



# **UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

## **FACULTAD 3**

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

**Título:** Propuesta para el Levantamiento de Requisitos Funcionales en  
función del Aseguramiento de la Calidad de Software.

**Autor(es):** Yeilin Martínez Torres.

**Tutor(es):** Ing. Yaumarys Pino Cueto.

Ing. Maylin Bacallao Martínez.

Ing. Heney Díaz Perez.

**La Habana, Cuba**

**Junio, 2009**

---

# DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Yeilin Martínez Torres

Ing. Yaumarys Pino Cueto

\_\_\_\_\_

\_\_\_\_\_

Firma del Autor

Firma del Tutor

Ing. Maylin Bacallao Martínez

Ing. Heney Díaz Pérez

\_\_\_\_\_

\_\_\_\_\_

Firma del Tutor

Firma del Tutor

*“Si una persona es perseverante, aunque sea dura de entendimiento, se hará inteligente; y aunque sea débil se transformará en fuerte.”*

*Leonardo Da Vinci*

## DEDICATORIA

*A mis padres:*

Gracias por brindarme este apoyo inmenso durante mi vida, no solo en mi preparación en la universidad, si no desde que me trajeron al mundo. Por confiar en mí, por saber que lograría ser hoy una profesional, pero no lo hubiese hecho sin ustedes, hoy no sería lo que he alcanzado, y palabras para agradecerles lo que tengo no existen, porque es demasiado grande lo que han depositado en mí y de esta manera al menos recompensarlos dedicándole mi esfuerzo durante estos años.

*A mi hermano:*

A pesar de las peleas siempre me has guiado, siempre has confiado en mí y has tratado de que siga los pasos y los sueños de nuestros padres, hoy tú también formas parte de estas hojas que guardan el esfuerzo de varios años de lucha y desesperación.

*A mi sobrina:*

Para que te sirva de ejemplo y logres ser alguien en la vida, que te prepares como futura profesional de este país y logres tus sueños.

*A mi abuela Teresa y mi abuelo Ernesto Torres (Cachimba):*

Por guiarme cuando niña y lograr que cada día fuera mejor, a pesar de que me exigían y me regañaban para no cometer fallos durante mi crecimiento, y a ti abuelo que aunque no estés aquí con nosotros, quiero también dedicarte este momento que tanto trabajo me ha costado.

**Gracias...**

### AGRADECIMIENTOS

Primero que a nadie mis gracias totales a mis tutoras y más que tutoras amigas, hermanas, a **May** y **Yaumi** que desde el primer día que nos conocimos me han brindado su apoyo, su amistad, su respeto, me han guiado a ser mejor cada día. Han logrado que este trabajo sea hoy por hoy mi graduación. Una vez una sabia amiga me dijo, en la universidad encontrarás a muchas personas pero amigos son difíciles, eso sí, si lo encuentras, no lo dejes nunca, y este es nuestro caso, llevamos momentos duros en la carrera, lloramos y reímos por culpa de las materias o motivos extras, pero siempre lo hicimos juntas y eso de verdad para mí tiene mucho valor. Gracias por su disposición y ayuda que sin ella no lo hubiese logrado.

A mi otro tutor **Heney**, gracias por el apoyo que me diste durante la realización de este trabajo, que sin tu ayuda no hubiese sido posible.

A todos los profesores por contribuir a mi formación como profesional.

A los que colaboraron y me ayudaron a realizar el cuestionario, **Roig, Geiser, Yudisbel, Diana, Danaysa** y **Lourdes**. A todos mil gracias.

A la universidad (**UCI**) y a **Fidel** por permitirme formar parte de este sueño.

A mi tía **Tere**, mi prima **Belkis, Boris** y los niños **Yandry** y **Andy** por confiar en mí, depositar tanto apoyo y aliento durante todo este tiempo.

A **Luly, Eilys, Odaimy, Olia, Yoe**, por esos momentos buenos y malos que hemos cursado durante la carrera, gracias por su apoyo.

A **Glen** por escucharme en los días que mi mente estaba tensa, y darme el aliento de seguir adelante y confiar en mí misma, dándole prioridad a lo que realmente estaba construyendo.

A **losmel** por la gran ayuda que me ofreció con mis ensayos de tesis y su hermano **losmany** por ayudarme en estos momentos de arduo trabajo, por su apoyo y su dedicación, tratando de que mi mente despejara.

A todos mis amigos, gracias por ayudarme, por su preocupación en todo momento y hacerme sentir que estaban ahí, conmigo.

A mis compañeros de aula y compañeros de producción, gracias por compartir todos esos momentos de tantas horas delante de un monitor trabajando para lograr que el proyecto saliera adelante.

## AGRADECIMIENTOS

---

A todas aquellas personas que en un momento preguntaron "... ¿Yei y la tesis?"

Mis disculpa si se me queda alguien por mencionar, pero les estoy totalmente agradecida a todos los que tuvieron que ver de una forma u otra con la realización de este trabajo.

A todos Gracias...

### RESUMEN

El Levantamiento de Requisitos es el desarrollo de las actividades necesarias para el inicio de un software. La Universidad de las Ciencias Informáticas (UCI) está desarrollando varios proyectos de software en los que incumple con algunas de las actividades dentro del flujo de trabajo de Requerimientos. Las revisiones por parte del equipo de calidad interna no se están desarrollando óptimamente lo que provoca la no conformidad del cliente. De ahí la necesidad de realizar una Propuesta para el Levantamiento de Requisitos en función del Aseguramiento de la Calidad de Software.

En este trabajo se hace un estudio de las principales técnicas y herramientas para el Levantamiento de Requisitos que tribute a la calidad del producto. Se toma el proyecto de Cooperación Integral del Convenio Cuba-Venezuela (CCV) como fuente específica de información, en cuanto al desarrollo del Levantamiento de Requisitos. A partir de este estudio se definen los principios y premisas que debe cumplir la propuesta a desarrollar en esta investigación, la cual está orientada a realizar una guía para el buen desarrollo de la captura de la información en el comienzo de un software, determinando las etapas de la Ingeniería de Requisitos como Elicitación, Análisis, Especificación y Validación.

Por cada etapa se definen las actividades que se deben realizar, los roles involucrados, responsables y además se definen los artefactos de entrada y salida por cada una de ellas. Para esto, se elabora un Plan de Revisiones en el que se determinan, como paso esencial, los objetivos que se persiguen con la Gestión de Calidad interna durante la realización de cada una de las actividades definidas en la propuesta. Dedicamos también una parte importante a la organización del Equipo de Desarrollo, el cual tendrá una persona encargada de asegurar todo el proceso de Levantamiento de Requisitos, proponiéndose además la utilización de los estándares de calidad **ITIL**, **COBIT** y la **IEE 830** para la mejora continua de los procesos, además de la utilización del modelo de proceso **CMMi**. La propuesta está dirigida a los proyectos que utilicen la metodología de desarrollo “El Proceso Unificado del Software (RUP)”.

### PALABRAS CLAVE

Calidad, Estándares, Gestión, Plan, Requisitos, Software.

ÍNDICE

**INTRODUCCIÓN** ..... 1

**CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA** ..... 5

1.1. Proceso de desarrollo de software. .... 5

1.2. Requisitos dentro de las Metodologías de Desarrollo de Software. .... 6

1.2.1. ¿Qué es una Metodología? ..... 6

1.2.2. El proceso Unificado del Software. RUP. .... 7

1.2.3. Extreme Programming (XP) ..... 10

1.2.4. Microsoft Solution Framework (MSF) ..... 11

1.2.5. SCRUM..... 12

1.2.6. ¿Por qué usar metodología de desarrollo RUP? ..... 13

1.3. Ingeniería de Requisitos..... 14

1.3.1. ¿Qué es la Ingeniería de Requisitos (IR)? ..... 14

1.3.2. Características de los Requerimientos. .... 16

1.3.3. Dificultades para definir los requerimientos. .... 17

1.3.4. Importancia de la Ingeniería de Requisitos. Beneficios. .... 17

1.3.5. Personal involucrado en la Ingeniería de Requerimientos..... 18

1.3.6. Levantamiento de Requisitos. .... 18

1.3.7. Técnicas utilizadas en las actividades de la IR..... 19

1.4. Definiciones de Calidad de Software..... 21

1.4.1. ¿Qué es la Gestión de la Calidad de Software? Aspectos fundamentales. .... 22

1.4.2. ¿Qué es la Calidad de Software?..... 23

1.4.3. Aseguramiento de la Calidad de Software. .... 23

1.4.4. Factores que determinan la Calidad de Software..... 26

1.5. Modelos y Estándares de Calidad..... 27

1.5.1. ¿Qué es ITIL? ..... 27

1.5.2. ¿Qué es COBIT? ..... 28

1.5.3. ¿Qué es CMMi? ..... 29

1.5.4. Modelo de Madurez de los Procesos. .... 31

1.5.5. Requisitos dentro de los Estándares de Calidad. .... 32

1.6. La Calidad de Software desde la Gestión de Requisitos. .... 35

1.6.1. ¿Cómo obtener un software con calidad a partir del Levantamiento de Requisitos? .... 35

1.7. Conclusiones ..... 36

<b>CAPÍTULO 2: PROPUESTA DE SOLUCIÓN</b> .....	37
2.1. Descripción de la Propuesta. ....	37
2.2. Objetivos de la Propuesta. ....	39
2.3. Flujo de la Propuesta. ....	40
2.4. Roles involucrados en la Propuesta. ....	41
2.5. Documentos que se generan en la Propuesta.....	47
2.6. Estándares y Modelo de procesos utilizados en la Propuesta. ....	49
2.7. Técnicas y Herramientas a utilizar en la Propuesta.....	50
2.8. Resolución de problemas y acciones correctivas para el Levantamiento de Requisitos. ....	51
2.9. Análisis de Riesgo.....	52
2.10. Plan de Revisiones de la Propuesta. ....	52
2.11. Pruebas y Evaluaciones que se realizan en la Propuesta.....	53
2.12. Descripción de las Etapas de la Propuesta.....	54
2.12.1. Elicitación de los Requisitos de Software. ....	54
2.12.2. Fase de Análisis de los Requisitos de Software. ....	57
2.12.3. Fase de Especificación de los Requisitos de Software.....	58
2.12.4. Fase de Validación de los Requisitos de Software. ....	62
2.13. Conclusiones.....	64
<b>CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA</b> .....	65
3.1. Tipos de Evaluación.....	65
3.2. Criterio de Evaluación .....	67
3.3. Diseño del cuestionario .....	67
3.4. Selección de los especialistas.....	67
3.5. Especialistas seleccionados.....	68
3.6. Entrega de los cuestionarios y de la propuesta .....	70
3.7. Resultados de los cuestionarios.....	70
3.8. Análisis de los resultados.....	72
3.9. Análisis de la concordancia en la valoración de aspectos (coeficiente de Kendall). ....	73
3.9.1. Gráficas estadísticas de resultados de las evaluaciones.....	74
3.10. Conclusiones.....	75
<b>CONCLUSIONES</b> .....	76
<b>RECOMENDACIONES</b> .....	78
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	79
<b>BIBLIOGRAFÍA</b> .....	81

<b>ANEXOS</b> .....	83
Anexo 1: Plantilla Plan de Revisiones .....	83
Anexo 2: Plantilla Revisión Técnica Formal.....	87
Anexo 3: Plantilla Minuta de entrega .....	91
Anexo 4: Plantilla Glosario de Terminos. ....	93
Anexo 5: Plantilla Documento de los procesos del negocio.....	98
Anexo 6: Plantilla Listado de Requisitos Funcionales.....	102
Anexo 7: Plantilla de Lista de Riesgos .....	104
Anexo 8: Plantilla Listas de Chequeo.....	110
Anexo 9: Plantilla Informe de No Conformidades. ....	143
Anexo 10: Plantilla Documento Visión.....	146
Anexo 11: Plantilla Documento de Especificación de los Requisitos de Software .....	155
Anexo 12 Plantilla Registro de Revisiones .....	160
Anexo 13 Plantilla Minuta de Aceptación .....	164
Anexo 14 Plantilla Modelo de Caso de Uso del Sistema .....	166
Anexo 15 Elicitación de los Requisitos de Software. ....	170
Anexo 16 Análisis de los Requisitos de Software.....	177
Anexo 17 Especificación de los Requisitos de Software.....	181
Actividad: Especificar los Requisitos de Software. ....	181
Anexo 18 Validación de los Requisitos de Software.....	189
Anexo 19 Cuestionario para la evaluación de la propuesta .....	191
<b>GLOSARIO DE TÉRMINOS</b> .....	196

## ÍNDICE DE FÍGURAS

Figura 1: Proceso de Desarrollo de Software .....	6
Figura 3: Desarrollo Iterativo de RUP.....	8
Figura 4: Extreme Programming (XP). .....	11
Figura 5: Microsoft Solution Framework (MSF). .....	12
Figura 6: Estándar ITIL, COBIT y CMMi. ....	32
Figura 7: Imagen del Flujo de la Propuesta. ....	40
Figura 8: Roles involucrados por parte del Cliente. ....	41
Figura 9: Roles involucrados en el Equipo de Desarrollo. ....	43
Figura 10: Imagen de las Actividades de la etapa de Elicitación .....	56
Figura 11: Imagen de las Actividades de la etapa de Análisis. ....	58
Figura 12: Imagen de las Actividades de la etapa de Especificación. ....	61
Figura 13: Imagen de las Actividades de la etapa de Validación. ....	63
Figura 14: Tabla de Criterio de Evaluaciones.....	72
Figura 15: Tabla de los aspectos a evaluar.....	72
Figura 16: Tabla Estadística Descriptiva. ....	73
Figura 17: Promedio de evaluaciones por parámetros. ....	74
Figura 18: Promedio de evaluaciones por especialistas. ....	74

## INTRODUCCIÓN

Actualmente todas aquellas empresas que desarrollan software han de moverse en un entorno altamente competitivo, habiéndose visto obligadas a reducir el tiempo de llegada al mercado de sus aplicaciones, manteniéndose las mismas exigencias de calidad por parte de sus clientes.

