confección de un ERS, así como las métricas que permiten medir el cumplimiento de dichos atributos. Se tuvieron en cuenta las características del método Delphi como parte de la propuesta de validación y se valoró la propuesta mediante su aplicación en un proyecto de desarrollo de software.

Como resultado se obtuvo una propuesta de validación de requisitos que tiene como base: características del método Delphi, la aplicación de listas de chequeo y métricas que permiten medir los atributos de calidad de un ERS, con el objetivo de minimizar la cantidad de errores generados en la fase de IR que puedan trascender a etapas posteriores del desarrollo del software.

**ÍNDICE**

<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>5</b>
Introducción.....	5
Ingeniería de Requisitos .....	5
Técnicas de Validación de Requisitos: .....	7
Atributos de Calidad y Métricas: .....	10
Figura 1. Medición de la comprensión de requisitos. ....	12
Método Delphi: .....	16
Características de pronóstico Delphi. ....	16
Pasos para aplicar el Método Delphi. ....	17
<b>CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES.</b> .....	<b>19</b>
Verificación:.....	19
Lista de Chequeo. ....	19
Medir Atributo de Calidad Verificable: .....	20
Validación. ....	22
Actividad: Seleccionar Especialistas Funcionales para la validación: .....	23
Actividad: Informar a los Especialistas Funcionales sobre la validación y solicitar disposición para la misma:.....	24
Actividad: Diseñar el cuestionario a aplicar:.....	24
Actividad: Entregar cuestionario y documentación ERS a Especialistas Funcionales: .....	25
Actividad: Analizar respuestas de los cuestionarios: .....	25
Actividad: Elaborar Informe de Incidencias: .....	26
Actividad: Modificar el ERS:.....	26
Actividad: Entregar Informe de Incidencias a Especialistas Funcionales: .....	26
Actividad: Preparar el Informe Final: .....	26
Actividad: Certificar el ERS: .....	26



## ÍNDICE

Medición de los Atributos de Calidad:.....	27
<b>CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA .....</b>	<b>31</b>
Proyecto MOCIC.....	31
Aplicación de la Estrategia de Validación de requisitos del Proyecto: .....	33
Aplicación de la Estrategia de Validación de Requisitos con Especialistas Funcionales: .....	35
Impacto y mejoras del proyecto Motor de Categorización Inteligente de Contenidos (MOCIC) a partir de la validación de los requisitos para etapas posteriores.....	47
<b>CONCLUSIONES .....</b>	<b>48</b>
<b>RECOMENDACIONES.....</b>	<b>49</b>
<b>BIBLIOGRAFÍA.....</b>	<b>50</b>
<b>ANEXOS.....</b>	<b>52</b>
Anexo 1: ERS Validación1 v1.0. ....	52
Anexo 2: ERS Validación1 v1.1. ....	53
Anexo 3: ERS Validación2 v1.0. ....	54
Anexo 4: ERS Validación2 v1.1. ....	55
Anexo 5: ERS Validación2 v1.2. ....	56
Anexo 6: Respuesta 1 del Cuestionario.....	57
Anexo 7: Respuesta 2 del Cuestionario.....	60
Anexo 8: Firmas de especialistas. ....	63
Anexo 9: Impacto y mejoras para MOCIC.....	64
Anexo 10: Matriz de trazabilidad. ....	65
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>66</b>

## ÍNDICE DE FIGURAS

### ÍNDICE DE FIGURAS

<b>Figura 1. Medición de la comprensión de requisitos. ....</b>	<b>12</b>
<b>Figura 2. Validación con Especialistas Funcionales. ....</b>	<b>22</b>
<b>Figura 3. ERS Validación1 v1.0. ....</b>	<b>52</b>
<b>Figura 4. ERS Validación1 v1.1. ....</b>	<b>53</b>
<b>Figura 5. ERS Validación2 v1.0. ....</b>	<b>54</b>
<b>Figura 6. ERS Validación2 v1.1. ....</b>	<b>55</b>
<b>Figura 7. ERS Validación2 v1.2. ....</b>	<b>56</b>

### **INTRODUCCIÓN**

El desarrollo del software ha evolucionado en gran medida durante las últimas décadas al punto de expandirse a todos los sectores de la vida cotidiana del ser humano. Con el cursar de los años las empresas han optado por la informatización de sus negocios con el objetivo de agilizar los procesos y facilitar el trabajo de sus empleados, reportándose elevadas ganancias y beneficios que han llevado a estas empresas e instituciones a fomentar el desarrollo de esta industria para favorecer sus activos económicos e introducirse en la gran mayoría de los casos en el mercado mundial.

Actualmente, las tecnologías han avanzado con mucha rapidez, a una velocidad mayor de la que el ser humano emplea en adaptarse a ellas. Por su relativa juventud respecto a otras industrias que llevan siglos de desarrollo e investigación, aún la industria de software se encuentra en un arduo proceso de investigación y documentación para la mejora continua de sus procesos.

Dentro del desarrollo de software una de las áreas más importantes es la Ingeniería de Requisitos (IR), pues de ella depende en gran medida el resto de las actividades que vayan a desarrollarse para la obtención del producto final.

En el desarrollo del software los requisitos constituyen el eslabón fundamental para el desarrollo de las etapas posteriores de producción y de ellos depende en gran medida el éxito o fracaso de un proyecto, por esto es importante que exista una garantía de que estén correctamente especificados y que en realidad satisfagan las necesidades del cliente que solicita el software.

Actualmente los principales problemas en la IR se enmarcan en tres categorías fundamentalmente:

- Problemas de alcance: Límite del sistema incorrectamente definido.
- Problemas de comprensión: Los usuarios y/o clientes no saben en realidad lo que necesitan, se omiten detalles que pueden traer como resultado requisitos ambiguos, no existe un total entendimiento del dominio del negocio, no hay un consenso entre clientes y desarrolladores sobre las demandas del cliente y lo que éste realmente necesita.

## INTRODUCCIÓN

- Problemas de volatilidad: Lidar con la naturaleza volátil de los requisitos que cambian con el tiempo.

Un mal desarrollo de la IR puede provocar que en ocasiones lo que se implementa no es realmente lo que el cliente necesita o solicitó, que se detecte un gran número de no conformidades en etapas finales del desarrollo de software (construcción, soporte) que provienen de una mala comprensión del negocio y/o IR.

Muchas veces existe más de un especialista vinculado a determinados requisitos de un módulo o subsistema y no se tienen en cuenta todos los criterios, provocando que en etapas posteriores se tenga que hacer una regresión porque el software implementado no cumple eficazmente con todas las funcionalidades necesarias.

Cuando varios especialistas atienden un área específica del negocio, suelen existir también contradicciones o divergencias de criterios y los proyectos suelen realizar validaciones con sólo un especialista por lo general, obviando los criterios y necesidades del resto.

En ocasiones cuando se realiza un análisis en conjunto con especialistas de diferentes rangos de dirección, algunos de menor rango restringen sus criterios por no contradecir a sus superiores y esto a largo plazo afecta el desarrollo al punto que en etapas posteriores a una validación de requisitos se agreguen nuevas funcionalidades no previstas en un inicio.

Los clientes suelen interpretar de maneras diferentes un mismo requisito y a veces no tienen conocimiento de esta divergencia de interpretación, el requisito es aprobado y esta ambigüedad puede provocar retrasos en el desarrollo.

Cuando las validaciones se centran en los prototipos, suelen presentarse problemas con el cliente si este no es capaz de entender que no es realmente el sistema lo que se está mostrando y que no todas las funcionalidades internas del sistema podrán ser observadas en dichos prototipos.

También suele ocurrir que la validación se basa en una mera tarea de revisar una especificación y esperar a que un cliente diga si está bien o mal, luego se realizan cambios a los defectos detectados y supuestamente los errores fueron erradicados sin pasar por una segunda revisión.

## INTRODUCCIÓN

Una elicitación muy pobre de los requisitos conlleva a la implementación de cambios con demasiada frecuencia y la definición incorrecta del alcance del sistema, elevando el presupuesto a costear y teniendo que realizarse ajustes y nuevas planificaciones. Esta problemática podría devenir en una posible cancelación del proyecto o un retraso en el desarrollo del producto, provocando siempre la insatisfacción del cliente. Boehm afirma que sólo entre un 9% y un 12% de la duración de un proyecto se invierte en la IR. Estudios más recientes afirman que entre el 44% y el 80% de los errores proceden de la IR. (Boehm, 1976)

Cuando no se desarrolla una buena validación de los requisitos, se queda a expensas de que posibles errores sean detectados en momentos de diseño e implementación del requisito y las consecuencias pudieran ser drásticas en un sistema donde la trazabilidad entre requisitos sea abundante y compleja, más cuando ya se haya trabajado sobre algunos de estos requisitos.

Los requisitos son actualmente un tema bastante polémico dentro del desarrollo del software, al punto que muchas de las causas de fracasos y retrasos de los proyectos, provienen de esta área. Sucede que muchas veces los requisitos no se validan correctamente y/o se escapan muchos errores importantes que trascienden a etapas posteriores del desarrollo de software como diseño, implementación y soporte.

Actualmente existen técnicas de validación de requisitos que son utilizadas para certificar los mismos como correctos y continuar con el desarrollo del producto. A pesar de esto, se siguen presentando incongruencias en fases posteriores a las validaciones realizadas a los requisitos. Por lo anteriormente expresado y la importancia que sugiere la calidad de los requisitos de un sistema, se necesita definir un correcto proceso de validación que garantice el cumplimiento de los atributos de calidad de los requisitos, pudiéndose evitar muchos problemas que afectan posteriormente el desarrollo del producto y por ende la satisfacción del cliente.

De esta manera se llega al siguiente **problema de la investigación**: ¿Cómo minimizar la cantidad de errores que se cometen en la fase de Ingeniería de Requisitos en proyectos de desarrollo de Software de la UCI?

Constituyó **objeto de estudio** de la investigación: la validación de requisitos en proyectos de desarrollo de software; siendo el **campo de acción**: el proceso de validación de requisitos en proyectos de la UCI.

## INTRODUCCIÓN

Para dar solución al problema de investigación se define como **objetivo general**: definir un proceso de validación de requisitos que garantice el cumplimiento de los atributos de calidad para minimizar la cantidad de errores generados en la fase de IR que puedan trascender a etapas posteriores del desarrollo del software.

Para dar cumplimiento al objetivo de la investigación se acometieron las siguientes **tareas de la investigación**:

1. Caracterizar los aspectos fundamentales de la validación de requisitos enfatizando en los atributos de calidad de la documentación del ERS.
2. Caracterizar métricas de medición de atributos de calidad de un ERS.
3. Especificar atributos de calidad por los que se deben regir los requisitos de software.
4. Definir actividades, involucrados y artefactos del proceso de validación.
5. Definir métricas y listas de chequeo a emplear en el proceso de validación.
6. Valorar la propuesta de validación a partir de su aplicación en un proyecto.

La **estructura del contenido** de la investigación consta de 3 capítulos:

- Capítulo 1: Aborda la fundamentación teórica de la investigación haciendo énfasis en el proceso de validación de requisitos, las técnicas de validación existentes, los atributos que deben medir la calidad de los requisitos y métricas que permiten realizar dichas mediciones.
- Capítulo 2: Abarca el diseño de la solución de la investigación, que constituye la propuesta de validación de requisitos con especialistas funcionales.
- Capítulo 3: Hace referencia a la validación de la propuesta a partir de su aplicación en un proyecto de desarrollo de software.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

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

### **Introducción:**

El presente capítulo abarca los conceptos más importantes de IR, validación de requisitos, así como atributos de calidad que deben cumplir los requisitos y métricas que permiten medir el cumplimiento de los mismos.

### **Ingeniería de Requisitos:**

La IR es una de las etapas más importante en el desarrollo de software. La definición de las necesidades de un sistema es un proceso complejo, pues hay que identificar los requisitos que el sistema debe cumplir para satisfacer las necesidades de los clientes y usuarios finales, así como las restricciones sobre las que el sistema debe operar.