Muchas empresas relegan fases relacionadas con la calidad del software al plano meramente administrativo o incluso llegan a obviarlas, apremiadas por las presiones del cliente o del mercado. La calidad en un proceso de desarrollo de software debe empezar a construirse desde sus inicios y una de las fases decisivas a la hora de determinar el éxito es: la especificación clara de los requisitos de software.

Con frecuencia se encuentran situaciones en las que los requisitos se ven sometidos a un cambio continuo. En ocasiones ocurre que el propio cliente no es capaz de definir de una manera clara los requisitos para su producto, se siente indeciso, o bien plantea una serie de requisitos que incluyen alguna contradicción. Otras veces el problema radica en la falta de comprensión entre lo que el cliente realmente desea y lo que el equipo de desarrollo interpreta, aunque ambas partes piensen que han enfocado la problemática de la misma manera.

Entre lo que se especifica y se diseña, y el momento en que se desarrolla un producto de software surgen motivos de otra naturaleza, difícilmente detectables en la fase de definición de requisitos, y que dan origen a modificaciones. Entre ellos pueden estar los cambios en el negocio o cambios que surjan en las leyes que rige el sistema de gobierno del país al cual pertenezca y de una manera u otra afecten el desarrollo del software. En consecuencia la funcionalidad prevista puede llegar a tener una mínima relación con los requisitos del producto final.

Existen empresas que desarrollan software a través de convenios que se firman entre ellas mismas, o instituciones las cuales llegan a firmar un contrato y a partir de ahí logran desarrollar un producto. Muchos de esos contratos están relacionados con nuestro país, ya sea a nivel nacional o internacional. Como ejemplo de esto se tienen convenios entre la República de Cuba y la República Bolivariana de Venezuela, que mantienen la informatización de actividades colaboradas de los proyectos firmados según lo planteado en los principios del ALBA, de modo que se permita la automatización de los procesos, la conceptualización de los mismos y los acuerdos entre las partes involucradas en función de las características y alcance de los proyectos. Estas naciones han logrado desarrollar habilidades en el comercio y los tratados, las cuales presentan un mutuo interés productivo en cuanto a la economía de los dos países, conjuntamente con las ventajas que esto promueve, el alcance Económico - Social y la integración de América Latina y el Caribe.

Esta integración ha logrado desarrollar el proyecto Informatización Integral del Convenio Cuba – Venezuela, el cual le fue asignado a la Universidad de las Ciencias Informáticas (UCI) para su desarrollo. La UCI desde su fundación ha logrado un avance en cuanto a su producción en el desarrollo de software, tanto para la exportación como para la informatización del país. Tiene como tarea fundamental preparar futuros ingenieros que garanticen el nivel de producción y el desarrollo de la informática. Sin embargo, para la producción de software se ha de tener en cuenta la utilización de los estándares internacionales de calidad, para que desde el Levantamiento de Requisitos que es la etapa inicial de desarrollo de software, se tenga en cuenta el Aseguramiento de la Calidad.

Partiendo siempre de lo que el cliente quiere, desea y necesita para su mayor satisfacción, sólo se le podrá ofertar un producto de garantía si se establece una Gestión de Calidad de Software durante la ejecución de los proyectos de software, realizándose para ello las pertinentes evaluaciones, inspecciones, auditorías y revisiones que aseguren que se cumplan las responsabilidades asignadas, se utilicen eficientemente los recursos y se logre el cumplimiento del alcance del producto. La fase de Levantamiento de Requisitos se realiza a través de un grupo de personas encargadas de llevar el control de cada una de las especificaciones de requisitos identificadas, con el fin de mantener bajo control el proceso y eliminar las causas de los defectos que se puedan producir durante esta fase.

Sin embargo, muchas veces estos elementos se incumplen o por una falta de coordinación de los implicados en el proceso, o por el líder del software, o la mayoría de las veces porque no se dispone de suficiente tiempo para poder realizar la planificación requerida.

En el ámbito del proyecto Informatización Integral del Convenio Cuba - Venezuela se detectaron algunas deficiencias relacionadas con el Levantamiento de Requisitos y el Aseguramiento de la Calidad de Software de dicho proceso, que han influido de alguna manera en el resultado final del proyecto. Algunas de estas deficiencias fueron debidos a:

- No existe una planificación, control, seguimiento y revisión de la calidad durante el levantamiento de requisitos.
- No existe una revisión por parte del equipo de desarrollo de las plantillas a utilizar durante el levantamiento de requisitos, lo cual trae consigo que existan errores en los artefactos generados durante esta etapa.
- No existe un entendimiento entre el cliente y el equipo de desarrollo en cuanto a las necesidades a automatizar.
- No se involucran todos los roles necesarios del equipo de desarrollo en las definiciones que se realizan para el Levantamiento de Requisitos.

El presente trabajo de diploma pretende dar solución a la situación problemática anteriormente expuesta, para lo cual se plantea como **problema** de investigación lo siguiente:

¿Cómo realizar un Levantamiento de Requisitos Funcionales que tribute a la calidad del producto final?

Para solucionar las deficiencias anteriores, se establece como **Objeto de estudio**: Proceso de Desarrollo de Software. Siendo el **Campo de acción**: Levantamiento de Requisitos Funcionales.

Para lo cual se define como **Objetivo general**: Realizar una propuesta para el Levantamiento de Requisitos Funcionales en función del Aseguramiento de la Calidad de Software.

La Hipótesis de la que se parte es la siguiente:

Si se realiza un Levantamiento de Requisitos Funcionales en función del Aseguramiento de la Calidad de Software, se logrará que la realización del mismo tribute a la calidad del producto final.

Las **tareas** que se proponen desarrollar para el cumplimiento del objetivo son:

- Elaborar el marco teórico de la investigación.
- Realizar la propuesta de desarrollo para el Levantamiento de Requisitos Funcionales en función del Aseguramiento de la Calidad de Software.
- Realizar plantillas para los artefactos generados en el desarrollo de todo el proceso de Levantamiento de Requisitos Funcionales en función del Aseguramiento de la Calidad de Software.
- Realizar la validación de la Propuesta para el Levantamiento de Requisitos Funcionales en función del Aseguramiento de la Calidad de Software.

El presente documento se estructura en tres Capítulos:

**Capítulo 1 Fundamentación Teórica:** Se abordan temas referentes a la calidad de software, así como también los conceptos y definiciones de la misma, realizándose un análisis enfocado a las industrias de software a nivel mundial, nacional y centrándolo en la universidad. Se sintetizan algunos modelos y estándares de calidad considerándose los más importantes aplicados al desarrollo del software para la etapa de Levantamiento de Requisitos. Se abarca la etapa de Levantamiento de Requisitos haciendo énfasis en los conceptos básicos de la etapa y lográndose un mayor análisis para el aseguramiento de la calidad del mismo.

**Capítulo 2 Propuesta de solución:** Después de haber realizado un análisis con las dificultades encontradas en el proyecto CCV y utilizando la documentación elaborada en el capítulo 1 se define una propuesta para el Levantamiento de Requisitos Funcionales en función del Aseguramiento de la Calidad de Software. Esta propuesta propone actividades y la puesta en práctica de técnicas, herramientas y estándares para el Levantamiento de Requisitos de Software en todo el desarrollo de sus etapas.

**Capítulo 3: Validación de la Propuesta:** Se realiza la revisión, valoración y evaluación de la propuesta para el levantamiento de requisitos en función del aseguramiento de la calidad de software, con el objetivo de validar la solución, y de conocer el criterio de especialistas en el tema que de alguna forma u otra han participado y adquirido conocimientos sobre la etapa de levantamiento de requisitos y el aseguramiento de la calidad de software. Se aplicó un cuestionario que permitió conocer dichos criterios y analizar los resultados de forma cuantitativa, generando estadísticas que pueden ilustrar de forma general la aceptación, adaptabilidad, importancia, necesidad, completitud, entre otros parámetros de evaluación definidos y evaluados por los especialistas.

**Para el desarrollo de la investigación utilizaron diferentes métodos entre los que se destacan los que se relacionan a continuación:**

Se utilizaron tres Métodos Científicos de la investigación a continuación se destaca la descripción de su uso:

**Histórico - Lógico:** Para determinar las necesidades históricas y tendencias actuales de la Ingeniería de Requerimiento y la Gestión de la Calidad en la elaboración de Software.

**Análisis y Síntesis:** Permite profundizar en la esencia del fenómeno "objeto de estudio", un análisis sobre las actividades de la Ingeniería de Requisitos, así como realizar una valoración acerca de los modelos, estándares y procedimientos de gestión de calidad existentes que nos permitiese ajustar de una forma coherente nuestra propuesta de solución.

**Inducción-deducción:** Posibilitó llegar a conclusiones acerca del problema investigado, a partir del estudio de la bibliografía consultada y del resultado de los instrumentos utilizados.

Los Métodos Empíricos utilizados fueron los que a continuación se especifican:

**Entrevista:** Se realizó a miembros de la dirección del proyecto con el objetivo de recopilar información sobre el proceso de Levantamiento de Requisitos y la gestión de los mismos en cuanto al Aseguramiento de la Calidad de software enfocada en las necesidades concreta de los proyectos.

**Cuestionario:** Se realizó a un grupo de especialistas en el tema, con el objetivo de validar la propuesta de solución, lo que permitió conocer su criterio y analizar los resultados de forma cuantitativa.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En la realización de un software es de vital importancia el control y seguimiento de la calidad. Cuando se comienza a desarrollar desde su primera etapa de Levantamiento de Requisitos (LR) es preciso mantener el equilibrio del mismo, para esto lleva un chequeo continuo para asegurar su calidad. En este capítulo es preciso efectuar una serie de estudios preliminares que den lugar a los objetivos trazados. Se realiza un estudio sobre la Ingeniería de Requisitos (IR) lo que permite un análisis de diferentes definiciones y métodos para realizar dicho proceso, técnicas y herramientas para realizar las actividades que se proponen en cada etapa. Se lleva a cabo el estudio de algunos modelos y estándares de calidad considerando los más importantes aplicados al desarrollo del software, todo para la obtención de un producto con calidad.

### Propósito de un Proyecto

“Un proyecto es un conjunto de acciones interrelacionadas y dirigidas a lograr unos de los resultados para transformar o mejorar una situación, es un plazo limitado y con recursos presupuestados”. (Palomar, 2006).

“Un proyecto es esencialmente un conjunto de actividades interrelacionadas, con un inicio y una finalización definida, que utiliza recursos limitados para lograr un objetivo deseado”. (Perisse, 2001).

La concepción de una idea de proyecto puede venir de diferentes puntos. Normalmente se trata de una idea de un equipo de desarrollo originada en los conocimientos adquiridos y puestos en práctica, o bien de una idea del usuario que encarga el proyecto. Una parte importante para la concepción y elaboración de la idea es la revisión y estudio de productos similares o cercanos al que se desea elaborar. No se trata de espiar, sino de analizar qué existe para obtener carencias y puntos fuertes consiguiendo con ello una propuesta más competitiva.

### 1.1. Proceso de desarrollo de software.

El proceso de ingeniería de software se define como “un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad” (Jacobson, y otros, 2002).

El proceso de desarrollo de software “es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo”. Concretamente “define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo”. (Jacobson, y otros, 2002).



Figura 1: Proceso de Desarrollo de Software

Uno de los objetivos de cualquier proyecto o mantenimiento de software, independientemente de su envergadura, es entregar un producto con calidad, esto significa que el software construido cumpla con las especificaciones establecidas, dentro de los tiempos, costos y recursos planificados o acordados. Para esto, se hace necesario que el equipo de trabajo adopte un proceso de desarrollo de software, es decir, un marco que defina las actividades necesarias para garantizar, técnica y administrativamente, que un software pueda ser construido o mantenido de manera organizada, disciplinada y previsible.

## 1.2. Requisitos dentro de las Metodologías de Desarrollo de Software.

### 1.2.1. ¿Qué es una Metodología?

Las metodologías de desarrollo de software abarcan todo el ciclo de vida del software, y se definen como “un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software (Barzanallana, 2008), adoptando una metodología de software se obtiene un producto más predecible y permite ciertas características deseables como:

- Existencias de reglas bien definidas.
- Verificaciones intermedias.
- Planificación y control.
- Comunicación efectiva.
- Utilización sobre un abanico amplio de proyectos.
- Fácil formación.
- Herramienta CASE (Computer Aided Software Engineering).
- Actividades que mejoren el proceso de desarrollo.
- Soporte al mantenimiento.
- Soporte de la reutilización de software.

De acuerdo a su definición de metodología, los procedimientos detallan consejos para elaborar una actividad; las técnicas serían la forma de ejecutar un procedimiento para obtener un resultado determinado; las herramientas software son las que hacen posible automatizar el proceso de

desarrollo del software y la documentación es la que identifica el software que se está desarrollando (Barzanallana, 2008).

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no se tiene en cuenta lo que propone la metodología de desarrollo de software puede provocar que los clientes y desarrolladores no queden satisfechos con el resultado obtenido. Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños de dos o tres meses. Lo que se hace con este tipo de proyectos es separar rápidamente el aplicativo en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo (Pressman, 2002). Cuando los proyectos que se van a desarrollar son de mayor envergadura, ahí sí toma sentido el basarse en una metodología de desarrollo, y empezar a buscar cuál sería la más apropiada para nuestro caso. Lo cierto es que muchas veces no se encuentra la más adecuada y se termina haciendo o diseñando nuestra propia metodología, algo que por supuesto no está mal, siempre y cuando cumpla con el objetivo (Jacobson, 2002).

Se realizó un estudio de varias metodologías para utilizar en el desarrollo de un software para luego poder seleccionar cuál es la más factible a aplicar. Se hace mención a cuatro metodologías de desarrollo de software de las que se consideraron más apropiada, partiendo de la metodología que se determinó en el proyecto en el cual fueron detectados los problemas antes expuesto, así como destacando de ellas la más importante:

- RUP
- XP
- MSF
- SCRUM

### **1.2.2. El proceso Unificado del Software. RUP.**

RUP (Process, 2005) es un proceso de desarrollo de software que ayuda a mejorar la productividad del equipo de trabajo, definiendo claramente sus actividades, roles y responsabilidades, desde los jefes de proyectos a los analistas y desde los desarrolladores a los probadores. RUP propone un conjunto de mejores prácticas a todos los miembros del equipo de proyecto como son: desarrollo de software iterativo (releases y versionamiento), utilización de arquitectura basada en componentes (diseño de arquitectura flexible), modelamiento de requisitos y diseño de software (con la utilización de UML), y verificación de la calidad.

RUP describe cómo aplicar efectivamente enfoques comprobados comercialmente para el desarrollo de software. Estos enfoques son llamados "mejores prácticas" pues son utilizados en la industria por organizaciones exitosas. RUP provee a cada miembro del equipo de las guías de proceso, plantilla

y de herramientas necesarias para que el equipo completo tome ventaja de las siguientes mejores prácticas:

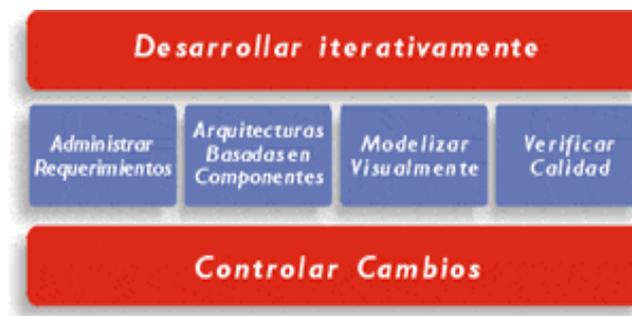


Figura 2: Desarrollo Iterativo de RUP.

### **Desarrollar software iterativamente:**

Es necesario un enfoque iterativo que permita una comprensión creciente del problema a través de refinamientos sucesivos, llegando a una solución efectiva luego de múltiples iteraciones acotadas en complejidad.

RUP utiliza y soporta este enfoque iterativo que ayuda a atacar los riesgos mediante la producción de releases ejecutables progresivos y frecuentes que permiten la opinión del usuario.

A través de las iteraciones que generan releases ejecutables se logra detectar en forma temprana los desajustes e inconsistencias entre los requerimientos, el diseño, el desarrollo y la implementación del sistema, manteniendo al equipo de desarrollo enfocado en producir resultados (Process, 2005).

### **Administrar los Requerimientos:**

Los requerimientos son las condiciones o capacidades que el sistema debe conformar. La Administración de Requerimientos es un enfoque sistemático para hallar, documentar, organizar y monitorear los requerimientos cambiantes de un sistema.

La Administración de Requerimientos permite:

- a) Que las comunicaciones estén basadas en requerimientos claramente definidos.
- b) Que los requerimientos puedan ser priorizados, filtrados y monitoreados.
- c) Que sea posible realizar evaluaciones objetivas de funcionalidad.
- d) Que las inconsistencias se detecten más fácilmente. (Pressman, 2005)

## **RUP describe:**

Las nociones de Casos de Uso y de Escenarios utilizadas en RUP han demostrado ser una manera excelente de capturar los requerimientos funcionales y asegurarse que direccionan el diseño, la implementación y la prueba del sistema, logrando así que el sistema satisfaga las necesidades del usuario (Process, 2005).

## **Utilizar arquitecturas basadas en componentes:**

El proceso de software debe focalizarse en el desarrollo temprano de una arquitectura robusta ejecutable, antes de comprometer recursos para el desarrollo en gran escala. RUP describe como diseñar una arquitectura flexible, que se acomode a los cambios, comprensible intuitivamente y promueve una más efectiva reutilización de software. Provee un enfoque sistemático para definir una arquitectura utilizando componentes nuevos y preexistentes (Process, 2005).

## **Modelizar software visualmente:**

RUP muestra cómo modelizar software visualmente para capturar la estructura y comportamiento de arquitecturas y componentes. Las abstracciones visuales ayudan a comunicar diferentes aspectos del software, comprender los requerimientos, ver cómo los elementos del sistema se relacionan entre sí, mantener la consistencia entre diseño e implementación y promover una comunicación precisa. El estándar UML (Lenguaje de Modelado Unificado), creado por Rational Software, es el cimiento para una modelización visual exitosa (Process, 2005).

## **Verificar la Calidad de Software:**

Es necesario evaluar la calidad de un sistema respecto a sus requerimientos de funcionalidad, confiabilidad y performance. La actividad fundamental es la prueba, que permite encontrar las fallas antes de la puesta en producción. RUP asiste en el planeamiento, diseño, implementación, ejecución y evaluación de todos estos tipos de prueba.

El aseguramiento de la calidad se construye dentro del proceso, en todas las actividades, involucrando a todos los participantes, utilizando medidas y criterios objetivos, permitiendo así detectar e identificar los defectos en forma temprana (Process, 2005).

## **Controlar los cambios al software:**

La capacidad de administrar los cambios es esencial en ambientes en los cuales el cambio es inevitable. RUP describe cómo controlar, rastrear y monitorear los cambios para permitir un desarrollo iterativo exitoso.

Es también una guía para establecer espacios de trabajo seguros para cada desarrollador, suministrando el aislamiento de los cambios hechos en otros espacios de trabajo y controlando los cambios de todos los elementos de software (modelos, código, documentos). Describe cómo automatizar la integración y administrar la conformación de releases (Process, 2005).

### **Fases de RUP.**

En RUP se han agrupado las actividades en grupos lógicos definiéndose nueve flujos de trabajo principales. Los tres últimos son utilizados de apoyo y los seis primeros son conocidos como flujos de ingeniería, entre los que se encuentra “Requerimientos” el cual define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen. (Rational, 2003)

Los objetivos del flujo de trabajo Requerimientos son:

- Definir el ámbito del sistema.
- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.
- Establecer y mantener un acuerdo entre clientes y otros involucrados sobre lo que el sistema debería hacer.
- Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- Proveer una base para estimar recursos y tiempo de desarrollo del sistema.
- Proveer una base para la planeación de los contenidos técnicos de las iteraciones.

Como parte de este flujo de trabajo las principales actividades que se realizan son:

- Identificar y clasificar requisitos.
- Encontrar actores y casos de uso.
- Priorizar casos de uso. Detallar casos de uso.
- Prototipos de interfaz de usuario.
- Estructurar el modelo de casos de uso.

Los requisitos se pueden clasificar en funcionales. Los funcionales son capacidades o condiciones que el sistema debe cumplir. Se mantienen invariables sin importar con qué propiedades o cualidades se relacionen.

### **1.2.3. Extreme Programming (XP)**

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo, un pequeño equipo y cuyo plazo de entrega es el mínimo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto (Coyner, y otros, 2005)



Figura 3: Extreme Programming (XP).

## Características de XP.

Pruebas Unitarias: se basa en las pruebas realizadas a los principales procesos, de tal manera que se adelante en algo hacia el futuro y que se puedan hacer pruebas de las fallas que pudieran ocurrir.

Re fabricación: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

Programación en pares: una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa (Coyner, y otros, 2005).

## ¿Qué es lo que propone XP?

El manejo del cambio se convierte en parte fundamental del proceso. El costo del cambio no depende de la fase o etapa, no introduce funcionalidades antes que sean necesarias. El cliente o el usuario se convierten en miembro del equipo. Saber el estado real y el progreso del proyecto para añadir, cambiar o quitar requerimientos en cualquier momento. Obtener lo máximo de cada semana de trabajo. Decidir cómo se implementan los procesos para crear el sistema con la mejor calidad posible. Pedir al cliente en cualquier momento aclaraciones de los requerimientos para poder estimar el esfuerzo para implementar el sistema y empieza en pequeño y añade funcionalidad con retroalimentación continua (Coyner, y otros, 2005).

### 1.2.4. Microsoft Solution Framework (MSF)

Esta es una metodología que se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Es flexible y se interrelaciona con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos (Torres, 2006).



Figura 4: Microsoft Solution Framework (MSF).

## MSF Características:

**Flexible:** Es utilizada en el ambiente de desarrollo de cualquier cliente.

**Adaptable:** Es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.

**Tecnología Agnóstica:** Porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

**Escalable:** Puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas o más.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo, de Proceso, de Gestión del Riesgo, de Diseño de Proceso y finalmente el de Aplicación (Torres, 2006).

### 1.2.5. SCRUM.

Es una metodología ágil de gestión de proyectos cuyo objetivo primordial es elevar al máximo la productividad de un equipo. Reduce al máximo la burocracia y las actividades no orientadas a producir software que funcione, y produce resultados en períodos muy breves de tiempo (cada 30 días). Sólo abarca prácticas de gestión sin entrar en las prácticas de desarrollo como puede hacer XP (Schwaber, 2005). Más bien delega completamente en el equipo la responsabilidad de decidir la mejor manera de trabajar para ser lo más productivos posibles. Sus raíces teóricas están en las teorías de la auto-organización.

Se puede decir que Scrum se basa en cierto “caos controlado”. El proceso de desarrollo de Scrum parte del conocimiento de que el proceso de desarrollo fundamental del producto está totalmente indefinido, es incluso caótico, pero establece ciertos mecanismos para controlar esta indeterminación, manipular lo impredecible y controlar la flexibilidad (Schwaber, 2005).

En la metodología Scrum se establecen tres fases: PRE-aplicación, aplicación y post-aplicación:

En la PRE – aplicación se definen y/o revisan las funcionalidades que ha de tener el sistema, haciendo la lista de funcionalidades que cuando estén completadas se podrá decir que el desarrollo del sistema está acabado. Dicha lista recibe el nombre de Release Back Log. Una vez hecho esto se elige un subconjunto de estas funcionalidades, llamado Sprint Back Log y entonces comienza la siguiente fase.