La IR se divide en cinco procesos importantes para su buen desarrollo: elicitación, análisis, especificación, validación y gestión. La elicitación de los requisitos tiene como objetivo esencial identificar las necesidades del cliente y los usuarios finales; comienza cuando el equipo de desarrolladores del software reúne todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de los requisitos que debe tener el sistema para satisfacer las necesidades del negocio en cuestión, así como la información que puede ser brindada por documentos, aplicaciones existentes, entrevistas u otras vías. Luego los requerimientos identificados son especificados y documentados, posteriormente se realiza el proceso de análisis de requisitos para identificar el nivel de prioridad con que deben ser desarrollados los requisitos y poder así establecer una vista arquitectónica funcional del sistema software y por último son validados con el cliente para tener en cuenta si falta alguno por definir y si no existen inconsistencias. El proceso de gestión de requisitos consiste en el control de los cambios que se realizan sobre los requisitos y el seguimiento de los mismos para mantener la integridad del producto software.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### Elicitación:

1. Obtener información sobre el dominio del problema y el sistema actual.
2. Preparar y realizar las reuniones de elicitación.
3. Identificar y/o revisar los objetivos del sistema.
4. Identificar y/o revisar los requerimientos de almacenamiento de información.
5. Identificar y/o revisar los requisitos funcionales.
6. Identificar y/o revisar los requisitos no funcionales.

### Especificación:

1. Descripción de los requerimientos funcionales.
2. Descripción de los requerimientos no funcionales.
3. Descripción de las características de diseño.
4. Modelado de los requerimientos mediante el artefacto seleccionado (Diagramas de Flujo, Casos de Uso, ERS).

### Análisis:

1. Se priorizan los requerimientos modelados en los artefactos del sistema.
2. Se genera el documento de la arquitectura del sistema y de las diferentes vistas del mismo.

### Validación:

1. Trazabilidad de los requerimientos.
2. Evaluación de los requerimientos.
3. Generación del plan de pruebas.

### Gestión:

1. Gestión de los Involucrados.
2. Gestión de la planeación de requerimientos.
3. Gestión de la información y Configuración.



## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### 4. Gestión de Aseguramiento de la Calidad.

Es de vital importancia que la IR se realice correctamente pues de esto depende el éxito del proyecto. Es el punto de partida, en donde se define qué se debe automatizar, las cualidades o capacidades que tienen que ser alcanzadas por el sistema para satisfacer las necesidades de los clientes, se define el ámbito del sistema a construir y se realiza una estimación de los recursos y el tiempo que se necesita para el desarrollo del sistema.

#### **Técnicas de Validación de Requisitos:**

Para la validación de requisitos se utilizan diversas técnicas, generalmente se realiza una reunión con el cliente para revisar los modelos y detectar inconsistencias o errores. La validación de los requisitos es un proceso que tiene como misión principal asegurar que los requisitos definidos en el ERS son realmente las funcionalidades que el cliente solicitó y que los mismos tengan una descripción correcta, para lograr un entendimiento de los requisitos al pasar a etapas de diseño e implementación de los mismos. La definición de los requisitos define realmente qué debe realizar el sistema para satisfacer las necesidades del cliente.

- **Revisión de requisitos:**

La revisión de requisitos es una de las mejores técnicas de validación de requisitos. Generalmente las revisiones de requisitos permiten:

- Descubrir una gran cantidad de defectos en los requisitos.
- Reducir los costos de desarrollo entre un 25% y un 30%.
- Reducir el tiempo de pruebas entre un 50% y un 90%.

Las revisiones de requisitos consisten en una o varias reuniones planificadas, donde se intenta confirmar que los requisitos poseen los atributos de calidad deseados. Estas reuniones son llevadas a cabo por el analista principal del proyecto y la representación del cliente. El resultado final de las reuniones de revisión es un documento que contiene la lista de defectos localizados y una lista de acciones recomendadas.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- **Auditorías:**

La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una lista de chequeo predefinida o definida por el equipo de calidad del proyecto, sólo una muestra es revisada.

- **Matrices de trazabilidad:**

Consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos corresponden a cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos.

La matriz de trazabilidad se usa también para ver si los casos de uso que tenemos satisfacen todos los requerimientos del sistema. Una matriz de trazabilidad de los requisitos se utiliza para comprobar si los requisitos del proyecto actual se están cumpliendo. Una matriz de trazabilidad garantiza la integridad de los requisitos al implementar cambios en el sistema.

Las matrices de trazabilidad pueden utilizarse para verificar y validar las especificaciones de un sistema, así como garantizar que en todos los documentos entregables finales estén incluidas las especificaciones del sistema. De esta forma se puede mejorar la calidad de un sistema mediante la identificación de requisitos que no son abordados por los elementos de configuración durante el diseño e implementación y la identificación de los elementos de configuración adicionales que no son necesarios.

Aunque el desarrollo y mantenimiento de las matrices de trazabilidad puede llevar mucho tiempo, son una referencia rápida durante las actividades de verificación y validación.

- **Prototipos:**

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Otras propuestas se basan en obtener de la definición de requisitos, prototipos de la interfaz de usuario que sin tener la totalidad de las funcionalidades del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema.

Los prototipos son un método de validación ampliamente utilizado en muchas disciplinas y en todos los casos, los principios subyacentes son los mismos; el prototipado consiste en la creación de una maqueta o versión del producto final. Los objetivos de los prototipos varían en función de la disciplina. En el caso de los requisitos, los prototipos se utilizan fundamentalmente, para comprobar la corrección y completitud de la especificación de requisitos. Existen varios tipos de prototipos, cada uno de los cuales permite la realización de un tipo determinado de pruebas y con un determinado nivel de realismo. Presenta como problema que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final.

- **Generación de casos de prueba (test de requisitos):**

Esta técnica tiene como objetivo comprobar la verificabilidad de los requisitos. Consiste en la definición de casos de prueba que permitan verificar el cumplimiento de los requisitos funcionales. El único método existente para comprobar la verificabilidad de un requisito es que sea posible definir uno o varios casos de prueba para dicho requisito. Los casos de prueba son artefactos bien definidos en el contexto de la prueba del software. En dicho contexto, un caso de prueba es la descripción de una acción bien definida que se debe realizar con el software. Por acción bien definida, debe entenderse que están perfectamente descritos tanto los datos de entrada como las tareas a realizar y los resultados esperados. Durante la validación de requisitos, los casos de prueba se utilizan del mismo modo que durante la prueba del software; es necesario poder describir, como se ha indicado anteriormente, tanto los datos de entrada como las tareas a realizar y los resultados esperados, para lo cual es necesario que estén perfectamente descritos los requisitos.

- **Creación de manuales de usuario:**

Esta técnica consiste en verificar si la especificación de requisitos contiene el suficiente detalle como para preparar el manual de usuario del sistema. De no ser así, podría ocurrir que la especificación fuera incompleta.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- **Reviews o Walk-throughs:**

Esta técnica consiste en la lectura y corrección de la documentación o modelado de la definición de requisitos completa. Trayendo consigo que solamente se puede validar la correcta interpretación de la información transmitida o ya capturada. Siendo más difícil verificar la consistencia de la documentación o información faltante.

### **Atributos de Calidad y Métricas:**

Para que la validación de los requisitos sea realizada con éxito se tiene que tener en cuenta que el documento ERS cuente con los atributos de calidad.

Existen métricas enfocadas a la medición de los atributos de calidad. El proceso de estimación del software es un área en el que se están proponiendo métodos y técnicas desde hace 15 años. La mayoría de los procesos existentes describen cómo aplicar un método simple de predicción, que normalmente está basado en uno o más modelos algorítmicos. Estimar consiste en determinar el valor de una variable desconocida a partir de otras conocidas, o de una pequeña cantidad de valores conocidos de esa misma variable.

- **Verificable:**

Un ERS es verificable sólo si cada requisito especificado es verificable.

Un requisito es verificable sólo si existe un proceso finito de costo-efectivo con el cual una persona o una máquina puedan verificar que el producto de software cumple el requisito. Cualquier requisito ambiguo es no verificable.

Una manera de medir este atributo es a partir del porcentaje de requisitos verificables respecto al total de requisitos documentados en el ERS.

**Medición del Atributo de Calidad Verificable:** (Davis, Alan; Overmyer, Scott; Jordan, Kathleen; Caruso, Joseph; Dandashi, Fatma; Dinh, Anhtuan; Kincaid, Gary; Ledebouer, Glen; Reynolds, Patricia; Sitaram, Pradip; Ta, Anh; Theofanos, Mary, 2005)

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Existen casos en los que resulta difícil verificar un requisito:

- **Ambiguo:** Si existen diferentes interpretaciones de un requisito. La medición de verificabilidad es difícil sobre todo en el caso de ambigüedades, donde es imposible una adecuada medición.
- Cualquier requisito equivalente a detener problemas de rendimiento, no es verificable.

Una técnica que podría ayudar en la medición de este atributo de calidad es la revisión del ERS por probadores experimentados que puedan determinar altos costos de cronograma probando sus implicaciones.

- **No ambiguo:**

Un ERS es no ambiguo sólo si cada requisito especificado tiene una única interpretación. Cuando un requisito es interpretado de forma distinta por varias personas se dice que es ambiguo y puede traer consigo desacuerdos en el grupo de trabajo de desarrollo del software.

**Medición del Atributo de Calidad No ambiguo:** (Davis, Alan; Overmyer, Scott; Jordan, Kathleen; Caruso, Joseph; Dandashi, Fatma; Dinh, Anhtuan; Kincaid, Gary; Ledebouer, Glen; Reynolds, Patricia; Sitaram, Pradip; Ta, Anh; Theofanos, Mary, 2005)

La ambigüedad puede venir relacionada con el lenguaje, ciertos lenguajes como el español son inherentemente más ambiguos que otros por su complejidad y por la riqueza de juegos de palabras.

Una manera de medir este atributo es a partir del porcentaje de requisitos interpretados de una única manera.

- **Comprensible:**

Un ERS es comprensible sólo si todos los requisitos que lo componen son entendidos por el cliente, desarrolladores y usuarios y los mismos cuentan con un mínimo de explicación. Cualquier requisito ambiguo es no comprensible.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

**Medición del Atributo de Calidad Comprensible:** (Davis, Alan; Overmyer, Scott; Jordan, Kathleen; Caruso, Joseph; Dandashi, Fatma; Dinh, Anhtuan; Kincaid, Gary; Ledebouer, Glen; Reynolds, Patricia; Sitaram, Pradip; Ta, Anh; Theofanos, Mary, 2005)

Un ERS es comprensible cuando todos los que lo leen, clientes, usuarios, desarrolladores u otros, comprenden el significado de todos los requisitos con un mínimo de explicación. Según la siguiente figura, el nivel de comprensión del ERS aumenta en un movimiento hacia el noreste.

Una manera de medir este atributo es a partir del porcentaje de requisitos comprensibles respecto al total de requisitos documentados en el ERS.

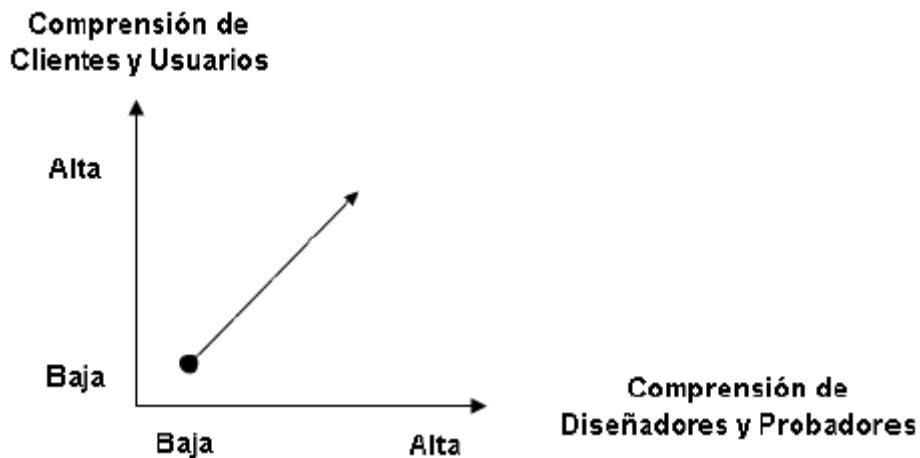


Figura 1. Medición de la comprensión de requisitos.

- **Completo:**

Un ERS es completo sólo si incluye los siguientes elementos:

- Todos los requisitos significativos que respondan a funcionalidad, desempeño, restricciones de diseño, atributos, interfaces externas o cualquier requisito externo impuesto por una especificación del sistema, deben ser reconocidos y tratados.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Se definieron las respuestas del software a todos los tipos posibles de clases de datos de entrada en todos los tipos posibles de clases de situaciones. Es importante especificar las respuestas tanto para valores de entrada válidos como inválidos.
- Existe evidencia de etiquetas y referencias completas a todas las figuras, tablas y diagramas en el ERS, así como la definición de todos los términos y unidades de medida.

Si un ERS es incompleto puede provocar insatisfacción en el cliente y que los desarrolladores tengan que hacer suposiciones incorrectas en posteriores etapas del desarrollo.