En la aplicación es donde se desarrolla el Sprint, las reglas de esta fase son sencillas, se distribuyen las tareas para cada miembro del equipo, se trabaja duro y se intenta conseguir el objetivo. Todos los miembros del equipo han de participar en una reunión diaria que en ningún caso deberá exceder los 30 minutos, llamada Sprint-meeting.

En esta reunión cada desarrollador debe dar respuesta a tres preguntas:

¿Qué hizo desde la última reunión?

¿Qué dificultades se están teniendo en el desarrollo de la tarea?

¿Qué va a hacer hasta la próxima reunión diaria?

En la post-aplicación se evalúa la entrega de funcionalidades del Sprint, se ven las tareas pendientes, se evalúa el progreso del proyecto y se redefine el tiempo de entrega del mismo si fuera necesario. En este paso se comparan las funcionalidades actuales con el Release Back Log y en el caso de no cumplirse se vuelve a la fase de PRE-juego para realizar una nueva iteración. Si por el contrario, sí se cumplen, entonces se pasa a la creación de la versión final y a la creación de la documentación pertinente (Schwaber, 2005).

### **1.2.6. ¿Por qué usar metodología de desarrollo RUP?**

El estudio de las metodologías existentes, fueron necesarias para decidir de acuerdo a las características de la propuesta de solución a desarrollar y demás factores que propician a la calidad de software, cual es la más conveniente, finalmente decidiéndose que la metodología idónea a seguir es RUP, ya que es una de las más aplicadas en la actualidad en cuestiones de proyectos grandes y complejos, por ser un proceso iterativo e incremental, flexible, que divide el trabajo en fases teniendo bien definidas las tareas a realizar en cada una de ellas, que establece roles de trabajo dentro del proyecto con el fin de establecer una organización del equipo de trabajo, asignando tareas y responsabilidades a cada uno de sus miembros.

### 1.3. Ingeniería de Requisitos.

La parte más difícil en la construcción de sistemas software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan dificultosa como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con humanos, máquinas y otros sistemas de software. Ninguna otra parte del trabajo puede perjudicar tanto el resultado final si es realizada de forma errónea. Ninguna otra parte es tan dificultosa de rectificar posteriormente". (BROOKS, 2003).

#### 1.3.1. ¿Qué es la Ingeniería de Requisitos (IR)?

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

"Ingeniería de Requerimientos es el proceso por el cual se transforman los requerimientos declarados por los clientes<sup>1</sup>, ya sean hablados o escritos, a especificaciones precisas, no ambiguas, consistentes y completas del comportamiento del sistema, incluyendo funciones, interfaces, rendimiento y limitaciones". (Guide, 1987).

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

Los requerimientos<sup>2</sup> pueden dividirse en requerimientos funcionales y requerimientos no funcionales. Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares.

La IR no es más que el proceso sistemático de desarrollar requisitos mediante un proceso iterativo y cooperativo de analizar el problema, documentar las observaciones resultantes en varios formatos de representación y comprobar la precisión del conocimiento obtenido (Christel), contando con cuatro etapas fundamentales:

---

<sup>1</sup> El "cliente" puede ser interpretado como un usuario de contabilidad, el grupo de mercadeo, otra organización interna o un cliente externo.

<sup>2</sup> En el documento se usa la palabra requisito como sinónimo de requerimiento.

**Elicitación de Requisitos:** Se refiere a la captura y descubrimiento de los requisitos. Esta etapa representa el comienzo de cada ciclo. Elicitación es el nombre comúnmente dado a las actividades involucradas en el descubrimiento de los requerimientos del sistema. Aquí, los analistas de requerimientos deben trabajar junto al cliente para descubrir el problema que el sistema debe resolver, los diferentes servicios que el sistema debe prestar, las restricciones que se pueden presentar. Es importante, que la elicitación sea efectiva, ya que la aceptación del sistema dependerá de cuán bien éste satisfaga las necesidades del cliente.

Esto no suele ser tarea fácil ya que se pueden presentar un conjunto de problemas como son:

- Muchas veces los clientes/usuarios no tienen una idea clara de sus necesidades reales.
- Diversas personas dentro de la organización tienen necesidades encontradas.
- Pueden existir limitaciones técnicas o tecnológicas para cumplir con algunos requerimientos.

Pero, en definitiva, descubrir los requerimientos del sistema no sólo implica preguntar a las personas que quieren: es un proceso delicado que involucra tener presente un conjunto de características:

- Comprender el dominio de aplicación, es decir, obtener un conocimiento del área general de la aplicación del sistema.
- Comprender el problema en sí, lo que implica que se debe extender y especializar el conocimiento sobre el dominio general para que se aplique al cliente en particular.
- Comprender el negocio, por tanto, se debe entender en profundidad cómo es que este sistema interactuará y afectará a las partes del negocio que estarán involucradas.
- Como puede contribuir a lograr las metas de la empresa o del lugar donde se implante el software finalmente.
- Comprender las necesidades y restricciones de los usuarios del sistema.
- Se deben entender los procesos de trabajo que se supone que el sistema apoyará y el rol de cualquier otro sistema que actualmente se involucre en dichos procesos.

Es importante, entonces, que la extracción sea efectiva, ya que la aceptación del sistema dependerá de cuán bien éste satisfaga las necesidades del cliente y de cuán bien asista a la automatización del trabajo.

**Análisis de Requisitos:** Sobre la base de la extracción realizada previamente, comienza esta etapa en la cual se enfoca en descubrir problemas con los requerimientos del sistema identificados hasta el momento. Usualmente se hace un análisis luego de haber producido un bosquejo inicial del documento de requerimientos; en esta etapa se leen los requerimientos, se conceptúan, se investigan, se intercambian ideas con el resto del equipo, se resaltan los problemas, se buscan alternativas y soluciones, y luego se van fijando reuniones con el cliente para discutir los requerimientos. Se debe destacar que no es posible convertir el análisis en un proceso estructurado y sistemático, lo que convierte a esta etapa en "subjetiva" porque depende en gran medida del juicio y de la experiencia del analista.

**Especificación de Requisitos:** En esta etapa se documentan los requerimientos acordados con el cliente, en un nivel apropiado de detalle. En la práctica, esta etapa se va realizando conjuntamente con el análisis, se puede decir que la especificación es el "pasar en limpio" el análisis realizado previamente aplicando técnicas y/o estándares de documentación, como la notación UML (Lenguaje de Modelado Unificado), que es un estándar para el modelado orientado a objetos, por lo que los casos de uso y la obtención de requerimientos basada en casos de uso se utiliza cada vez más para la obtención de requerimientos.

**Validación de Requisitos:** La validación es la etapa final de la IR. Su objetivo es, ratificar los requerimientos, es decir, verificar todos los requerimientos que aparecen en el documento especificado para asegurarse que representan una descripción, por lo menos, aceptable del sistema que se debe implementar. Se puede apreciar que el proceso de ingeniería de requerimientos es un conjunto estructurado de actividades, mediante las cuales se obtiene, se valida y se logra dar un mantenimiento adecuado al documento de especificación de requerimientos, que es el documento final, de carácter formal, que se obtiene de este proceso. Es necesario recalcar que no existe un proceso único que sea válido de aplicar en todas las organizaciones. Cada organización debe desarrollar su propio proceso de acuerdo al tipo de producto que se esté desarrollando, a la cultura organizacional, y al nivel de experiencia y habilidad de las personas involucradas en la ingeniería de requerimientos. Hay muchas maneras de organizar el proceso de ingeniería de requerimientos y en otras ocasiones se tiene la oportunidad de recurrir a consultores, ya que ellos tienen una perspectiva más objetiva que las personas involucradas en el proceso.

### 1.3.2. Características de los Requerimientos.

Las características de un requerimiento son sus propiedades principales. Un conjunto de requerimientos en estado de madurez, deben presentar una serie de características tanto individualmente como en grupo. A continuación se presentan las más importantes (Ralph, 2004).

**Necesario:** Un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser reemplazados por otras capacidades del producto o del proceso.

**Conciso:** Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.

**Completo:** Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.

**Consistente:** Un requerimiento es consistente si no es contradictorio con otro requerimiento.

**No ambiguo:** Un requerimiento no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.

**Verificable:** Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

### 1.3.3. Dificultades para definir los requerimientos.

Durante la etapa de especificación de requerimientos se pueden presentar muchos inconvenientes los cuales, son importantes identificar y prevenir. A continuación se presenta un listado con los problemas más comunes en este proceso: (Ralph, 2004)

- Los requerimientos no son obvios y vienen de muchas fuentes.
- Son difíciles de expresar en palabras (el lenguaje es ambiguo).
- Existen muchos tipos de requerimientos y diferentes niveles de detalle.
- La cantidad de requerimientos en un proyecto puede ser difícil de manejar.
- Nunca son iguales. Algunos son más difíciles, más riesgosos, más importantes o más estables que otros.
- Los requerimientos están relacionados unos con otros, y a su vez se relacionan con otras partes del proceso.
- Cada requerimiento tiene propiedades únicas y abarcan áreas funcionales específicas.
- Un requerimiento puede cambiar a lo largo del ciclo de desarrollo.
- Son difíciles de cuantificar, ya que cada conjunto de requerimientos es particular para cada proyecto.

### 1.3.4. Importancia de la Ingeniería de Requisitos. Beneficios.

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

- Permite gestionar las necesidades del proyecto en forma estructurada: Cada actividad de la IR consiste de una serie de pasos organizados y bien definidos.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: La IR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
- Disminuye los costos y retrasos del proyecto: Muchos estudios han demostrado que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro, especialmente aquellas decisiones tomadas durante la especificación de requisitos.
- Mejora la calidad del software: La calidad en el software tiene que ver con cumplir un conjunto de requerimientos (funcionalidad, facilidad de uso, confiabilidad, desempeño).
- Mejora la comunicación entre equipos: La especificación de requerimientos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.

- Evita rechazos de usuarios finales: La ingeniería de requerimientos obliga al cliente a considerar sus requerimientos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto. (Ralph, 2004)

### **1.3.5. Personal involucrado en la Ingeniería de Requerimientos.**

Realmente, son muchas las personas involucradas en el desarrollo de los requerimientos de un sistema. Es importante saber que cada una de esas personas tienen diversos intereses y juegan roles específicos dentro de la planificación del proyecto. El conocimiento de cada papel desempeñado asegura que se involucren a las personas correctas en las diferentes fases del ciclo de vida y en las diferentes actividades de la IR (Boehm, 1979).

No conocer estos intereses puede ocasionar una comunicación poco efectiva entre clientes y desarrolladores, que a la vez traería impactos negativos tanto en tiempo como en presupuesto. Los roles más importantes pueden clasificarse como sigue:

- Usuario final: Son las personas que usarán el sistema desarrollado. Ellos están relacionados con la usabilidad, la disponibilidad y la fiabilidad del sistema; están familiarizados con los procesos específicos que debe realizar el software, dentro de los parámetros de su ambiente laboral. Serán quienes utilicen las interfaces y los manuales de usuario.
- Usuario Líder: Son los individuos que comprenden el ambiente del sistema o el dominio del problema en donde será empleado el software desarrollado. Ellos proporcionan al equipo técnico los detalles y requerimientos de las interfaces del sistema.
- Personal de Mantenimiento: Para proyectos que requieran un mantenimiento eventual, estas personas son las responsables de la administración de cambios, de la implementación y resolución de anomalías. Su trabajo consiste en revisar y mejorar los procesos del producto ya finalizado.
- Analistas y programadores: Son los responsables del desarrollo del producto en sí, ellos interactúan directamente con el cliente.
- Personal de pruebas: Se encargan de elaborar y ejecutar el plan de pruebas para asegurar que las condiciones presentadas por el sistema son las adecuadas. Son quienes van a validar si los requerimientos satisfacen las necesidades del cliente.

Otras personas que pueden estar involucradas, dependiendo de la magnitud del proyecto, pueden ser: administradores de proyecto, documentadores, diseñadores de base de datos, entre otros (Boehm, 1979).

### **1.3.6. Levantamiento de Requisitos.**

Comprender los requisitos del cliente es fundamental para ejecutar un desarrollo de software exitoso, pero hay que tener en cuenta los problemas que pueden presentarse al respecto, ya que los usuarios

raramente tienen la habilidad para articular lo que realmente necesita. Las necesidades de los usuarios cambian cuando ellos ven el potencial del sistema y comprenden lo que el sistema puede hacer por ellos. Tanto la tecnología como el dominio cambian y mientras más tiempo tome el desarrollo, más cambiarán los requisitos. Por lo tanto, al intentar conocer todos los requisitos del producto al comienzo del proyecto no es factible a un costo razonable.

### **1.3.7. Técnicas utilizadas en las actividades de la IR.**

Existen varias técnicas propuestas para la IR (Herrera, y otros, 2003), de las cuales solo se abarcarán seis.

Es importante resaltar que estas técnicas pueden ser aplicables a las distintas fases del proceso de la IR, haciendo la salvedad de que hay que tomar en cuenta las características propias del proyecto en particular que se esté desarrollando para aprovechar al máximo su utilidad.

#### **Entrevistas.**

Las entrevistas y cuestionarios se emplean para reunir información proveniente de personas o de grupos. Durante la entrevista, el analista conversa con el encuestado. El cuestionario consiste en una serie de preguntas relacionadas con varios aspectos de un sistema. Por lo común, los encuestados son usuarios de los sistemas existentes o usuarios en potencia del sistema propuesto. En algunos casos, son gerentes o empleados que proporcionan datos para el sistema propuesto o que serán afectados por él. El éxito de esta técnica, depende de la habilidad del entrevistador y de su preparación para la misma.

#### **Sistemas Existentes.**

Esta técnica consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el sistema a ser construido. Por un lado, analizar las interfaces de usuario, observando el tipo de información que se maneja y cómo es manejada. Por otro lado también es útil analizar las distintas salidas que los sistemas producen (listados, consultas), porque siempre pueden surgir nuevas ideas sobre la base de estas.

#### **Lluvia de ideas (Brainstorm).**

Este es un modelo que se usa para generar ideas. La intención en su aplicación es la de generar la máxima cantidad posible de requerimientos para el sistema. No hay que detenerse en pensar si la idea es o no del todo utilizable. La intención de este ejercicio es generar, en una primera instancia, muchas ideas. Luego, se irán eliminando en base a distintos criterios como, por ejemplo, "caro", "impracticable", "imposible". Las reglas básicas a seguir son:

- Los participantes deben pertenecer a distintas disciplinas y preferentemente deben tener mucha experiencia. Esto trae aparejado la obtención de una cantidad mayor de ideas creativas.

- Conviene suspender el juicio crítico y se debe permitir la evolución de cada una de las ideas, porque si no se crea un ambiente hostil que no alienta la generación de ideas.
- Por más locas o salvajes que parezcan algunas ideas, no se las debe descartar, porque luego de maduras probablemente se tornen en un requerimiento sumamente útil.
- A veces ocurre que una idea resulta en otra idea, y otras veces podemos relacionar varias ideas para generar una nueva.
- Escribir las ideas sin censura.

### **Prototipos**

Durante la actividad de elicitación de requerimientos puede ocurrir que algunos requerimientos no estén demasiado claros o que no se esté muy seguro de haber entendido correctamente los requerimientos obtenidos hasta el momento, todo lo cual puede llevar a un desarrollo no eficaz del sistema final. Para validar los requerimientos hallados, se construyen prototipos. Los prototipos son simulaciones del posible producto, que luego son utilizados por el usuario final, permitiéndonos conseguir una importante retroalimentación en cuanto a si el sistema diseñado en base a los requerimientos recolectados le permite al usuario realizar su trabajo de manera eficiente y efectiva. El desarrollo del prototipo comienza con la captura de requerimientos. Desarrolladores y clientes se reúnen y definen los objetivos globales del software, identifican todos los requerimientos que son conocidos, y señalan áreas en las que será necesaria la profundización en las definiciones. Luego, tiene lugar un “diseño rápido” que se centra en una representación de aquellos aspectos del software que serán visibles al usuario (por ejemplo, entradas y formatos de las salidas) y que conlleva a la construcción de un prototipo.

### **Casos de Uso<sup>3</sup>**

Los casos de uso permiten describir la posible secuencia de interacciones entre el sistema y uno o más actores, en respuesta a un estímulo inicial proveniente de un actor. Los casos de uso es una descripción de un conjunto de escenarios, cada uno de ellos comenzado con un evento inicial desde un actor hacia el sistema. La mayoría de los requerimientos funcionales se pueden expresar con casos de uso. Según el autor Somerville, los casos de uso son una técnica que se basa en escenarios para la obtención de requerimientos. Actualmente, se han convertido en una característica fundamental de la notación UML (Lenguaje de Modelado Unificado), que se utiliza para describir modelos de sistemas orientados a objetos.

---

<sup>3</sup> Es una técnica para especificar el comportamiento de un sistema.

## Arqueología de Documento

Con la aplicación de esta técnica se tratan de determinar posibles requerimientos sobre la base de inspeccionar la documentación utilizada por la empresa; por ejemplo, manuales de procedimientos, reglamentos, boletas y facturas.

Esta técnica sirve más que nada como complemento de las demás técnicas, y ayuda a obtener información que de otra manera sería sumamente difícil conseguir. Por ejemplo, en las facturas se puede encontrar información que no se pensaba manejar y que en definitiva resulta de suma utilidad, como un número propio de la empresa que se utiliza para saber el orden que tiene la factura en la carpeta y que permite encontrar las copias del documento con mayor rapidez. En definitiva, se debe recolectar cualquier formulario o documento que sea utilizado para registrar o enviar información. Para el análisis de cada uno de estos documentos, se deben realizar algunas preguntas, como:

- ¿Cuál es el propósito de este documento?
- ¿Quién lo usa? ¿Por qué? ¿Para qué?
- ¿Cuáles son las tareas que realizan con este documento?
- ¿Se puede encontrar una relación entre los documentos?
- ¿Cuál es el proceso que realiza la conexión?
- ¿Cuál es el documento que da más problemas a los usuarios?

La utilización de las técnicas para la captura de Requisitos es de vital importancia para la obtención de información y para garantizar los resultados del sistema, logrando una mayor calidad en el transcurso de sus etapas.

## 1.4. Definiciones de Calidad de Software.

A continuación se darán algunas definiciones de Calidad de Software:

- "Conjunto de esfuerzos efectivos de los diferentes grupos de una organización para la integración del desarrollo, del mantenimiento y de la superación de la calidad de un producto, con el fin de hacer posible la fabricación y servicio a satisfacción completa del consumidor y al nivel más económico" (Feigenbaun, y otros, 2007).
- "La gestión de calidad en la empresa es el proceso de identificar, aceptar, satisfacer y superar constantemente las expectativas y necesidades de todos los colectivos humanos relacionados con ella, clientes, empleados, directivos, propietarios, proveedores y la comunidad con respecto a los productos y servicios que esta proporciona" (Andersen).
- "Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente" (Pressman., 1998)

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

De todas estas definiciones se extraen una serie de parámetros básicos que definen la calidad: si se desea generar productos y servicios de buena calidad para el consumidor será necesario decidir por adelantado la calidad de producto (o servicio), la planificación (calidad de diseño), la producción (calidad de fabricación) y la calidad que desea el cliente. Lo que está claro a partir de estas definiciones es que la calidad es algo relativo. Siempre va a depender de los requisitos o necesidades que se desee satisfacer. Por eso, la evaluación de la calidad de un producto siempre va a implicar una comparación entre unos requisitos preestablecidos y el producto realmente desarrollado.

### **1.4.1. ¿Qué es la Gestión de la Calidad de Software? Aspectos fundamentales.**

Todas las metodologías y herramientas tienen un único fin: producir software con calidad. Cada día más clientes se tornan compradores con conciencia de la calidad. Desean saber desde el comienzo que la empresa puede satisfacer sus necesidades. Un Sistema de Gestión de la Calidad certificado demuestra el compromiso de su empresa con la calidad y la satisfacción del cliente. Teniendo un Sistema de Gestión de Calidad se demuestra previsibilidad en las operaciones internas, así como capacidad para satisfacer los requisitos del cliente. Por otro lado, la gestión de la calidad permite una visión general que hace más fácil manejar, medir y mejorar los procesos internos.

Cada organización posee ciertos procesos operacionales que son críticos a la hora de conseguir sus objetivos estratégicos. Los beneficios de una organización dependen de su capacidad para descubrir puntos fuertes, débiles y oportunidades de mejora (Antonio).

#### **Gestión de la calidad (ISO)**

- ✓ Conjunto de actividades de la función general de la dirección que determina la calidad, los objetivos y las responsabilidades y se implanta por medios tales como la planificación de la calidad, el control de la calidad, el aseguramiento (garantía) de la calidad y la mejora de la calidad, en el marco del sistema de calidad.

#### **Política de calidad (ISO)**

- ✓ Directrices y objetivos generales de una organización, relativos a la calidad, tal como se expresan formalmente por la alta dirección.

La gestión de la calidad se aplica normalmente a nivel de empresa. También puede haber una gestión de calidad dentro de la gestión de cada proyecto.

## 1.4.2. ¿Qué es la Calidad de Software?

La calidad del software es el conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

La calidad del software es medible y varía de un sistema a otro o de un programa a otro. Un software elaborado para el control de naves espaciales debe ser confiable al nivel de "cero fallas"; un software hecho para ejecutarse una sola vez no requiere el mismo nivel de calidad; mientras que un producto de software para ser explotado durante un largo período (10 años o más), necesita ser confiable, mantenible y flexible para disminuir los costos de mantenimiento y perfeccionamiento durante el tiempo de explotación.

La calidad del software puede medirse después de elaborado el producto. Pero esto puede resultar muy costoso si se detectan problemas derivados de imperfecciones en el diseño, por lo que es imprescindible tener en cuenta tanto la obtención de la calidad como su control durante todas las etapas del ciclo de vida del software. (Lovelie)

## 1.4.3. Aseguramiento de la Calidad de Software.

El aseguramiento de calidad del software *es el conjunto de actividades planificadas y sistemáticas necesarias para aportar la confianza en que el producto (software) satisfará los requisitos definidos de calidad* (Lovelie).

- El aseguramiento de calidad del software se diseña para cada aplicación antes de comenzar a desarrollarla y no después.
- Algunos autores prefieren decir garantía de calidad en vez de aseguramiento.
  - ✓ Garantía, puede confundir con garantía de productos.
  - ✓ Aseguramiento pretende dar confianza en que el producto tiene calidad.
- El aseguramiento de calidad del software está presente en:
  - ✓ Métodos y herramientas de análisis, diseño, programación y prueba.
  - ✓ Inspecciones técnicas formales en todos los pasos del proceso de desarrollo del software.
  - ✓ Estrategias de prueba multiescala.
  - ✓ Control de la documentación del software y de los cambios realizados.
  - ✓ Procedimientos para ajustarse a los estándares (y dejar claro cuando se está fuera de ellos).
  - ✓ Mecanismos de medida (métricas).
  - ✓ Registro de auditorías y realización de informes.
- Actividades para el aseguramiento de calidad del software.

- ✓ Métricas de software para el control del proyecto.
- ✓ Verificación y validación del software a lo largo del ciclo de vida.
- Incluye las pruebas y los procesos de revisión e inspección.
  - ✓ La gestión de la configuración del software.

## **Control de la Calidad de Software.**

El control de la Calidad del Software son técnicas y actividades de carácter operativo, utilizadas para satisfacer los requisitos relativos a la calidad, centradas en dos objetivos fundamentales: (Lovellette)

- *Mantener bajo control un proceso.*
- *Eliminar las causas de los defectos en las diferentes fases del ciclo de vida.*

En general son las actividades para evaluar la calidad de los productos desarrollados.

## **Sistema de Calidad.**

- Los sistemas de Calidad son estructuras organizativas, procedimientos, procesos y recursos necesarios para implantar la gestión de calidad. Un sistema de calidad se debe adecuar a los objetivos de calidad de la empresa, donde ella es la responsable de fijar la política de calidad y las decisiones relativas a iniciar, desarrollar, implantar y actualizar el sistema de calidad. Un sistema de calidad consta de varias partes (Lovellette):

### **Documentación:**

- Manual de calidad: Es el documento principal para establecer e implantar un sistema de calidad. Puede haber manuales a nivel de empresa, departamento, producto, específicos (compras, proyectos).
- Parte física: Locales, herramientas, ordenadores.
- Aspectos humanos: Formación de personal, creación y coordinación de equipos de trabajo.

## **Plan de Calidad de Software.**

El software es inmaterial y la calidad del software difícil de medir, pero tenemos algunas pautas, algunos indicadores que nos ayudan a diferenciar los productos de calidad de los que carecen de ella:

- El acercamiento a cero defectos.
- El cumplimiento de los requisitos intrínsecos y expresos.
- La satisfacción del cliente.