### **Medición del Atributo de Calidad Completo:** (Alexander, 2000)

Para medir la completitud de un ERS se ubican los requisitos en uno de los siguientes bloques propuestos por el modelo de Alexander:

Bloque A: Requisitos que se conocen y han sido documentados.

Bloque B: Requisitos que han sido documentados pero:

- han sido pobremente especificados.
- han sido especificados abstractamente.

Bloque C: Requisitos de los cuales se conoce su necesidad pero no se han documentado.

Bloque D: Requisitos potenciales que no se entienden lo suficientemente bien para ser documentados.

Lo ideal para que el ERS esté completo es que todos los requisitos pasen al bloque A.

La medición del atributo Completo se realiza a partir de la relación entre los requisitos del bloque A y la suma de los requisitos de todos los bloques.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- **No redundante:**

Un ERS es no redundante sólo si un mismo requisito no es indicado más de una vez. No siempre la redundancia es mala, frecuentemente puede ser usada para incrementar la legibilidad del ERS. El problema radica en que si todas las ocurrencias de un requisito redundante no son modificadas, el ERS se vuelve inconsistente.

### **Medición del Atributo de Calidad No Redundante:**

La medición del atributo No Redundante se realiza a partir de la relación entre los requisitos no redundantes y el total de requisitos especificados en el ERS.

- **Correcto:**

Un ERS es correcto sólo si cada requisito especificado es un requisito que el software debe cumplir y no expresa necesidades que realmente no existen o no son necesarias. Además el ERS debe cumplir con las reglas o restricciones del negocio y no deben existir contradicciones entre los requisitos.

### **Medición del Atributo de Calidad Correcto:**

La medición del atributo Correcto se realiza a partir de la relación entre los requisitos correctos y el total de requisitos especificados en el ERS.

- **Posible de Tracear:**

Un ERS es posible de tracear si el origen de cada uno de sus requisitos es claro y si facilita la referencia de cada requisito en el desarrollo futuro o mejora de la documentación.

Se recomiendan los siguientes tipos de trazabilidad:

- Trazabilidad hacia atrás (a estados previos del desarrollo). El requisito tiene referencias explícitas a sus fuentes en documentos anteriores.



## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Trazabilidad hacia adelante (a todos los documentos derivados del ERS). Depende de que cada requisito en el ERS tenga un nombre único o número de referencia. Es particularmente importante cuando el software entra en operación y mantenimiento. Cuando el código y los documentos de diseño son modificados, es esencial contar con la capacidad para conocer el conjunto completo de requisitos que pueden ser afectados por esas modificaciones.

- **Preciso:**

Un ERS es preciso sólo si son usadas cantidades numéricas siempre y cuando sea posible y se usan niveles apropiados de precisión para todas las cantidades numéricas.

- **Modificable:**

Un ERS es modificable sólo si su estructura y estilo son tales que, cualquier cambio a los requisitos pueden ser hechos con facilidad, de manera completa y consistentemente sin perder la estructura y el estilo. La modificabilidad generalmente requiere que un requisito:

- Tenga una organización coherente y fácil de usar con una tabla de contenido, un índice y referencias cruzadas explícitas.
- No sea redundante.
- Expresa cada requisito de manera separada, en vez de hacerlo mezclado con otros requisitos.

- **Consistente:**

Un ERS es consistente sólo si no se contradice a sí mismo es decir, si ningún subconjunto de los requisitos que describe se contradice o entra en conflicto con otros requisitos.

- **Jerarquizado de acuerdo a la importancia y/o estabilidad:**

Un ERS está calificado de acuerdo a la importancia y/o estabilidad si cada requisito tiene un identificador que indique la importancia o estabilidad del requisito.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

**Método Delphi:** (López, Alfaro; Mauricio, Ricardo; Rivera, Castro; Danilo, Víctor; Lainez, Romero; Osmar, Balmore, 2005)

**Características de pronóstico Delphi:** (PRONOSTICO DELPHI, Universidad del Salvador. , 2005.)

1. Anonimato: Cada especialista desconoce la identidad de los demás integrantes del panel. No debería haber contacto físico entre los participantes, pero el administrador de la encuesta sí puede identificar a cada participante y sus respuestas.
2. Iteración: Se pueden manejar tantas rondas como sean necesarias. Se extraen de los cuestionarios aquellos segmentos de información que son relevantes como insumos y se presentan al panel en la ronda posterior. La interacción de argumentos impersonales a favor o en contra de cada pronóstico contribuye a formar estados de consenso que hacen más transparentes los escenarios emergentes. Tanto las posturas minoritarias como las mayoritarias tienen presencia en el panel.
3. Heterogeneidad: Pueden participar especialistas de diferentes ramas de actividades sobre las mismas bases o "reglas de juego".
4. Retroalimentación Controlada: Los resultados totales de la ronda previa no son entregados a los participantes, sólo una parte seleccionada de la información circula.
5. Resultados estadísticos: La respuesta del grupo puede ser presentada estadísticamente (promedios y grado de dispersión).

**El método Delphi puede ser usado para:**

- Planear en grupos o mejorar la comunicación.
- Hacer predicciones sobre cómo va a ser o cómo debe ser un evento.
- Identificar necesidades y solucionar problemas.
- Identificar y estudiar un rango de posibles alternativas o estrategias para solucionar un problema.
- Encontrar información para generar consenso en un grupo de especialistas o expertos.
- Integrar juicios sobre un tema interdisciplinario.
- Investigar, como una variante de alguna encuesta.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- En Producción:
  - En el costo de materia prima.
  - En el costo de mano de obra.
  - En la disponibilidad de materia prima.
  - En la disponibilidad de mano de obra.
  - En los requerimientos de mantenimiento.
  - En la capacidad disponible de la planta para la producción.
  
- En el área de la Mercadotecnia:
  - En el tamaño del mercado.
  - En la participación en el mercado.
  - En la tendencia de precios.
  - En el desarrollo de nuevos productos.

En general puede ser utilizado en muchas ramas, empresariales, pronósticos del tiempo, pronósticos de enfermedades, en cualquier actividad en la cual se necesite un pronóstico se logra analizar y responder una pregunta sobre la variable en cuestión.

### **Pasos para aplicar el Método Delphi:**

Se pueden seguir diez pasos para la aplicación del método:

1. Definición del problema: En algunos casos esto puede ser precisamente el objeto de la utilización del método.
2. Formación de un grupo que aborde un tema específico: Usualmente los miembros del grupo son especialistas o expertos en el tema que se pretende estudiar. Dentro de ese grupo se pueden conformar uno o varios subgrupos (panel) para que participen en el proceso. Uno de los subgrupos puede dedicarse al análisis de los resultados de cada ronda de preguntas durante el proceso.
3. Diseño del cuestionario que se utilizará en la primera ronda de preguntas.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

4. Prueba del primer cuestionario: Muchas veces la estructura o formulación de las preguntas puede que no conduzca a las respuestas apropiadas. Se debe evitar la ambigüedad y la vaguedad en la redacción de las preguntas. Deben ser muy precisas, puntuales y que no sean sujetas a interpretación.
5. Entrega del cuestionario a los panelistas.
6. Análisis de las respuestas de la primera ronda de preguntas.
7. Preparación de la segunda ronda de preguntas y aprovechamiento de la primera ronda para perfeccionar las preguntas o quizás, si es necesario, probar otra vez las preguntas, como en el paso anterior.
8. Entrega del segundo cuestionario a los panelistas.
9. Análisis de las respuestas de la segunda ronda de preguntas: (los pasos del 5 al 9 deberán repetirse en forma iterativa hasta que se llegue a un consenso o se alcance una cierta estabilidad en las respuestas).
10. Preparación de un informe por parte del equipo que analiza los resultados para presentar las conclusiones del ejercicio y hacer oficial la aceptación del documento con el cumplimiento de las reglas en cuestión.

En general el método es muy útil para analizar y responder una pregunta sobre una sola variable. Cuando se tienen que analizar múltiples variables, los resultados pueden no ser los óptimos. Sin embargo, el hecho de no servir para múltiples variables, no reduce su utilidad. En muchos casos se cuentan con modelos complejos y variados, el gran problema es encontrar los datos para alimentar el modelo. El método se puede utilizar para analizar, una a una, las variables del modelo complejo y utilizar los resultados como datos de entrada de ese complejo modelo.

## CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES

### **CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES.**

Para dar solución a los problemas que se presentan en los casos en que en los proyectos no se realiza una validación correcta, se propone a continuación una propuesta de validación para aumentar la calidad de las funcionalidades en la construcción de un software.

#### **Propuesta de validación:**

El proceso de validación que se propone consta de dos etapas fundamentales:

- 1. Verificación:** La verificación se realizará internamente en el proyecto por el equipo de calidad a través de listas de chequeo, para posteriormente realizar la validación con los especialistas funcionales, además de la utilización de la métrica para validar si el documento ERS cumple con el atributo de calidad “Verificable”.
- 2. Validación:** La validación de los atributos de calidad se realizará con especialistas funcionales a través de una adaptación del método Delphi y la aplicación de métricas para medir el cumplimiento de los atributos de calidad del ERS.

#### **Verificación:**

#### **Lista de Chequeo:**

A continuación se muestra la lista de chequeo a ser utilizada por el equipo de calidad, donde las preguntas 1 y 2 responden a la estructura del documento; 3 y 4 a la semántica del documento; 5 y 6 responden al atributo de calidad “Comprensible”; 7 a “No ambiguo”; 8 y 9 a “Precisión”; 10 y 11 a “No redundante”; 12 a “Verificable”; 13 y 14 a “Completo”; 15 y 16 a Posible de tracear”; 17, 18 y 19 a “Correcto” y las restantes a aspectos generales del documento.

1. ¿Está el documento acorde con a la plantilla estándar del proyecto o del expediente de proyecto?
2. ¿Está identificado el origen del requisito (persona, norma, documento)?
3. ¿Contiene las secciones obligatorias definidas en el expediente de proyecto?

## CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES

4. ¿Se han identificado errores ortográficos?
5. ¿Están todos los requisitos redactados de forma simple y clara?
6. ¿Deberían especificarse algunos requisitos con más detalle?
7. ¿Tienen todos los requisitos una única interpretación? ¿Cuáles no?
8. ¿Los requisitos que son medibles cuantitativamente están delimitados en términos cuantitativos?
9. ¿Se utilizan en todos los requisitos delimitados en términos cuantitativos niveles apropiados de precisión para todas las cantidades numéricas?
10. ¿Existe redundancia en la definición de los requisitos? ¿En cuáles?
11. ¿Aparecen requisitos en más de un lugar del ERS? ¿Cuáles?
12. ¿Todos los requisitos son verificables?
13. ¿Se pueden probar todos los requisitos una vez implementados?
14. ¿Todos los requisitos documentados son correctos? ¿Cuáles no?
15. ¿Se puede trazar cada requisito al origen en el entorno del problema: caso de uso o proceso del negocio? ¿Cuáles no? (Para el caso de proyectos que realicen modelación de negocio)
16. ¿Existe referencia de la dependencia del requisito con otros requisitos?
17. ¿Existen requisitos que se contradicen con otros requisitos?
18. ¿Algún requisito incumple restricciones definidas (reglas del negocio)?
19. ¿Se pueden localizar todos los requisitos en el conjunto de objetivos del sistema/producto?
20. ¿Todos los requisitos pueden ser implementados?
21. ¿Se han definido los límites del sistema?
22. ¿Se han identificado los requisitos de software y de hardware?
23. ¿Se han identificado las restricciones de diseño e implementación?
24. ¿Se han identificado las restricciones de interfaz externa?
25. ¿Se han identificado los requisitos de soporte y usabilidad?
26. ¿Se han identificado los requisitos de seguridad (confidencialidad, integridad, disponibilidad)?
27. ¿Todos los requisitos identificados se centran en lo que el sistema debe hacer y no en cómo el sistema debe hacerlo?

### **Medir Atributo de Calidad Verificable:**

Para medir este atributo se puede usar la siguiente ecuación:

## CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES

$$Q_1 = \frac{n_r}{n_r + \sum_i c(r_i) + \sum_i t(r_i)}$$

Donde:

**c(ri)** = costo necesario para verificar la presencia del requisito.

**t(ri)** = tiempo necesario para verificar la presencia del requisito.

**n<sub>r</sub>** = número total de requisitos documentados en el ERS.

El valor resultante oscila en el rango entre 0 (pobre verificabilidad) y 1 (ERS totalmente verificable). Debido a que este atributo es muy crítico para el éxito de un proyecto se recomienda que  $Q_1 = 1$ .

## CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES

### Validación.

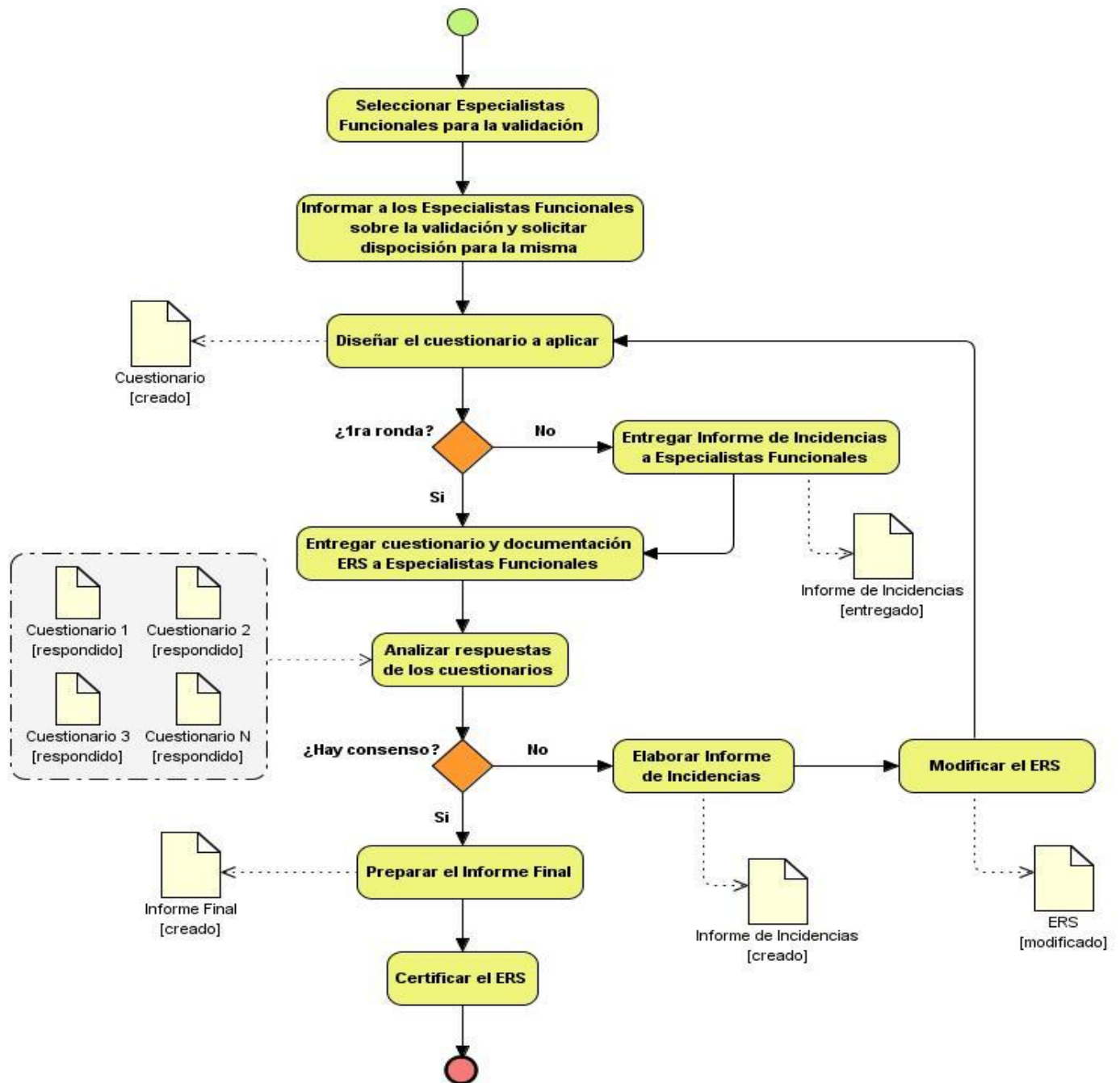


Figura 2. Validación con Especialistas Funcionales.



## CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES

### **Actividad: Seleccionar Especialistas Funcionales para la validación:**

Para la selección de los Especialistas Funcionales se tendrán en cuenta los siguientes criterios:

<b>Criterios de Aceptación</b>	<b>Nivel de Importancia</b>
Conocimiento del mercado	5
Conocimiento del negocio	5
Disponibilidad de tiempo	4
Habilidades de comunicación	3
Interacción con el sistema	3
Grado de autoridad	5
Compromiso con el proyecto	4

**Tabla 1. Criterios de aceptación y nivel de importancia.**

Se seleccionan a los Especialistas Funcionales que mayor puntuación obtengan en total, donde la puntuación de cada especialista se dará por:

$$P = \sum_{i=1}^n c_i * n_i$$

Donde:

**P**=Puntuación del Especialista.

**c<sub>i</sub>** =Criterio de aceptación "i".

**n<sub>i</sub>** =Nivel de importancia del criterio "i".

## CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES

### **Actividad: Informar a los Especialistas Funcionales sobre la validación y solicitar disposición para la misma:**

Se debe consultar con los Especialistas Funcionales seleccionados en el paso anterior la necesidad de validar los requisitos con ellos, explicando cómo se desarrollará la validación y preguntar su disposición o no a colaborar con la misma. Aquellos que estén de acuerdo serán los que participarán en el proceso de validación.

### **Actividad: Diseñar el cuestionario a aplicar:**

Se debe aplicar un cuestionario a ser respondido por cada uno de los especialistas anónimamente, de manera que el modelador del ejercicio de validación, el analista, sea el único que conozca la fuente de las respuestas del cuestionario y el resto de los especialistas no se sienta cohibido o presionado a responder lo que realmente opina.

La validación que se propone puede llevar “n” rondas hasta tanto no se hayan resuelto todos los problemas detectados que impidan la certificación del ERS.

Se propone el siguiente cuestionario a aplicar, aunque puede ser modificado según necesidades particulares.

Las preguntas 1, 2, 3, 4 y 5 responden al atributo de calidad “Correcto”; 6 y 7 a “Comprensible”; 8 y 9 a “No ambiguo”; 10 y 11 a “Preciso”; 12 y 13 a “Completo”; 14 y 15 a “Posible de Tracear” y el 16 a “Redundancia”.

1. ¿Están correctos todos los requisitos?
2. ¿Algún requisito incumple restricciones definidas por el negocio? ¿Cuáles?
3. ¿Están correctamente definidos los objetivos del sistema? ¿Cuáles no? ¿Cuáles faltan?
4. ¿Se pueden localizar todos los requisitos en el conjunto de objetivos del sistema/producto? ¿Cuáles no?
5. ¿Existen requisitos que se contradicen con otros requisitos? ¿Cuáles?
6. ¿Están todos los requisitos redactados de forma simple y clara? ¿Cuáles no?

## CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES

7. ¿Deberían especificarse algunos requisitos con más detalle? ¿Cuáles?
8. ¿Considera todos los requisitos libres de ambigüedad? ¿Cuáles no?
9. Realice una justificación breve de lo que ha interpretado en cada uno de los requisitos.
10. ¿Todos los requisitos están delimitados en términos cuantitativos (en caso de ser posible)? ¿Cuáles no?
11. ¿Se utilizan en todos los requisitos delimitados en términos cuantitativos niveles apropiados de precisión para todas las cantidades numéricas? ¿En cuáles no?
12. ¿Todos los requisitos documentados son correctos? ¿Cuáles no?
13. ¿Se han identificado todas las funciones que el usuario debe hacer? ¿Cuáles faltan?
14. ¿Las trazas de los requisitos a sus fuentes de origen, casos de uso o procesos del negocio son correctas? ¿Cuáles no? (Para el caso de proyectos que realicen modelación de negocio)
15. ¿Las trazas de dependencia definidas entre requisitos son correctas? ¿Cuáles no?
16. ¿Existe redundancia en la definición de los requisitos? ¿En cuáles?

Independientemente de la propuesta de encuesta anterior que puede ser útil sobre todo en la primera ronda, cuando se está aplicando la segunda o enésima ronda se deben tener en cuenta los resultados obtenidos en la anterior encuesta y se puede agregar alguna otra pregunta que pueda ser útil.

### **Actividad: Entregar cuestionario y documentación ERS a Especialistas Funcionales:**

Se hace entrega del cuestionario confeccionado a cada uno de los especialistas, así como de la documentación de ERS a ser revisada para validar.

### **Actividad: Analizar respuestas de los cuestionarios:**

Una vez respondidos los cuestionarios por los especialistas se realiza un análisis de los mismos para ver los resultados arrojados y según el atributo de calidad al cual corresponde cada pregunta, el modelador del ejercicio de validación realiza las mediciones pertinentes, teniendo en cuenta además las métricas definidas para los atributos de calidad descritas en la sección posterior.

## CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES

### **Actividad: Elaborar Informe de Incidencias:**

Si realizado el análisis de las respuestas a los cuestionarios existen problemas en algunos de los atributos de calidad medidos y no existe un consenso, se elabora un Informe de Incidencias que recoge los problemas detectados en la ronda y sugerencias para su resolución. De igual manera, se describen las secciones que fueron modificadas en el ERS para la próxima ronda y su justificación.

El formato del Informe de Incidencias debe recoger la relación: Requisito-Defecto-Sugerencia.

### **Actividad: Modificar el ERS:**

Si una vez realizado el análisis de las respuestas a los cuestionarios existen problemas en algunos de los atributos de calidad medidos y no existe un consenso, pueden realizarse modificaciones en el ERS que son recogidas con su justificación en el Informe de Incidencias para ser expuesto con los arreglos en la próxima ronda.

### **Actividad: Entregar Informe de Incidencias a Especialistas Funcionales:**

Cuando se va a realizar una segunda, tercera o enésima ronda del ejercicio se debe entregar a los especialistas además del ERS, el Informe de Incidencias de la ronda anterior.

### **Actividad: Preparar el Informe Final:**

Una vez que exista un consenso entre todos los especialistas y el resultado de la medición de los atributos de calidad sea factible, se procede a preparar el Informe Final del ejercicio de validación con un resumen histórico de los aspectos esenciales, para certificar el ERS.

### **Actividad: Certificar el ERS:**

La certificación del ERS consiste en la formalidad de aceptación del ERS por parte del cliente del producto a través de una firma del documento.

## CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES

### Medición de los Atributos de Calidad:

- **No ambiguo:**

$$Q_2 = \frac{n_{ui}}{n_r}$$

Donde:

$n_{ui}$  =Número de requisitos con una única interpretación.

$n_r$  =Número total de requisitos documentados en el ERS.

Cuando el valor resultante se aproxima a 0, se interpreta que cada requisito tiene múltiples interpretaciones; cuando el valor se aproxima a 1, se interpreta que cada requisito tiene una única interpretación. Debido a que este atributo es muy crítico para el éxito de un proyecto se recomienda que  $Q_2 = 1$ .

- **Completo:**

Siguiendo el modelo podemos medir la completitud de requisitos como el porcentaje de requisitos en el ERS que han sido correctamente entendidos:

$$Q_3 = \frac{n_A}{n_r}$$

Donde:

$n_A$  =Número de requisitos del Bloque A: Requisitos que se conocen y han sido documentados.

$n_r$  =Número total de requisitos documentados en el ERS.

## CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES

$$n_r = n_A + n_B.$$

$n_B$  = Número de requisitos del Bloque B: Requisitos que han sido documentados pero: han sido pobremente especificados y/o han sido especificados abstractamente.

El valor resultante oscila en el rango entre 0 (totalmente incompleto) y 1 (completitud). Debido a la importancia de este atributo para el proyecto se recomienda que  $Q_3 \geq 0,7$ .

Alternativamente el atributo Completo puede ser medido de la siguiente manera:

$$Q_3 = \frac{n_r}{n_A + n_B + n_C + n_D}$$

Donde:

$n_A$  =Número de requisitos del Bloque A: Requisitos que se conocen y han sido documentados.

$n_B$  =Número de requisitos del Bloque B: Requisitos que han sido documentados pero: han sido pobremente especificados y/o han sido especificados abstractamente.

$n_C$  =Número de requisitos del Bloque C: Requisitos de los cuales se conoce su necesidad pero no se han documentado.

$n_D$  =Número de requisitos del Bloque D: Requisitos potenciales que no se entienden lo suficientemente bien para ser documentados.

$n_r$  =Número total de requisitos documentados en el ERS.

## CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES

- **Comprensible:**

$$Q_4 = \frac{n_{ur}}{n_r}$$

Donde:

$n_{ur}$  =Número de requisitos que todos los revisores del ERS entienden.

$n_r$  =Número total de requisitos documentados en el ERS.

El valor resultante oscila en el rango entre 0 (ningún requisito comprendido) y 1 (todos los requisitos comprendidos).