Sobre todo la satisfacción del cliente. Ya se sabe que un suministrador puede engañar a todos alguna vez o a alguno muchas veces, pero no puede engañar a muchos durante largo tiempo. El cliente “casi” siempre tiene razón y para eso están las encuestas de satisfacción.

El argumento de la calidad es exhibido por las empresas como un factor diferenciador, como clave de sus procesos de negocio y como eslogan de competitividad empresarial. De hecho, la exigencia cada vez mayor por parte del mercado de la certificación ISO 9000 o el interés creciente por los modelos de calidad de gestión empresarial son indicadores de la percepción de la calidad como un elemento cada vez más necesario (Lovelley).

La calidad del software debe ser una disciplina más dentro de la Ingeniería del software. El principal instrumento para garantizar la calidad de las aplicaciones sigue siendo el Plan de Calidad. El plan se basa en unas normas o estándares genéricos y en unos procedimientos particulares.

Los procedimientos pueden variar en cada organización, pero lo importante es que estén escritos, personalizados, adaptados a los procesos de la organización y, lo que es más importante, que se cumplan. La calidad del software debe implementarse a lo largo de todo el ciclo de vida, debe correr paralela desde la planificación del producto hasta la fase de producción del mismo. Para ello se cuenta con una serie de ayudas, a las que el grupo dedica su atención y trabajo (Lovelley), a través de distintas actividades:

Para la fase de Planificación se pueden utilizar elementos y herramientas propias de la Gestión de proyectos, como la:

- Estimación de la duración, coste y esfuerzo para la construcción del producto. En lo referido a la estimación habrá que tener presentes las Métricas de software.
- Planificación de tareas que hay que realizar, asignación de personas, tiempo, coste y otros parámetros para construcción del producto.

Para los procesos de Análisis y Diseño se debe contar con:

- Herramientas CASE.
- Sistemas de obtención de requisitos.
- Métricas de software.
- Herramientas de Generación de Datos.
- Casos de Pruebas.

En los procesos de Construcción y Pruebas deberíamos utilizar:

- Herramientas de Gestión de la Configuración.
- Herramientas de Simulación.
- Casos de Pruebas.

Y, finalmente, para el proceso de Producción, básicamente habremos de utilizar:

- Herramientas de monitorización de resultados.
- Pruebas de producción.

Estos aspectos de la Calidad del Software constituyen el marco de trabajo en el que el grupo ATI Calidad del Software lleva a cabo sus actividades (Lovellette).

#### **1.4.4. Factores que determinan la Calidad de Software.**

Existen varios tipos de factores los cuales se clasifican en tres grupos.

##### **Operaciones del producto:**

Características operativas

- *Corrección* (¿Hace lo que se le pide?)

El grado en que una aplicación satisface sus especificaciones y consigue los objetivos encomendados por el cliente.

- *Fiabilidad* (¿Lo hace de forma fiable todo el tiempo?)

El grado en que se puede esperar en una aplicación que lleve a cabo las operaciones especificadas y con la precisión requerida.

- *Eficiencia* (¿Qué recursos hardware y software necesito?)

La cantidad de recursos hardware y software que necesita una aplicación para realizar las operaciones con los tiempos de respuesta adecuados.

- *Integridad* (¿Puedo controlar su uso?)

El grado con que puede controlarse el acceso al software o a los datos a personal no autorizado.

- *Facilidad de uso* (¿Es fácil y cómodo de manejar?)

El esfuerzo requerido para aprender el manejo de una aplicación, trabajar con ella, introducir datos y conseguir resultados.

##### **Revisión del producto:**

Capacidad para soportar cambios.

- *Facilidad de mantenimiento* (¿Puedo localizar los fallos?)

El esfuerzo requerido para localizar y reparar errores.

- *Flexibilidad* (¿Puedo añadir nuevas opciones?)

El esfuerzo requerido para modificar una aplicación en funcionamiento.

- *Facilidad de prueba* (¿Puedo probar todas las opciones?)

El esfuerzo requerido para probar una aplicación de forma que cumpla con lo especificado en los requisitos.

**Transición del producto:** adaptabilidad a nuevos entornos.

- *Portabilidad* (¿Podré usarlo en otra máquina?)

El esfuerzo requerido para transferir la aplicación a otro hardware o sistema operativo.

- *Reusabilidad* (¿Podré utilizar alguna parte del software en otra aplicación?)

Grado en que partes de una aplicación pueden utilizarse en otras aplicaciones.

- *Interoperabilidad* (¿Podrá comunicarse con otras aplicaciones o sistemas informáticos?)

El esfuerzo necesario para comunicar la aplicación con otras aplicaciones o sistemas informáticos (Lovellette).

### 1.5. Modelos y Estándares de Calidad.

Las áreas de Tecnologías de Información (TI) están sometidas a una incesante presión por parte de las áreas de negocio y del mercado en general (ITIL, 2008).

Existe un gran número de estándares de trabajo de buenas prácticas que pueden ayudar a la mejora de los procesos de TI: (ITIL, ISO, COBIT, CMMi).

- Sin embargo, hay que tener en cuenta que:
  - ✓ La implantación de estándares no es la solución para todos los problemas.
  - ✓ Ningún estándar cubrirá todas las necesidades.
  - ✓ Los estándares se pueden implantar de muchas maneras y deben adecuarse a cada problemática.

La mayoría de los estándares se han creado desde un específico punto de vista (auditoría y control), que tenemos que ver si están acorde o no en nuestra organización.

#### 1.5.1. ¿Qué es ITIL?

Information Technology Infrastructure Library (siglas en inglés ITIL). ITIL se ha convertido en uno de los principales facilitadores de una disciplina denominada IT Service Management (ITSM) para la gestión de los sistemas de información, cuya filosofía está centrada en la perspectiva de los clientes y la contribución que las tecnologías pueden hacer al negocio (ITIL, 2008).

Fue creado por la OGC (Office Government Commerce). El modelo muestra los objetivos, actividades generales, entradas y salidas de varios procesos IT, centrándose en la recomendación de mejores prácticas, sin entrar en el detalle que puede y debe ser distinto para cada organización.

Su foco central es la entrega y el soporte de servicios. Esto es principalmente en la operación de los sistemas. Otros procesos que incorpora son: Gestión de Infraestructura, Seguridad y Gestión de Aplicaciones. Implica a personas, procesos y tecnología.

### **ITIL puede aportar a las empresas:**

- Ayuda proporcionando mejores prácticas para la estandarización de los procesos.
- Subraya la importancia de implementar procesos en las organizaciones.
- Identifica las actividades para estos procesos.

### **Sin embargo no cubre una serie de aspectos importantes:**

- No cubre por igual todos los procesos IT.
- No hace referencia a los aspectos organizacionales.
- Debilidad a la hora de cubrir los aspectos de indicadores y de gobierno.

ITIL puede convivir y ser complementado con otra serie de estándares y framework como pueden ser:

- COBIT.
- CMMi.

### **1.5.2. ¿Qué es COBIT?**

COBIT proporciona a los gestores, auditores o usuarios de TI un conjunto de medidas, indicadores, procesos y mejores prácticas para maximizar los beneficios derivados del uso de las tecnologías de la información y desarrollando un apropiado control y gobierno de TI en las organizaciones. Fue creado por el Information System Audit and Control Association (ISACA) y por el IT Governance Institute en 1996 (ISO, 2008).

### **Como pueden convivir COBIT e ITIL.**

ITIL se enfoca en la eficacia y eficiencia de los procesos IT y la gestión de los mismos para mejorar la percepción del cliente, mientras que COBIT se enfoca más hacia la auditoría del cumplimiento de los procesos IT con los estándares de autoridades, el control sobre las funciones de IT, la medición y la gestión del riesgo. Por lo que se recomienda utilizarlos de manera conjunta y no tratar de comparar y elegir entre COBIT e ITIL, ya que tienen diferentes aproximaciones y pueden ser complementarios.

Por lo que se debe centrar ITIL para los temas de mejora de la eficiencia de los procesos y COBIT para controles de medición, métricas y gestión del riesgo.

### 1.5.3. ¿Qué es CMMi?

CMM establece un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Fue diseñado a finales de los ochenta por Software Engineering Institute (SEI) a instancias del Congreso Norteamericano, como medio para evaluar a las empresas suministradoras de software para el Departamento de Defensa Norteamericano (Gracia, 2005).

En el año 2000 fue integrado en el nuevo CMMi que se expande como aproximación para la mejora de procesos en otras áreas que complementan al desarrollo de software, incluyendo hasta 22 procesos distintos de IT.

- Está enfocado hacia la mejora continua de los procesos.
- Es bastante útil como herramienta de valoración.

#### Niveles de CMMi.

El CMM - CMMi es un modelo de Calidad del Software que clasifica las empresas en niveles de madurez. Estos niveles sirven para conocer la madurez de los procesos que se realizan para producir software.

Los niveles CMM - CMMI son 5:

#### Inicial o Nivel 1 CMM - CMMI.

Este es el nivel en donde están todas las empresas que no tienen procesos. Los presupuestos se disparan, no es posible entregar el proyecto en fechas. No hay control sobre el estado del proyecto, el desarrollo del proyecto es completamente opaco, no se sabe lo que pasa en él.

Es el típico proyecto en el que se da la siguiente situación:

- ¿Cómo va el proyecto?
- Bien, bien.

Dos semanas después...

- ¿Cómo va el proyecto?
- Bien, bien.

Tres semanas después...

- El lunes hay que entregar el proyecto.- No sé por qué pero los proyectos se entregan los lunes.
- El lunes! Todavía falta mucho!

- *¿Cómo? Me dijiste que el proyecto iba bien!! Arréglatelas como quieras, pero el proyecto tiene que estar terminado para el lunes.*

Si no sabes el tamaño del proyecto y no sabes cuánto lleva hecho, nunca sabrás cuando vas a terminar (Gracia, 2005).

### **Repetible o Nivel 2 CMM – CMMI.**

Quiere decir que el éxito de los resultados obtenidos se puede repetir. La principal diferencia entre este nivel y el anterior es que el proyecto es gestionado y controlado durante el desarrollo del mismo. El desarrollo no es opaco y se puede saber el estado del proyecto en todo momento. Los procesos que hay que implantar para alcanzar este nivel son: (Gracia, 2005)

- Gestión de requisitos.
- Planificación de proyectos.
- Seguimiento y control de proyectos.
- Gestión de proveedores.
- Aseguramiento de la calidad.
- Gestión de la configuración.

### **Definido o Nivel 3 CMM - CMMI.**

Resumiéndolo, este nivel significa que la forma de desarrollar proyectos (gestión e ingeniería) está definida, pero definida quiere decir que está establecida, documentada y que existen métricas (obtención de datos objetivos) para la consecución de objetivos concretos (Gracia, 2005).

Los procesos que hay que implantar para alcanzar este nivel son:

- Desarrollo de requisitos.
- Solución Técnica.
- Integración del producto.
- Verificación.
- Validación.
- Desarrollo y mejora de los procesos de la organización.
- Definición de los procesos de la organización.
- Planificación de la formación.
- Gestión de riesgos.
- Análisis y resolución de toma de decisiones.

La mayoría de las empresas que llegan al nivel 3 paran aquí, ya que es un nivel que proporciona muchos beneficios y no ven la necesidad de ir más allá porque tienen cubiertas la mayoría de sus necesidades (Gracia, 2005).

### **Cuantitativamente Gestionado o Nivel 4 CMM - CMMI.**

Los proyectos usan objetivos medibles para alcanzar las necesidades de los clientes y la organización. Se usan métricas para gestionar la organización (Gracia, 2005).

Los procesos que hay que implantar para alcanzar este nivel son:

- Gestión cuantitativa de proyectos.
- Mejora de los procesos de la organización.

### **Optimizado o Nivel 5 CMM - CMMI.**

Los procesos de los proyectos y de la organización están orientados a la mejora de las actividades. Mejoras incrementales e innovadoras de los procesos que mediante métricas son identificadas, evaluadas y puestas en práctica (Gracia, 2005).

Los procesos que hay que implantar para alcanzar este nivel son:

- Innovación organizacional.
- Análisis y resolución de las causas.

Normalmente las empresas que intentan alcanzar los niveles cuatro y cinco lo realizan simultáneamente ya que están muy relacionados. La implantación de un modelo de estas características es un proceso largo y costoso que puede costar varios años de esfuerzo. Aún así el beneficio obtenido para la empresa es mucho mayor que lo invertido.