Debido a que este atributo es muy crítico para el éxito de un proyecto se recomienda que  $Q_4 = 1$ .

- **No Redundante:**

Para medir la redundancia en un ERS tenemos la ecuación:

$$Q_5 = \frac{n_u}{n_f}$$

Donde:

$Q_5$ , Representa el porcentaje de funciones únicas no repetidas.

$n_f$  =Número de funciones especificadas.

$n_u$  =Número de funciones únicas especificadas.

## CAPÍTULO 2: VALIDACIÓN DE REQUISITOS CON ESPECIALISTAS FUNCIONALES

El valor resultante oscila en el rango entre 0 (completamente redundante) y 1 (no redundante).

- **Correcto:**

Este atributo puede ser medido de la siguiente manera:

$$Q_6 = \frac{n_c}{n_r}$$

Donde:

$n_c$  =Número de requisitos correctos.

$n_r$  =Número total de requisitos documentados en el ERS.



## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

### **CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA.**

Para la validación de la presente investigación, se tomó como referencia al proyecto Motor de Clasificación Inteligente de Contenidos (MOCIC) de la facultad # 10. Se tuvieron en cuenta los resultados de la validación de requisitos que se aplica en el proyecto, luego se aplicó la de validación de requisitos propuesta en el presente trabajo y se realiza una comparación entre los resultados obtenidos en cada validación de requisitos aplicada.

#### **Proyecto MOCIC:**

El proyecto MOCIC, consta de 5 subsistemas, uno de ellos el Subsistema de Imágenes, la propuesta se aplicó en de dicho subsistema, que se encuentra en la fase de Elaboración.

**Polo productivo:** Soluciones Informáticas para Internet.

**Clasificación del proyecto:** Investigación y Desarrollo.

#### **¿Qué es MOCIC?**

Está enfocado en el producto “Servicio de Listas de URLs Clasificadas”, que consta de un conjunto de subsistemas que mayoritariamente estarán dotados de inteligencia artificial, los cuales de forma unida, permitirán automatizar el proceso de clasificación de contenidos y será utilizado para categorizar URLs provenientes de Internet.

**Los subsistemas identificados con sus respectivos módulos son:**

#### **Subsistema de Imágenes.**

- Módulo de Detección de Rostros en Imágenes.
- Módulo de Reconocimiento de Objetos en Imágenes.
- Módulo de Clasificación de Imágenes Pornográficas.

#### **Subsistema de Almacenamiento.**

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

- Módulo de Almacenamiento.
- Subsistema de Texto.
- Módulo de Clasificación de Textos.
- Módulo de Pre-procesamiento de Contenido.
- Módulo de Identificación de Idiomas.

### **Subsistema de control.**

- Módulo de Control.
- Módulo de Administración Web.

### **Sub-Sistema Decisor.**

- Módulo de Decisión.

### **Misión:**

Desarrollar una herramienta informática que permita, de forma automatizada e inteligente, clasificar diferentes tipos de contenidos, para crear listas de URLs categorizadas, lo cual permitirá:

- Actualizar la base de datos del proyecto Filtrado de Paquetes por Contenido (FILPACON).
- Obtener/Descargar (en línea) listas de URLs categorizadas, para clientes adscritos al servicio.

### **Visión:**

Ser en la UCI, el proyecto rector en materia de clasificación inteligente de contenidos, principalmente para aplicaciones vinculadas con Internet.

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

### **Proyecto FILPACON:**

FILPACON es un sistema de filtrado de contenidos para Internet bajo licencia GPL que tiene como objetivo evitar el acceso por parte de los usuarios a sitios con contenidos inadecuados. Este sistema está diseñado para interactuar con un servidor Squid proxy; cuenta fundamentalmente con una base de datos de URL clasificadas que rigen el acceso de los usuarios; posee además una interfaz de usuario que facilita las tareas de administración. Aunque la primera versión resolvió las necesidades para las que fue diseñado, en la actualidad muchas de sus características no solucionan los problemas existentes. El desarrollo de la versión 2.0 constituyó un reto mayor por la necesidad de una arquitectura base capaz de dar soporte para futuras versiones y al mismo tiempo resolver las dificultades detectadas en la versión anterior. Algunos conceptos como; política de navegación, administración dinámica por roles y redirector en forma de plugins forman parte del engranaje que permiten que esta versión tenga un alcance mucho mayor. Aunque las perspectivas son prometedoras quedan características por incluir que permitirán analizar el tráfico entrante de la red obtener estadísticas del funcionamiento del sistema y analizar el contenido de manera inteligente.

### **Aplicación de la Estrategia de Validación de requisitos del Proyecto MOCIC:**

En el proyecto MOCIC la validación realizada por la analista del subsistema fue de la siguiente manera:

#### **Paso 1:**

Se mostraron al cliente los requisitos documentados en el ERS. Ver Anexo # 1 ERS Validación1 v1.0.

#### **Paso 2:**

El cliente analizó los requisitos especificados y arribó a las siguientes conclusiones:

- Se detectó un requisito que no era realmente importante, llegando a ser innecesario pues no representa una funcionalidad necesaria para el sistema a construir. Ver Anexo # 1: ERS Validación1 v1.0, requisito # 7.

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

- Se detectaron 5 faltas de ortografía. Ver faltas de ortografía señaladas en Anexo # 1: ERS Validación1 v1.0.
- Se detectó un requisito que no se encontraba documentado y era necesario para el sistema.

### **Paso 3:**

Se asignaron los cambios a realizar a la analista involucrada, la cual resolvió las inconformidades.

**Cambios realizados en el ERS:** Ver Anexo # 2: ERS Validación1 v1.1.

- Se eliminó el requisito # 7.
- Se corrigieron las faltas de ortografía.
- Se agregó un requisito que aborde la segmentación de una imagen por el color de la piel, necesario para el sistema.

### **Paso 4:**

Se mostraron al cliente los requisitos documentados en el ERS. Ver Anexo # 2: ERS Validación1 v1.1.

### **Paso 5:**

El cliente analizó los requisitos especificados y arribó a las siguientes conclusiones:

Las inconformidades detectadas fueron corregidas satisfactoriamente y los requisitos especificados están correctos. Existe conformidad con los requisitos documentados.

### **Paso 6:**

Se certifica el ERS, lo cual consiste en la formalidad de aceptación del ERS por parte del cliente del producto a través de una firma del documento.

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

### **Se detectaron como deficiencias de la validación:**

- No se aplica un procedimiento para seleccionar los especialistas que realmente deben participar en la validación.
- A pesar de existir más de un especialista asociado a la definición de requisitos del módulo, la validación se realiza sólo con un especialista, obviando los criterios y necesidades del resto de los especialistas.
- Pueden existir requisitos con cierto nivel de ambigüedad no identificado, producto de la validación realizada con sólo un especialista donde se involucran otros.
- No se chequea internamente en el proyecto cuán correcto está el ERS antes de validarlo con el cliente.

### **Aplicación de la Estrategia de Validación de Requisitos con Especialistas Funcionales:**

Para la validación del presente trabajo se obtuvieron los siguientes resultados al aplicar esta estrategia:

#### **Verificación:**

#### **Aplicar Lista de Chequeo:**

Se realizó la Verificación teniendo en cuenta la Lista de Chequeo propuesta en el Capítulo 2, obteniéndose los siguientes resultados.

- Se detectó 1 requisito con dificultad de comprensión, pues no era lo suficientemente claro y preciso. Ver Anexo # 3: ERS Validación2 v1.0 requisito # 10.
- Se detectaron 5 faltas de ortografía. Ver faltas de ortografía señaladas en Anexo # 3: ERS v1.0.
- Se detectó redundancia en la definición de 3 requisitos. Ver Anexo # 3: ERS Validación2 v1.0, requisitos # 7, 8 y 9.

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

- Se detectó un requisito que no representa una funcionalidad necesaria para el sistema a construir, pues esta funcionalidad es recogida en otros 2 requisitos. Ver Anexo # 3: ERS Validación2 v1.0, requisito # 7.

### **Medir Atributo de Calidad Verificable:**

Se tiene en cuenta además la métrica definida para el atributo de calidad Verificable descrita en el capítulo anterior, arrojando el siguiente resultado:

Para aplicar esta métrica se necesita de personas que cuenten con experiencia, probadores experimentados que puedan determinar altos costos de cronograma probando sus implicaciones. Se contactó con el Líder de Gestión y Jefe de Calidad del proyecto por la experiencia de los mismos para lograr una aproximación en cuanto al costo y tiempo de verificación de los requisitos, obteniéndose la siguiente tabla.

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13
Costo	0	0	0	0	0	0	0	0	0	0	0	0	0
Tiempo	0	0	0	0	0	0	0	0	0	0	0	0	0

**Tabla 2. Costo y tiempo de verificación de los requisitos.**

Para la asignación de valores se determinó que los requisitos documentados requieren de un bajo costo y tiempo para su verificación, por tanto se les asigna el valor 0 de costo y tiempo necesarios para verificar la presencia del atributo, donde:

- Valor de bajo costo y/o tiempo: 0.
- Valor de mediano costo y/o tiempo: 0,5.
- Valor de alto costo y/o tiempo: 1.

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

$$Q_1 = 11/11 + (0+0+0+0+0+0+0+0 +0+0+0+0+0) + (0+0+0+0+0+0+0+0+0+0+0+0+0) = 1.$$

El valor resultante oscila en el rango entre 0 (pobre verificabilidad) y 1 (ERS totalmente verificable). Como  $Q_1$  es 1 y se está midiendo un atributo crítico para el sistema, el procedimiento de validación logró obtener como resultado un ERS verificable.

Se procedió a realizar los cambios en el ERS teniendo en cuenta los errores detectados por el equipo de desarrollo del proyecto al aplicar la lista de chequeo propuesta por el trabajo investigativo en la parte, la Verificación. Ver Anexo # 4: ERS Validación2 v1.1.

### **Cambios realizados en el ERS:**

- Se corrigieron las faltas de ortografía. Ver Anexo # 4: ERS Validación2 v1.1.
- Se eliminó el requisito # 7 para eliminar la redundancia detectada en los requisitos # 7, 8 y 9, causada porque que la funcionalidad de este requisito se encuentra descrita en los requisitos # 8 y 9. El requisito # 8 pasa a ser el # 7 y el requisito # 9 pasa a ser el # 8. Ver Anexo # 4: ERS Validación2 v1.1.
- Se especificó con un mayor nivel de detalle la descripción del requisito encontrado con dificultad de comprensión. Ver Anexo # 4: ERS Validación2 v1.1, requisito # 9.

### **Se realizó la Validación con los Especialistas Funcionales posteriormente:**

Se le aplicó esta validación al ERS resultante luego de realizar los cambios de los errores detectados a los requisitos, al aplicar la lista de chequeo en la etapa de Verificación.

El objetivo consiste en demostrar que existen deficiencias que permanecen en etapas de desarrollo posteriores a la Validación de los Requisitos, las cuales no son detectadas durante las validaciones que se desarrollan actualmente en muchos proyectos. Por consiguiente se tomó como referencia dicho subsistema y proyecto, atendiendo a que por la parte del cliente interactúan varios Especialistas

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Funcionales que presentan cierta divergencia de criterios, lo cual suele dificultar el trabajo en el momento de arribar a un consenso para la definición del subsistema en cuestión.

### **Actividad: Selección de los Especialistas Funcionales:**

Se realizó un análisis de los posibles especialistas que podrían realizar la validación de los requisitos, los cuales fueron categorizados para la selección de los más preparados.

Nº	Nombre y Apellidos	Especialidad
1	Luis Enrique Sánchez Arce.	Ingeniero en Ciencias Informáticas.
2	Kiuver Kaddiel Ibáñez Castro.	Ingeniero en Ciencias Informáticas.
3	Mailen Proenza Guerra.	Ingeniera en Ciencias Informáticas.
4	Karel A. Verdecia Ortiz.	Licenciado en Ciencias de la Computación.

**Tabla 3. Nombre de los Especialistas Funcionales.**

Se realiza la selección utilizando los criterios de aceptación que se encuentran en la Tabla 1: Criterios de aceptación y nivel de importancia del Capítulo 2. Quedando como resultado las siguientes puntuaciones:

Nº	Conocimiento del mercado	Conocimiento del negocio	Disponibilidad de tiempo	Habilidades de comunicación	Interacción con el proyecto	Autoridad	Compromiso con el proyecto
1	5	4	4	5	5	5	4
2	5	5	4	5	4	5	4
3	3	3	3	3	2	2	4
4	3	3	4	2	3	3	4

**Tabla 4. Criterios de aceptación y nivel de importancia aplicados a los Especialistas Funcionales.**

### **Actividad: Aplicar la fórmula para obtener la puntuación de cada especialista obteniendo:**

Para:  $P_1 = 5*5+5*4+4*4+3*5+3*5+5*5+4*4=122$ .