#### **1.5.4. Modelo de Madurez de los Procesos.**

CMMi aporta además un modelo de madurez que puede aportar un método de evolución continua y de autoevaluación. Aunque ITIL y COBIT tiene Modelos de Madurez, el modelo de evaluación de la madurez de los procesos TI que se está imponiendo en la mayoría de los casos es el propuesto en CMMI.

IT Service - Capability Maturity Model (ITS-CMM) es el Modelo de Madurez que complementa al modelo de madurez para el desarrollo de software que ofrece CMMi y ofrece el mejor marco para la medición y evaluación del modelo de procesos IT (Model, 2008).

## 1.5.4.1. ITIL en conjunción con COBIT y CMMi.

Los tres estándares (ITIL, COBIT y CMMi) pueden convivir de manera conjunta.

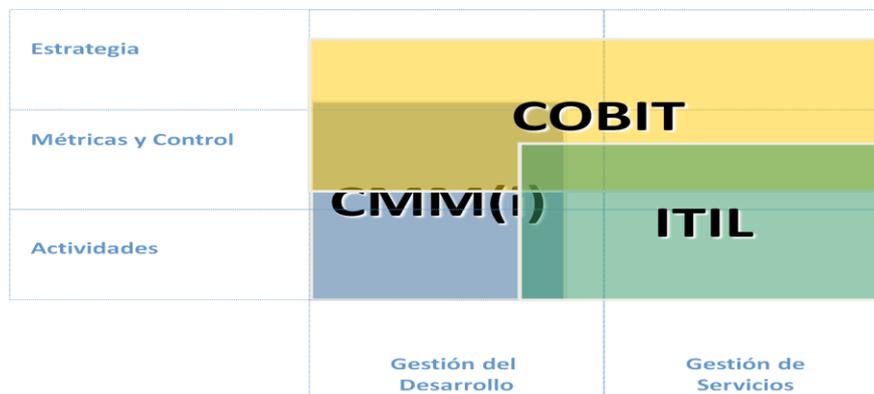


Figura 5: Estándar ITIL, COBIT y CMMi.

## 1.5.4.2. ¿Por dónde empezar y cómo ir evolucionando?

Utilizar **ITIL** como estándar de mejores prácticas para la definición y estandarización de los procesos de Gestión de Servicios IT.

Empezar con aquellos procesos **ITIL** más sencillos de implantar y que más impacto en cuanto a mejoras puedan aportar. Generalmente, gestión de incidencias y gestión de cambios.

Definir un modelo de control e indicadores basado en **COBIT** que permita realizar el control sobre el cumplimiento de los procesos y el valor aportado.

Utilizar los modelos **CMMi** e **ISO 27002** cuando sea necesario abordar los procesos relacionados con otros aspectos de desarrollo/mantenimiento de aplicaciones y seguridad de la información.

Alinear todos los procesos con las posibles iniciativas de calidad ya definidas en la organización (**ISO 9000**).

Establecer un modelo de autoevaluación y medición basado en **CMMi** que permita la mejora continua.

Iniciar los pasos y marcar objetivos para conseguir la certificación **ISO 20000**.

## 1.5.5. Requisitos dentro de los Estándares de Calidad.

Los Estándares de Calidad permiten definir un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software. Los estándares suministran los medios para que todos los procesos se realicen de la misma forma y son una guía para lograr la productividad y la calidad. A principios de los 80 surgió la necesidad de desarrollar estándares para la evaluación de la calidad de software. A nivel mundial, numerosas instituciones han desarrollado modelos y estándares en función de organizar y formalizar los procesos en la industria del software. Entre ellos, los más conocidos son el Modelo de Madurez de Capacidad Integrado (CMMI), las prácticas recomendadas por el Instituto

de Ingenieros Eléctricos y Electrónicos (IEEE) para la Especificación de Requisitos de Software y la Organización Internacional para la Estandarización (ISO).

### **ISO 9001:2000** (9001:2000)

ISO es un organismo que se dedica a publicar normas a escala internacional. La norma ISO 9001:2000 especifica los requisitos para un sistema de gestión de la calidad aplicable a toda organización cuando necesite demostrar su capacidad para proporcionar de forma coherente productos que satisfagan los requisitos del cliente y los reglamentarios aplicables; y aspira a aumentar la satisfacción del cliente a través de la aplicación eficaz del sistema, incluidos los procesos para la mejora continua del sistema y el aseguramiento de la conformidad con los requisitos del cliente y los reglamentarios aplicables. Plantea que en la determinación de los requisitos relacionados con el producto, la organización debe determinar los requisitos especificados por el cliente, incluyendo los requisitos para las actividades de entrega y las posteriores a la misma; los requisitos no establecidos por el cliente pero necesarios para el uso especificado o para el uso previsto, cuando sea conocido; los requisitos legales y reglamentarios relacionados con el producto y cualquier requisito adicional determinado por la organización.

La revisión de los requisitos relacionados con el producto, según la norma debe efectuarse antes de que la organización se comprometa a proporcionar un producto al cliente y debe asegurarse de que estén definidos los requisitos del producto, están resueltas las diferencias existentes entre los requisitos del contrato o pedido y los expresados previamente, y la organización tiene la capacidad para cumplir con los requisitos definidos. Define que deben mantenerse registros de los resultados de la revisión y de las acciones originadas por la misma.

Cuando el cliente no proporcione una declaración documentada de los requisitos, antes de la aceptación la organización debe confirmar los requisitos del cliente.

Cuando se cambien los requisitos del producto, la organización debe asegurarse de que la documentación pertinente sea modificada y de que el personal correspondiente conozca los requisitos modificados. En este estándar los requisitos se ven como necesidades o expectativas establecidas generalmente implícitas u obligatorias.

### **IEEE 830** (IEEE-STD-830-1998, 2000.)

IEEE 830 plantea que todos los requisitos que pertenecen a un producto del software no son igualmente importantes. Algunos requisitos pueden ser esenciales, sobre todo para las aplicaciones de vida crítica, mientras otros pueden ser deseables. Según la IEEE 830 los requisitos del proyecto representan una comprensión entre el cliente y el proveedor sobre materias contractuales que pertenecen a la producción de software y así no deben ser incluidos en la Especificación de

Requisitos de Software. Los requisitos del proyecto se especifican en documentos, generalmente en el plan de desarrollo del software o en una declaración de trabajo.

Este estándar recomienda que los requisitos específicos deban estar a un nivel de detalle suficiente para permitir a los diseñadores diseñar un sistema que satisfaga los requisitos. Los requisitos específicos en el desarrollo del software deben tener referencias a documentos más actuales que los relacionen, ser singularmente identificables y prestársele la atención debida a organizar los requisitos para aumentar al máximo la legibilidad.

### **ISO 9000:2000** (9000:2000, 2000)

La Norma ISO 9000:2000 describe los principios de los sistemas de gestión de la calidad y especifica la terminología de los sistemas de gestión de la calidad.

Plantea que los sistemas de gestión de la calidad pueden ayudar a las organizaciones a alcanzar la satisfacción del cliente. Los clientes necesitan productos con características que satisfagan sus necesidades y expectativas. Estas necesidades y expectativas se expresan en la especificación del producto y son generalmente denominadas como requisitos del cliente. Los requisitos del cliente pueden estar especificados por el cliente de forma contractual o pueden ser determinados por la propia organización. En cualquier caso, es finalmente el cliente quien determina la aceptabilidad del producto. Dado que las necesidades y expectativas de los clientes son cambiantes, las organizaciones deben mejorar continuamente sus procesos y productos. El enfoque del sistema de gestión de la calidad anima a las organizaciones a analizar los requisitos del cliente, definir los procesos que proporcionan productos aceptables para el cliente y a mantener estos procesos bajo control.

Un sistema de gestión de la calidad puede proporcionar el marco para la mejora continua con el objetivo de aumentar la probabilidad de alcanzar la satisfacción del cliente y de otras partes interesadas. Proporciona confianza tanto a la organización como a sus clientes de su capacidad para proporcionar productos que satisfagan los requisitos de forma consistente.

Los requisitos para los productos pueden ser especificados por los clientes o por la organización, anticipándose a los requisitos del cliente o por reglamentación. Los requisitos para los productos, y en algunos casos, los procesos asociados pueden estar contenidos por ejemplo en especificaciones técnicas, normas de producto, normas de proceso, acuerdos contractuales y requisitos reglamentarios.

### **1.6. La Calidad de Software desde la Gestión de Requisitos.**

De acuerdo a la definición más extendida, la calidad de un sistema o un producto de software están estrechamente relacionados con el cumplimiento o no de todos los requisitos. Compilando todos los requisitos dentro de una determinada documentación y orientando el proceso de desarrollo hacia el cumplimiento de estos requisitos, se logra un resultado de alta calidad.

Sin embargo, si la serie de requisitos de partida o la trazabilidad del desarrollo de los requisitos es incompleta, la calidad de los entregables será claramente deficiente.

A veces la atención se fija solamente en los requisitos funcionales, ignorando el hecho de que estos no representan más que un subconjunto de las expectativas que deben ser alcanzadas mediante las entregas del proyecto. Para ir más allá de esta apreciación, los requisitos deben incluir modelos de calidad reflejando varios estándares, criterios relevantes y restricciones resultantes de la producción o la logística en dependencia del proyecto en particular.

La suma de los requisitos describe el perfil del sistema que está bajo desarrollo. Así como los requisitos reflejan una multitud de necesidades, características, aplicaciones y funciones, muchos grupos de actores involucrados necesitan involucrarse en la elección, análisis y especificación de requisitos. Como añadido a la organización de requisitos, otras contribuciones pueden asegurar la calidad: marketing, ventas, fabricación y clientes.

#### **1.6.1. ¿Cómo obtener un software con calidad a partir del Levantamiento de Requisitos?**

La obtención de un software con calidad implica la utilización de metodologías o procedimientos estándares para el análisis, diseño, programación y prueba del software que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del software.

La política establecida debe estar sustentada sobre tres principios básicos: tecnológico, administrativo y ergonómico.

- El principio tecnológico define las técnicas a utilizar en el proceso de desarrollo del software.
- El principio administrativo contempla las funciones de planificación y control del desarrollo del software, así como la organización del ambiente o centro de ingeniería de software.
- El principio ergonómico define la interfaz entre el usuario y el ambiente automatizado.
- La adopción de una buena política contribuye en gran medida a lograr la calidad del software, pero no la asegura. Para el aseguramiento de la calidad es necesario su control o evaluación.

### 1.7. Conclusiones

Lograr el éxito en la producción de software es hacerlo con calidad. Esto sólo es posible con la implantación de una organización para el Aseguramiento de la Calidad del Software directamente relacionado con la política establecida para su elaboración desde la etapa temprana del Levantamiento de Requisitos.

En el presente capítulo se realizó un análisis de las características de las metodologías de desarrollo existentes a nivel mundial, como también las técnicas para el Levantamiento de Requisitos, la utilización de estándares de calidad y modelos de procesos y el uso de las herramientas para evaluar los indicadores de los artefactos generados. Es necesario destacar que cada técnica, herramienta o estándar tiene sus características y su forma de utilizar, por lo que se debe tener en cuenta la aplicación correcta de ellas.

Como resultado del estudio se determinó utilizar la metodología de desarrollo RUP por las características que esta presenta, de ser un proceso iterativo e incremental, flexible y que divide el trabajo en fases teniendo bien definidas las tareas a realizar en cada una de ellas.

Se identificaron las principales etapas que contempla la Ingeniería de Requisitos como son:

- Elicitación
- Análisis
- Especificación
- Validación

Se define los estándares de calidad de mayor reconocimiento en los temas relacionados con la Ingeniería de Requisitos como son ITL, COBIT y la IEEE 830. Como proceso de mejora continua CMMi para la Gestión de los Requisitos y el Aseguramiento de la Calidad de Software.

Luego de un estudio de toda la bibliografía consultada se determina que en la propuesta a desarrollar se definirán los roles, actividades, artefactos, técnicas, herramientas, estándares y revisiones de calidad que se deben tener en cuenta para la realización de un Levantamiento de Requisitos basados fundamentalmente en los planteamientos de la metodología de desarrollo RUP y adaptándola a las necesidades de los proyectos de la universidad.

### CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

Luego de haber analizado como se realizó el proceso de Levantamiento de Requisitos en el proyecto CCV y partiendo de los problemas que existieron durante el mismo, surge la necesidad de elaborar una Propuesta para el Levantamiento de Requisitos Funcionales en función del Aseguramiento de la Calidad de Software, la cual se abordará en el presente Capítulo.