Para:  $P_2 = 5*5+5*5+4*4+3*5+3*4+5*5+4*4=134$ .



## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Para:  $P_3 = 5*3+5*3+4*3+3*3+3*2+5*2+4*4=83$ .

Para:  $P_4 = 5*3+5*3+4*4+3*2+3*3+5*3+4*4=92$ .

**Siendo escogidos los siguientes Especialistas Funcionales por obtener las mayores puntuaciones:**

Ing. Luis Enrique Sánchez Arce.

Ing. Kiuver Kaddiel Ibáñez Castro.

**Actividad: Diseñar el cuestionario a aplicar:**

En esta actividad se aplicó el cuestionario descrito anteriormente en el Capítulo 2.

**Actividad: Entregar cuestionario y documentación ERS a los Especialistas Funcionales:**

Se hace entrega del cuestionario confeccionado a cada uno de los especialistas, así como de la documentación ERS a ser revisada para validar. Ver ERS en el Anexo # 4: ERS Validación2 v1.1.

**Actividad: Analizar respuestas de los cuestionarios:**

Las respuestas de los cuestionarios emitidas por los especialistas se encuentran en, ver Anexo # 6, Respuesta 1 del Cuestionario y Anexo # 7, Respuesta 2 del Cuestionario. Aclarar que se determinó sólo mostrar la breve descripción de los requisitos detectados como ambiguos, dado que los Especialistas Funcionales contaban con un amplio dominio de la descripción de los restantes requisitos, dichas descripciones son las mostradas en los anexos.

Se encontraron varias inconsistencias y problemas durante la validación de requisitos con los Especialistas Funcionales, entre estos se encuentran:

Completo:

- Se detectó un requisito que no se encontraba documentado y era necesario para el sistema.

Ambiguo:

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

- Se detectaron tres requisitos ambiguos, interpretados de diferentes maneras por los Especialistas Funcionales que participaron en la validación. Ver Anexo # 4: ERS Validación2 v1.1 requisitos # 2, 3 y 4.

Comprensible:

- Se detectó un requisito con dificultad de comprensión, pues no era lo suficientemente claro y preciso. Ver Anexo # 4: ERS Validación2 v1.1 requisito # 10.
- Siempre que exista al menos un requisito ambiguo, el ERS no cumple con este atributo de calidad.

### **Medición de los atributos de calidad:**

Se tuvo en cuenta además que las métricas definidas para los atributos de calidad descritas en el capítulo anterior, teniendo en cuenta los resultados descritos anteriormente luego de aplicado el cuestionario en su primera ronda, arrojando los siguientes resultados:

- No ambiguo:

$$Q_2 = 7/10 = 0,7.$$

Como  $Q_2$  es 0,7 y se está midiendo un atributo crítico para el sistema, se recomienda continuar el procedimiento de validación hasta tanto se logre obtener que  $Q_2$  de como resultado 1.

- Completo:

Bloque A: 10 requisitos.

Requisitos que conocemos y han sido documentados.

Bloque B: 4 requisitos.

Requisitos que han sido documentados pero:

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

- han sido pobremente especificados,
- han sido especificados abstractamente,

Bloque C: 1 requisito.

Requisitos que sabemos son necesarios pero no hemos documentado.

Bloque D: 0 requisitos.

Requisitos potenciales que no entendemos lo suficientemente bien para ser documentados.

$$Q_3 = 10/10 + 4 + 1 + 0 = 0,6.$$

Como  $Q_3$ , aplicando la variante de la fórmula más completa obtiene un resultado de 0,6. Se está midiendo un atributo crítico para el sistema, por lo que se recomienda continuar el procedimiento de validación hasta tanto se logre obtener que  $Q_3$  de como resultado 1.

- Comprensible:

$$Q_4 = 6/10 = 0,6.$$

Se incluyen los 3 requisitos ambiguos y el no comprensible, pues ningún requisito que sea ambiguo es comprensible, para un total de 4 requisitos no comprensibles.

Como  $Q_4$  es 0,6 y se está midiendo un atributo crítico para el sistema, se recomienda continuar el procedimiento de validación hasta tanto se logre obtener que  $Q_4$  de como resultado 1.

- No redundante:

$$Q_5 = 10/10 = 1.$$

Como  $Q_5$  es 1, se cumple en un 100 % con el atributo de calidad No redundante.

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

- Correcto:

$$Q_6 = 10/10 = 1.$$

Como  $Q_6$  es 1, se cumple en un 100 % con el atributo de calidad Correcto.

### **Actividad: Elaborar Informe de Incidencias:**

Se determinó que existen problemas en algunos de los atributos de calidad medidos, a partir de la aplicación de las métricas anteriormente calculadas, donde se manifestó que los atributos críticos: no ambiguo, comprensible y completo, no obtuvieron un resultado satisfactorio y se recomienda por su que en la medición de estos atributos se obtenga como resultado 1 pues no existe un consenso. Se elaboró un Informe de Incidencias que recoge todos los problemas detectados en la ronda y sugerencias para su resolución.

<b>Requisitos</b>	<b>Defecto</b>	<b>Sugerencia</b>
2, 3, 4	Ambigüedad.	Especificar con un mayor nivel de detalle la descripción del requisito.
11	No comprensible.	Especificar con un mayor nivel de detalle la descripción del requisito.
	Ausencia de un requisito necesario y no documentado.	Agregar un requisito que aborde la segmentación de una imagen por el color de la piel.

**Tabla 5. Sección de Informe de Incidencias.**

### **Actividad: Modificar el ERS:**

Al realizar el análisis de las respuestas a los cuestionarios se determinó que existen problemas en algunos de los atributos de calidad medidos y no existe un consenso, por lo que se realizaron modificaciones en el ERS que son recogidas con su justificación en el Informe de Incidencias para ser expuesto con los arreglos en la próxima ronda. Ver Anexo # 5: ERS Validación2 v1.2.

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

### **Cambios realizados en el ERS:**

- Se agregó un requisito que aborde la segmentación de una imagen por el color de la piel, necesario para el sistema. Ver Anexo # 5: ERS Validación2 v1.2, requisito # 11.
- Se especificó con un mayor nivel de detalle la descripción de los requisitos que contaban con ambigüedad. Ver Anexo # 5: ERS Validación2 v1.2, requisitos # 2, 3 y 4.
- Se especificó con un mayor nivel de detalle la descripción del requisito que contaba con poca comprensión. Ver Anexo # 5: ERS Validación2 v1.2, requisito # 10.

### **Actividad: Entregar Informe de Incidencias a Especialistas Funcionales:**

Dados los resultados anteriormente explicados se realizó una segunda ronda de validación. Se entregó a los Especialistas Funcionales el Informe de Incidencias de los resultados de la ronda anterior. Ver Tabla 6. Sección Informe de Incidencias.

Al concluir esta actividad, se realizan nuevamente las actividades: “Entregar cuestionario y documentación ERS a Especialistas Funcionales” y “Analizar respuestas de los cuestionarios”.

### **Actividad: Entregar cuestionario y documentación ERS a Especialistas Funcionales:**

Se hace entrega del cuestionario inicialmente aplicado a cada uno de los especialistas, así como la documentación del ERS a ser revisada para validar con los cambios realizados. Ver Anexo # 5: ERS Validación2 v1.2.

### **Actividad: Analizar respuestas de los cuestionarios:**

Realizadas todas las modificaciones a partir de los errores detectados en una primera ronda, la segunda fue muy satisfactoria, pues se logró un consenso entre los Especialistas Funcionales principalmente para los atributos de calidad: no ambiguo, comprensible y completo. No se detectaron errores. Los Especialistas Funcionales se encontraron satisfechos con los cambios realizados.

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

### **Medición de los atributos de calidad:**

Se tienen en cuenta además las métricas definidas para los atributos de calidad descritas en el capítulo anterior, teniendo en cuenta los resultados descritos anteriormente luego de aplicado el cuestionario en su segunda ronda, arrojando un resultado muy satisfactorio en la aplicación de las métricas, para los atributos críticos se obtuvo un resultado satisfactorio de  $Q = 1$ , lo que se muestra a continuación:

- No ambiguo:

$$Q_2 = 11/11 = 1.$$

Como  $Q_2$  es 1 y se está midiendo un atributo crítico para el sistema, se obtuvo un resultado satisfactorio, un ERS libre de ambigüedad.

- Completo:

Bloque A: 11 requisitos.

Requisitos que conocemos y han sido documentados.

Bloque B: 0 requisito.

Requisitos que han sido documentados pero:

- han sido pobremente especificados,
- han sido especificados abstractamente,

Bloque C: 0 requisito.

Requisitos que sabemos son necesarios pero no hemos documentado.

Bloque D: 0 requisitos.

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Requisitos potenciales que no entendemos lo suficientemente bien para ser documentados.

$$Q_3 = 11/11+0+0+0=1.$$

Como  $Q_3$ , aplicando la variante de la formula más completa es 1 y se está midiendo un atributo crítico para el sistema, se logró pasar todos los requisitos al bloque A, obteniendo un resultado satisfactorio, un ERS completo.

- Comprensible:

$$Q_4 = 11/11=1.$$

Como  $Q_4$  es 1 y se está midiendo un atributo crítico para el sistema, se obtuvo un resultado satisfactorio, un ERS comprensible.

- No redundante:

$$Q_5 = 11/11=1.$$

Como  $Q_5$  es 1, se cumple en un 100 % con el atributo de calidad No redundante.

- Correcto:

$$Q_6 = 11/11=1.$$

Como  $Q_6$  es 1, se cumple en un 100 % con el atributo de calidad Correcto.

### **Actividad: Preparar el Informe Final:**

Se elaboró un Informe Final que refleja el ERS y los resultados satisfactorios de la aplicación de las métricas. El informe final, está formado por el Informe de Incidencias de la primera ronda, dado que no se detectó ningún error en la segunda ronda (Ver Tabla 7. Sección Informe de Incidencias), el ERS al que se

## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

le aplicó la validación con sus cambios realizados (Ver Anexo # 5: ERS Validación2 v1.2), el ERS inicial (Ver Anexo # 1: ERS Validación1 v1.0) y el resultado de la medición de los atributos de calidad.

### **Actividad: Certificar el ERS:**

Se realizó la certificación del ERS que consiste en la formalidad de aceptación del ERS por parte del cliente del producto a través de una firma del documento.

### **Resultados de mejora obtenidos respecto a los problemas al concluirse las 2 rondas que fueron necesarias para lograr un consenso:**

- Se logró que cada requisito esté orientado a su origen.
- Se corrigieron faltas de ortografía en la documentación.
- Los requisitos no están redactados con una interpretación compleja, evitando la ambigüedad.
- Se logró claridad en la redacción del documento.
- Se realizó una validación más profunda y minuciosa de los requisitos.
- Se garantizó que los requisitos cumplen con los atributos de calidad:

#### Correcto:

- No existen errores en la definición de los objetivos del sistema, en este caso subsistema en desarrollo.
- No existe contradicción entre requisitos.

#### Comprensible:

- Los requisitos están redactados de forma simple y clara.
- Los requisitos están especificados en detalle.

#### No ambiguo:

- Se comprobó mediante las iteraciones y los criterios de los especialistas, la interpretación de los requisitos.



## CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Preciso:

- Los requerimientos se encuentran delimitados en términos cuantitativos, con niveles apropiados de precisión para cualquier cantidad numérica.

Completo:

- Los requisitos son correctos, cumplen con todas las funcionalidades que el usuario debe realizar.

Trazabilidad:

- Las trazas de dependencia entre los requisitos son correctas.

Luego de la aplicación de la Validación de Requisitos con Especialistas Funcionales propuesta por la presente investigación, se obtuvieron resultados de mejora respecto a la solución de los problemas, pues al concluirse todas las rondas se lograron requisitos muy claros y específicos. Se realizaron todas las rondas necesarias hasta obtener un ERS con la calidad requerida. Aunque es necesario dejar claro que un ERS perfecto es prácticamente imposible de obtener. Si eliminamos toda la ambigüedad en el documento estaríamos adicionando mucha formalidad que no sería muy comprensible por personas no especializadas en la rama informática. Si eliminamos toda la redundancia el ERS podría tornarse difícil de leer. Por otra parte si optamos por una total completitud podemos perder en consistencia.

Además se detectó un mayor número de errores durante la validación que se propone, que no se detectó con el método de validación del proyecto MOCIC aplicado anteriormente. Se garantizó un resultado cuantitativo a partir de la aplicación de métricas que garantizan un por ciento de fiabilidad respecto al cumplimiento de los atributos de calidad.

**Impacto y mejoras del proyecto Motor de Categorización Inteligente de Contenidos (MOCIC) a partir de la validación de los requisitos para etapas posteriores en el Subsistema Imágenes.**

Ver Anexo # 8 y 9.

### **CONCLUSIONES**

La validación de requisitos es un campo que aún presenta muchas inconsistencias y que al no realizarse con la calidad requerida puede llevar al retraso o cancelación de un proyecto de software. El presente trabajo propone la validación de los requisitos apoyándose en la esencia del método Delphi, haciendo uso además de determinadas métricas para la medición de algunos de los atributos de calidad más importantes. Esta propuesta a partir de la retroalimentación constante de problemas detectados, garantiza que el ERS no sea validado hasta tanto no exista un consenso entre todos los Especialistas Funcionales que asegure que el mismo es: no ambiguo, completo, correcto, posible de tracear, preciso, verificable, comprensible y no redundante; evitando de esta manera la aparición de errores en requisitos en etapas posteriores que impliquen un costo más elevado en el desarrollo del software.

### **RECOMENDACIONES**

1. Se recomienda la utilización o implementación de un sistema experto para realizar la Selección de Especialistas y el análisis de las respuestas de los cuestionarios.
2. Profundizar en la medición de otros atributos de calidad no tratados en el presente trabajo para enriquecer la propuesta de validación y obtener un ERS con una calidad mucho mayor.
3. Aplicar la propuesta Validación de Requisitos con Especialistas Funcionales a proyectos de desarrollo de software de la UCI, para medir a mayor profundidad los resultados que traería consigo.

### **BIBLIOGRAFÍA**

- Davis, Alan, y otros.** Identifying and Measuring Quality in a Software Requirements Specification. [En línea] 2005.
- Alexander, L.** Selection Criteria for Alternative Software Life-Cycle Process Models. [En línea] 2000.
- Bernárdez, B., Durán, A. y Toro, M.** UNA PROPUESTA PARA LA VERIFICACIÓN DE REQUISITOS BASADA EN MÉTRICAS. Sevilla : Procesos y Métricas de las tecnologías de la información. [En línea] 2004. Vol. 1, 2..
- Boehm, B.** Software Engineering. s.l. : Computers, IEEE Trans. [En línea] 1976.
- Christel, Michael G. y Kang, Kyo C.** *Issues in Requirements Elicitation.* s.l. : SEI. 1992.
- Davis, Alan, y otros.** *Identifying and Measuring Quality in a Software Requirements Specification.* 2005.
- Toronto, Universidad de.** *Software Requeriments Specification and Validation.* s.l. :. 2003 : Easterbrook, Steve.
- Escalona, María José y Koch, Nora.** Ingeniería de Requisitos en Aplicaciones para la Web. [En línea] 2002.
- González Moreno, Juan Carlos.** Análisis de Requisitos.IEEE. [En línea] 2009.
- López, Alfaro, y otros.** Delphi, Pronóstico. s.l. [En línea] 2005.
- Puello, Plinio y García, Fabio.** Gestión de la Verificación y Validación de Requisitos. [En línea] 2008.
- SSQC.** Requirements Management. *San Jose : Software System Quality Consulting.* [En línea] 2008.
- Concepción del sistema. MOCIC. Facultad#10 . Informaticas., Universidad de las Ciencias.* 2010.
- PRONOSTICO DELPHI. otros, Alfaro Lopez y.* s.l. : Universidad del Salvador. , 2005.
- Método Delphi: sondeo de expertos. [En línea] <http://www.laboris.net/>.
- Dra. Escalona Cuaresma, María José y Dr. González Romano, José Mariano.** Metodología y Técnicas en Proyectos software para la Web. [En línea] Universidad de Sevilla. Curso , Curso 2006/2007.  
<http://www.lsi.us.es/docencia>.
- Leguiza, Adrián y Romagnano , María.** Universidad Nacional de la Matanza - Universidad Nacional de San Juan. [En línea] 2006.
- Escalona, María José y Koch, Nora.** Ingeniería de Requisitos en Aplicaciones para la Web –Un estudio comparativo. [En línea] 2002.

## BIBLIOGRAFÍA

**Dr. Hong Zhu, Dr. Lingzi Jin , Prof. Dan Diape , Mr. Ganhong Bai.** Software requirements validation via task analysis. [En línea] 2002. <http://www.sciencedirect.com/>.

**UC3M, Ing. Conocimiento.** Verificación y Validación. [En línea] 2004.

**Álvarez, José R. y Arias, Manuel.** Validación de requisitos. [En línea] 2002. <http://www.ia.uned.es/>.

**Bruegge, Bernd y Dutoit, Allen.** Ingeniería de Software Orientado a Objetos Prentice-Hall. [En línea] 2002.

IEEE Recommended Practice for Software Requirements Specifications. *Software Engineering Standards Committee of the IEEE Computer Society. ISBN.* [En línea] 0-7381-0332-2.

**McGraw-Hill, Roger.** Ingeniería del Software. Un enfoque práctico. *Pressman.* s.l. : 5ta. Edición, 2002. ISBN 84-481-3214-9.

**Prentice-Hall, Ian.** Ingeniería de Software. *6ta. Edición.* s.l. : Sommerville, 2002.

*PRESSMAN.* 2002.

## ANEXOS

## Anexo 1: ERS Validación1 v1.0.

## 3.1.2 Requisitos funcionales

Nº	Funcionalidad	Descripción	Complejidad
1	Categorizar <u>imagenes</u> en desnuda y no desnuda.	El sistema debe ser capaz de categorizar las <u>imagenes</u> realizando una diferenciación en imagen con desnudos e imagen sin desnudos.	Alta
2	Entrenar el categorizador de imágenes desnudas.	El sistema debe ser capaz de a partir de la clasificación de las <u>imagenes</u> en "desnuda" actualizar su base de aprendizaje y conocimiento.	Alta
3	Guardar entrenamiento.	Almacenar el entrenamiento realizado al <u>categorizador</u> de <u>imagenes</u> desnudas.	Alta
4	Cargar entrenamiento	Cargar el entrenamiento realizado.	Alta
5	Detectar rostros en una imagen.	El sistema debe de ser capaz de identificar el rostro de una persona en una imagen.	Alta
6	Reconocer objetos en una imagen.	El sistema debe ser capaz de identificar un objeto en una imagen.	Alta
7	Mejorar la imagen para su procesamiento.	El sistema debe ser capaz de mejorar la calidad de una imagen para un procesamiento de la misma con mayor calidad.	Alta
8	Redimensionar la imagen.	El sistema debe ser capaz de ajustar las dimensiones de una imagen para mejorar la calidad de la misma.	Alta
9	Ajustar niveles de luminosidad.	El sistema debe ser capaz de ajustar los niveles de luminosidad de una imagen para mejorar la calidad de la misma.	Alta
10	Leer fichero de peticiones.	El sistema debe ser capaz de leer el fichero de peticiones.	Media
11	Escribir respuesta en el fichero de respuesta.	El sistema debe ser capaz de escribir respuesta en el fichero de peticiones.	Media

Figura 3. ERS Validación1 v1.0.

## Anexo 2: ERS Validación1 v1.1.

## 3.1.2 Requisitos funcionales

Nº	Funcionalidad	Descripción	Complejidad
1	Categorizar imágenes en desnuda y no desnuda.	El sistema debe ser capaz de categorizar las imágenes realizando una diferenciación en imagen con desnudos e imagen sin desnudos.	Alta
2	Entrenar el categorizador de imágenes desnudas.	El sistema debe ser capaz de a partir de la clasificación en "desnuda" actualizar su base de aprendizaje y conocimiento.	Alta
3	Guardar entrenamiento.	Almacenar el entrenamiento realizado al categorizador de imágenes desnudas.	Alta
4	Cargar entrenamiento	Cargar el entrenamiento realizado.	Alta
5	Detectar rostros en una imagen.	El sistema debe de ser capaz de identificar el rostro de una persona en una imagen.	Alta
6	Reconocer objetos en una imagen.	El sistema debe ser capaz de identificar un objeto en una imagen.	Alta
7	Redimensionar la imagen.	El sistema debe ser capaz de ajustar las dimensiones de una imagen para mejorar la calidad de la misma.	Alta
8	Ajustar niveles de luminosidad.	El sistema debe ser capaz de ajustar los niveles de luminosidad de una imagen para mejorar la calidad de la misma.	Alta
9	Leer fichero de peticiones.	El sistema debe ser capaz de leer el fichero de peticiones.	Media
10	Escribir respuesta en el fichero de respuesta.	El sistema debe ser capaz de escribir respuesta en el fichero de peticiones.	Media
11	Segmentar una imagen por el color de la piel.	El sistema debe ser capaz de fraccionar imágenes poniendo todo en negro y solo en blanco el color de la piel, para sacar las características de la imagen.	Alta

Figura 4. ERS Validación1 v1.1.

## Anexo 3: ERS Validación2 v1.0.

## 3.1.2 Requisitos funcionales

Nº	Funcionalidad	Descripción	Complejidad
1	Categorizar <u>imagenes</u> en desnuda y no desnuda.	El sistema debe ser capaz de categorizar las <u>imagenes</u> realizando una diferenciación en imagen con desnudos e imagen sin desnudos.	Alta
2	Entrenar el categorizador de imágenes desnudas.	El sistema debe ser capaz de a partir de la clasificación de las <u>imagenes</u> en "desnuda" actualizar su base de aprendizaje y conocimiento.	Alta
3	Guardar entrenamiento.	Almacenar el entrenamiento realizado al <u>categorizador de imagenes desnudas</u> .	Alta
4	Cargar entrenamiento.	Cargar el entrenamiento realizado.	Alta
5	Detectar rostros en una imagen.	El sistema debe de ser capaz de identificar el rostro de una persona en una imagen.	Alta
6	Reconocer objetos en una imagen.	El sistema debe ser capaz de identificar un objeto en una imagen.	Alta
7	Mejorar la imagen para su procesamiento.	El sistema debe ser capaz de mejorar la calidad de una imagen para un procesamiento de la misma con mayor calidad.	Alta
8	Redimensionar la imagen.	El sistema debe ser capaz de ajustar las dimensiones de una imagen para mejorar la calidad de la misma.	Alta
9	Ajustar niveles de luminosidad.	El sistema debe ser capaz de ajustar los niveles de luminosidad de una imagen para mejorar la calidad de la misma.	Alta
10	Leer fichero de peticiones.	El sistema debe ser capaz de leer el fichero de peticiones.	Media
11	Escribir respuesta en el fichero de respuesta.	El sistema debe ser capaz de escribir respuesta en el fichero de peticiones.	Media

Figura 5. ERS Validación2 v1.0.



## Anexo 4: ERS Validación2 v1.1.

## 3.1.2 Requisitos funcionales

Nº	Funcionalidad	Descripción	Complejidad
1	Categorizar imágenes en desnuda y no desnuda.	El sistema debe ser capaz de categorizar las imágenes realizando una diferenciación en imagen con desnudos e imagen sin desnudos.	Alta
2	Entrenar el categorizador de imágenes desnudas.	El sistema debe ser capaz de a partir de la clasificación de las imágenes en "desnuda" actualizar su base de aprendizaje y conocimiento.	Alta
3	Guardar entrenamiento.	Almacenar el entrenamiento realizado al categorizador de imágenes desnudas.	Alta
4	Cargar entrenamiento	Cargar el entrenamiento realizado.	Alta
5	Detectar rostros en una imagen.	El sistema debe de ser capaz de identificar el rostro de una persona en una imagen.	Alta
6	Reconocer objetos en una imagen.	El sistema debe ser capaz de identificar un objeto en una imagen.	Alta
7	Redimensionar la imagen.	El sistema debe ser capaz de ajustar las dimensiones de una imagen para mejorar la calidad de la misma.	Alta
8	Ajustar niveles de luminosidad.	El sistema debe ser capaz de ajustar los niveles de luminosidad de una imagen para mejorar la calidad de la misma.	Alta
9	Leer fichero de peticiones.	El sistema debe ser capaz de leer el fichero de peticiones para detectar las peticiones enviadas por el controlador para la interpretación de imágenes desnudas, y poder buscar en su entrenamiento una respuesta.	Media
10	Escribir respuesta en el fichero de respuesta.	El sistema debe ser capaz de escribir respuesta en el fichero de peticiones.	Media

Figura 6. ERS Validación2 v1.1.

## Anexo 5: ERS Validación2 v1.2.

## 3.1.2 Requisitos funcionales

Nº	Funcionalidad	Descripción	Complejidad
1	Categorizar imágenes en desnuda y no desnuda.	El sistema debe ser capaz de categorizar las imágenes realizando una diferenciación en imagen con desnudos e imagen sin desnudos.	Alta
2	Entrenar el categorizador de imágenes desnudas.	El sistema debe ser capaz luego de categorizar en "desnuda" o "no desnuda" las imágenes guardar las características para entrenar la red neuronal.	Alta
3	Guardar entrenamiento.	Almacenar el entrenamiento del categorizador en la red neuronal, guardando las características para su posterior uso en el clasificador de imágenes desnudas.	Alta
4	Cargar entrenamiento	Cargar el entrenamiento de las características de las imágenes clasificadas en la red neuronal, para posterior uso del categorizador de imágenes desnudas con la red neuronal entrenada.	Alta
5	Detectar rostros en una imagen.	El sistema debe de ser capaz de identificar el rostro de una persona en una imagen.	Alta
6	Reconocer objetos en una imagen.	El sistema debe ser capaz de identificar un objeto en una imagen.	Alta
7	Redimensionar la imagen.	El sistema debe ser capaz de ajustar las dimensiones de una imagen para mejorar la calidad de la misma.	Alta
8	Ajustar niveles de luminosidad.	El sistema debe ser capaz de ajustar los niveles de luminosidad de una imagen para mejorar la calidad de la misma.	Alta
9	Leer fichero de peticiones.	El sistema debe ser capaz de leer el fichero de peticiones para detectar las peticiones enviadas por el controlador para la interpretación de imágenes desnudas, y poder buscar en su entrenamiento una respuesta.	Alta
10	Escribir respuesta en el fichero de respuesta.	El sistema debe ser capaz de escribir respuesta en el fichero de peticiones para obtener la categorización de la imagen enviada en el fichero de peticiones anteriormente leído.	Media
11	Segmentar una imagen por el color de la piel.	El sistema debe ser capaz de fraccionar imágenes poniendo todo en negro y solo en blanco el color de la piel, para sacar las características de la imagen.	Alta

Figura 7. ERS Validación2 v1.2.

**Anexo 6: Respuesta 1 del Cuestionario.**

**Cuestionario a ser respondido por Especialistas Funcionales:**

**Importante aclarar que las respuestas deben ser concretas, sin indecisión: SI o NO y en caso de ser necesario redactar con la mayor claridad posible.**

1. ¿Están correctos todos los requisitos?

Respuesta: si.

2. ¿Algún requisito incumple restricciones definidas por el negocio? ¿Cuáles?

Respuesta: no.

3. ¿Están correctamente definidos los objetivos del sistema? ¿Cuáles no? ¿Cuáles faltan?

Respuesta: si.

4. ¿Se pueden localizar todos los requisitos en el conjunto de objetivos del sistema/producto?  
¿Cuáles no?

Respuesta: si.

5. ¿Existen requisitos que se contradicen con otros requisitos? ¿Cuáles?

Respuesta: no.

6. ¿Están todos los requisitos redactados de forma simple y clara? ¿Cuáles no?

Respuesta: no, el requisito #10.

7. ¿Deberían especificarse algunos requisitos con más detalle? ¿Cuáles?

Respuesta: si, el requisito #10.

8. ¿Considera todos los requisitos libres de ambigüedad? ¿Cuáles no?

Respuesta: no, el requisitos #3 y #4.

9. Realice una justificación breve de lo que ha interpretado en cada uno de los requisitos que considera cuentan con ambigüedad.

Respuesta:

Requisito #3: Guardar el entrenamiento realizado anteriormente con las características obtenidas de la categorización de las imágenes durante el proceso del mismo para un uso posterior con más vigor, rapidez y eficacia en el proceso de clasificación de imágenes en desnudas o no desnudas.

Requisito #4: Cargar los entrenamientos guardados junto con sus características obtenidas, para una posterior clasificación de las nuevas imágenes con la red neuronal actualizada.

10. ¿Todos los requisitos están delimitados en términos cuantitativos (en caso de ser posible)?  
¿Cuáles no?

Respuesta: si.

11. ¿Se utilizan en todos los requisitos delimitados en términos cuantitativos niveles apropiados de precisión para todas las cantidades numéricas? ¿En cuáles no?

Respuesta: si.

12. ¿Todos los requisitos documentados son correctos? ¿Cuáles no?

Respuesta: si.

13. ¿Se han identificado todas las funciones que el usuario debe hacer? ¿Cuáles faltan?

Respuesta: no, falta Segmentar una imagen por el color de la piel.

14. ¿Las trazas de los requisitos a sus fuentes de origen, casos de uso o procesos del negocio son correctas? ¿Cuáles no? (Para el caso de proyectos que realicen modelación de negocio)

Respuesta: si.

15. ¿Las trazas de dependencia definidas entre requisitos son correctas? ¿Cuáles no?

Respuesta: si.

16. ¿Existe redundancia en la definición de los requisitos? ¿En cuáles?

Respuesta: no.

**Anexo 7: Respuesta 2 del Cuestionario.**

**Cuestionario a ser respondido por Especialistas Funcionales:**

**Importante aclarar que las respuestas deben ser concretas, sin indecisión: SI o NO y en caso de ser necesario redactar con la mayor claridad posible.**

1. ¿Están correctos todos los requisitos?

Respuesta: si.

2. ¿Algún requisito incumple restricciones definidas por el negocio? ¿Cuáles?

Respuesta: no.

3. ¿Están correctamente definidos los objetivos del sistema? ¿Cuáles no? ¿Cuáles faltan?

Respuesta: si.

4. ¿Se pueden localizar todos los requisitos en el conjunto de objetivos del sistema/producto?  
¿Cuáles no?

Respuesta: si.

5. ¿Existen requisitos que se contradicen con otros requisitos? ¿Cuáles?

Respuesta: no.

6. ¿Están todos los requisitos redactados de forma simple y clara? ¿Cuáles no?

Respuesta: no, el requisito #10.

7. ¿Deberían especificarse algunos requisitos con más detalle? ¿Cuáles?

Respuesta: si, el requisito #10.

8. ¿Considera todos los requisitos libres de ambigüedad? ¿Cuáles no?

Respuesta: no, los requisitos # 2 y 4.

9. Realice una justificación breve de lo que ha interpretado en cada uno de los requisitos que considera cuentan con ambigüedad.

Respuesta:

Requisito #2: El sistema debe ser capaz luego de un análisis de las imágenes y de una caracterización en desnudas o no desnudas obtener las características necesarias para un posterior entrenamiento de la red neuronal.

Requisito #4: Cargar los entrenamientos realizados anteriormente para una actualización de la red neuronal para una posterior clasificación de las imágenes con un mayor nivel de detalle y especificación.

10. ¿Todos los requisitos están delimitados en términos cuantitativos (en caso de ser posible)?  
¿Cuáles no?

Respuesta: si.

11. ¿Se utilizan en todos los requisitos delimitados en términos cuantitativos niveles apropiados de precisión para todas las cantidades numéricas? ¿En cuáles no?

Respuesta: si.

12. ¿Todos los requisitos documentados son correctos? ¿Cuáles no?

Respuesta: si.

13. ¿Se han identificado todas las funciones que el usuario debe hacer? ¿Cuáles faltan?

Respuesta: si.

14. ¿Las trazas de los requisitos a sus fuentes de origen, casos de uso o procesos del negocio son correctas? ¿Cuáles no? (Para el caso de proyectos que realicen modelación de negocio)

Respuesta: si.

15. ¿Las trazas de dependencia definidas entre requisitos son correctas? ¿Cuáles no?

Respuesta: si.

16. ¿Existe redundancia en la definición de los requisitos? ¿En cuáles?

Respuesta: no.



**Anexo 8: Firmas de especialistas.**

Los especialistas involucrados en la validación de los requisitos fueron los siguientes:



Ing. Kiuver Kaddiel Ibáñez Castro

J' Proyecto MOCIC



Ing. Luis Enrique Sánchez Arce

J' Proyecto FILPACON



Ing. Mailen Proenza Guerra

Analista principal de MOCIC



Lic. Karel Antonio Verdecia Ortiz

Arquitecto principal de MOCIC



Ing. David Antonio Riquelme Mendez

J' Dpto Soluciones Informáticas para Internet

Facultad 10

Anexo 9: Impacto y mejoras para MOCIC.

**Mejoras del proyecto Motor de Categorización Inteligente de Contenidos (MOCIC) a partir de la validación de los requisitos**

**Universidad de las Ciencias Informáticas (UCI)**

El proyecto MOCIC es un proyecto de Investigación y Desarrollo (I+D) por lo que los requisitos varían constantemente, de aquí la importancia de que los requisitos sean los adecuados, según las necesidades del cliente. Hasta la fecha los requisitos de MOCIC no son prácticamente validados, más allá de la aceptación o no de un solo cliente, no se realiza ningún otro procedimiento que asegure que los mismos son los correctos, por esto en muchas ocasiones eran incoherentes con respecto a lo que quería el cliente y en otros casos innecesarios. En busca de la solución a esta problemática se decidió aplicar la propuesta de validación de una tesis de grado, perteneciente a la Facultad 15, cuya investigación tiene como tema: Validación de Requisitos con Especialistas Funcionales para proyectos productivos de la UCI.

Para la validación de los requisitos se usaron especialistas, que fueron representados por los más experimentados del proyecto MOCIC y de FILPACON; los de este último proyecto corresponden al personal del cliente directo, pero por necesidades del proyecto y para su buen desempeño y desarrollo del mismo algunos especialistas de MOCIC se convierten en sus propios clientes. Estos especialistas seleccionados refinaron los requisitos mediante un proceso propuesto por la tesis de grado mencionada.

En el transcurso del desarrollo del Subsistema Imágenes de MOCIC al que se le aplicó la Validación con Especialistas Funcionales se evidenció la calidad de los requisitos que pasaron por el proceso de validación, ya que en todo el tiempo de desarrollo luego de aplicada la propuesta de validación, solo hubo la necesidad de reformular uno de ellos por razones ajenas al proyecto, enfocándolo a una nueva necesidad de la Universidad. La validación de los requisitos es algo necesario en cualquier proyecto productivo, existen diversas formas de validar los requisitos de un proyecto pero este método en concreto ha demostrado ser sencillo de aplicar, dinámico y efectivo.

Dado que se validaron los requisitos con el método propuesto por la tésista Arelis Nelsa Zamora Abreu y este demostró ser efectivo, al menos para el Subsistema Imágenes del proyecto en cuestión al que se le aplicó, se expide un certificado por parte del proyecto MOCIC dando fe de que el método propuesto es válido. Se recomienda a todos los proyectos de Investigación y Desarrollo (I+D) de la universidad el uso de la validación de los requisitos propuesta por el trabajo de diploma, Validación de Requisitos con Especialistas Funcionales para Proyectos de Desarrollo de Software en la UCI, dada la naturaleza dinámica de este tipo de proyecto, lo cual evitará en gran medida que se tengan que reformular los requisitos, disminuyendo así el tiempo de desarrollo y el costo.

**Anexo 10: Matriz de trazabilidad.**

**Matriz de trazabilidad.**

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11
Detección de Rostros en Imágenes		x	x	x	x		x	x	x	x	
Reconocimiento de Objetos en Imágenes		x	x	x		x	x	x	x	x	
Clasificación de Imágenes Pornográficas	x	x	x	x			x	x	x	x	x

**Objetivos del Subsistema & Los Requisitos**

### GLOSARIO DE TÉRMINOS

**ERS:** Documento Especificación de Requisitos de Software.

**MOCIC:** Motor de Clasificación Inteligente de Contenidos.

**Filpacon:** Filtrado de Paquetes por Contenido.

**IR:** Ingeniería de Requisitos.

**URLs:** Unified Resource Locate. Es el identificador de una página en Internet, es decir, un nombre único de un recurso de Internet.

**Métrica:** Medio para entender, monitorizar, controlar, predecir y probar el desarrollo de software y los proyectos de mantenimiento.

**Validación de Requisitos:** Procedimiento en el que se verifica si los requisitos del software son los correctos para que el producto de software final, cumpla con la solución de los problemas del cliente.

**Squid proxy:** popular programa de software libre que implementa un servidor proxy y un dominio para caché de páginas web, publicado bajo licencia GPL. Tiene una amplia variedad de utilidades, desde acelerar un servidor guardando en caché peticiones repetidas a DNS y otras búsquedas para un grupo de gente que comparte recursos de la red, hasta caché de web, además de añadir seguridad filtrando el tráfico. Está especialmente diseñado para ejecutarse bajo entornos UNIX.