Como se ha podido ver en el Capítulo anterior el proceso de Levantamiento de Requisitos es una etapa esencial para el desarrollo de un software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en las especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y completa el comportamiento del sistema, de esta manera se pretende minimizar los problemas relacionados al desarrollo de software asegurando la calidad del mismo en todas las etapas del flujo de trabajo de Requerimientos. La propuesta está basada en lo estudiado en el capítulo anterior, definiendo como base de la misma lo siguiente:

- La metodología a utilizar es RUP, producto a que es la más aplicada en la actualidad en cuanto a proyectos de gran envergadura, por ser un proceso iterativo e incremental, flexible, que divide el trabajo en varias fases, y la cual define un proceso para los flujos de trabajo de Modelamiento del Negocio y Requerimientos.
- Se hace un análisis de las actividades de la Ingeniería de Requisitos como son: Elicitación, Análisis, Especificación y Validación y se propone cómo realizar cada una de ellas.
- En cada actividad se proponen un conjunto de técnicas para el Levantamiento de Requisitos como son: Entrevistas, Lluvia de Ideas, Caso de Uso, Prototipos y Arqueología de Documentos, las cuales fueron explicadas en Capítulo 1 y que deben ser aplicadas con el objetivo de ayudar a la obtención de la información de la organización y validación de la información que se genere durante el proceso.
- Se hace uso de los estándares **ITIL**, **COBIT** y la **IEEE 830** para la obtención de información, el control y revisión de las etapas del Levantamiento de Requisitos y para la especificación de los requisitos, además del uso del modelo de proceso de **CMMi** para asegurar la calidad del mismo.

#### 2.1. Descripción de la Propuesta.

Para la realización de la Propuesta para el Levantamiento de Requisitos Funcionales en función del Aseguramiento de la Calidad de Software, se han definido cuáles son las actividades que se deben realizar en cada etapa de la Ingeniería de Requisitos, además el objetivo, los responsables de la realización de cada actividad, y el control de cada una de ellas para asegurar su calidad.

Se hace mención a las tareas que se realizan por cada una de las actividades definidas y los artefactos que se generan en las mismas.

La Propuesta abarcará como desarrollar las siguientes actividades de la Ingeniería de Requisitos:

**Elicitación:** Es la etapa de mayor interacción con el cliente. Es el momento en el que se recurre a la utilización de técnicas como: la observación, entrevistas y arqueología de documento para la captura de información para la modelación del negocio. Además de la identificación de los primeros requerimientos del sistema a desarrollar. También se realizan revisiones internas con el equipo de desarrollo y con el cliente de los artefactos generados.

**Análisis:** La etapa de análisis de los requisitos permite al analista representar el dominio de la aplicación a desarrollar, a través del uso de un lenguaje más técnico. Además se analizan los requisitos por separados, para identificar que no sean ambiguos, que sean consistentes, completos y verificables. También consiste en detectar y resolver conflictos entre requisitos y priorizar los mismos de acuerdo a su importancia.

**Especificación:** En esta actividad es donde se traducen los requisitos identificados en la etapa de análisis a un documento formal denominado Especificación de Requisitos. Luego se definen los casos de usos y los requisitos asociados a cada uno de ellos, realizando el documento Modelo de Casos de Uso del Sistema a través de la técnica de Casos de uso. También se definen los prototipos de interfaz de usuario. Estos documentos son revisados internamente por el equipo de desarrollo y por el cliente.

**Validación:** El objetivo de la Validación de Requisitos es descubrir problemas en el documento de Especificación de Requisitos y en el Modelo de Casos de uso del Sistema antes de comprometer recursos a su implementación. Estos documentos son revisados por el equipo de desarrollo una vez más; con el objetivo de descubrir omisiones o conflictos que no hayan sido descubiertos en etapas anteriores y validar con el cliente que los requisitos identificados satisfacen sus necesidades.

Para lograr un Levantamiento de Requisitos que tribute a la calidad del producto final, se debe proveer la Gestión de la Calidad de Software en cada una de estas actividades para lograr corregir los defectos de los requisitos desde los inicios y llegar a la validación con una menor cantidad de no conformidades, donde se haga más fácil el proceso de aceptación de los requisitos por parte de los clientes.

Este incluye las revisiones y las auditorías de los artefactos que se obtienen en esta etapa y de las actividades para verificar que ellas cumplen con los procedimientos y estándares que se establecen.

Buena parte de los problemas que surgen a lo largo del proceso de desarrollo del Levantamiento de Requisitos se deben a la carencia de un proceso adecuado de definición y entendimiento del problema y a la definición poco clara de las necesidades del cliente.

La importancia de la Ingeniería de Requisitos se comprueba en enfoques de mejoramiento del proceso como **CMMi**, que define la gestión y análisis de requisitos como unas de las prácticas básicas a la hora de evaluar el nivel de madurez de un proceso de desarrollo implantado en una organización, los estándares **ITIL** y **COBIT** para la captura de la información que se ha obtenido a través de los documentos entregados por el cliente y las auditorías y control de las pruebas a realizar por cada etapa del Levantamiento de Requisitos. Los modelos de calidad establecen objetivos generales, prácticas y metas que sirven de guía a la organización y que a su vez, ofrecen evidencias de que se tiene un proceso disciplinado y repetitivo. La organización, por su parte, debe implementar o adoptar los métodos, prácticas y herramientas adecuadas para llevar a cabo la compleja labor de Elicitación, Análisis, Especificación y Validación de los Requisitos de Software.

### 2.2. Objetivos de la Propuesta.

El objetivo que se persigue en esta Propuesta para el Levantamiento de Requisitos Funcionales en función del Aseguramiento de la Calidad de Software es definir los pasos para guiar el proceso de captura de la información y la entrega de la documentación tanto por la parte cliente como por el Equipo de Desarrollo. Es decir, definir y establecer el conjunto de etapas, actividades, tareas que se deben realizar como parte del proceso de Levantamiento de Requisitos, conjugando los artefactos, roles, técnicas, estándares y modelo de procesos que deben estar presentes en cada una con el fin de lograr el éxito, la aceptación por parte del cliente y que se logre un Levantamiento de Requisitos Funcionales con calidad.

Para esto nos apoyamos en los objetivos a desarrollar por parte de la calidad de software en el transcurso de todo el flujo de trabajo de Requerimientos. Se trata de controlar de cierto modo el cumplimiento de los procesos, las actividades y el mejoramiento del producto a entregar. Dichos objetivos son:

- Asegurar el cumplimiento de los estándares y procedimientos establecidos para el software.
- Asegurar la calidad del trabajo, a través de las revisiones y valoraciones sistemáticas en el proyecto durante el desarrollo de la etapa de Levantamiento de Requisitos.
- Mediante la Gestión de Riesgos identificar posibles errores antes de que se conviertan en puntos fatales o problemáticos.
- Realizar el levantamiento de requisitos sobre la base de los diferentes estándares y normas internacionales existentes.
- Velar porque los errores del flujo de trabajo de Requerimientos sean identificados a tiempo, garantizando que al finalizar dicha etapa existan la menor cantidad de errores posibles,

lográndose de esta forma un producto de mayor calidad y evitando que existan conflictos en etapas posteriores del proceso de desarrollo de software.

Lograr que el equipo de desarrollo cuente con el personal capacitado con el conocimiento y las habilidades necesarias para realizar las tareas y actividades encaminadas a lograr la calidad del producto.

### 2.3. Flujo de la Propuesta.

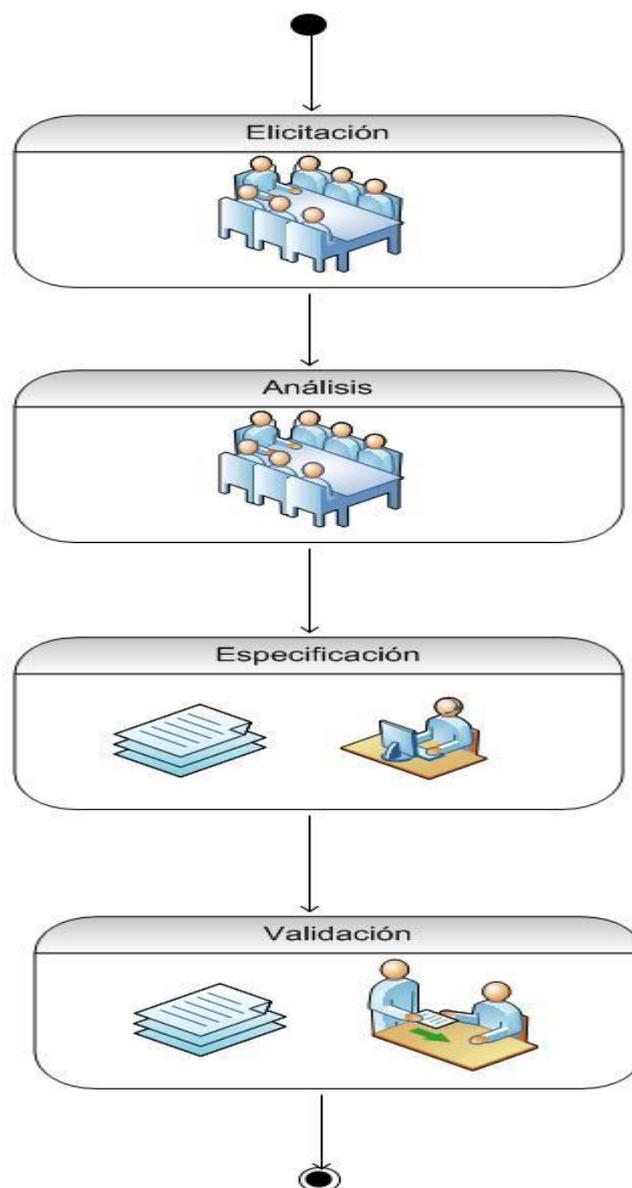


Figura 6: Imagen del Flujo de la Propuesta.

### 2.4. Roles involucrados en la Propuesta.

A continuación se presenta la definición de los roles involucrados en el proceso de Levantamiento de Requisitos por parte del Cliente.

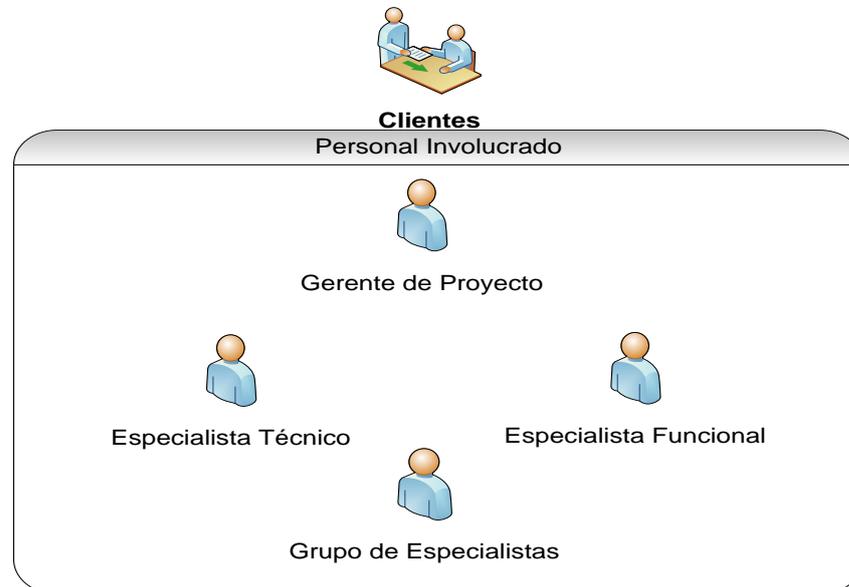


Figura 7: Roles involucrados por parte del Cliente.

#### Tareas y Responsabilidades por parte del Cliente.

**Nombre:** Gerente de Proyecto

**Descripción:** Es la persona que va a representar el equipo de desarrollo por la parte del cliente. Es el encargado de organizar y administrar todos los recursos que se emplean en el desarrollo del proyecto, de tal manera que se pueda terminar el trabajo requerido dentro del tiempo establecido, así como los costes definidos en el mismo. Se esfuerza por mantener el progreso y la interacción mutua de ambas partes en las esferas de producción. No actúa directamente en la producción de los resultados del software.

#### Responsabilidades:

- Gestionar los recursos humanos necesarios en la ejecución de las actividades por el cliente en el desarrollo del proyecto.
- Lograr el cumplimiento de las actividades planificadas por el Equipo de Desarrollo.

**Nombre:** Especialista Técnico

**Descripción:** Es la persona encargada de dirigir al equipo técnico y tomar las decisiones referentes al desarrollo del proyecto en este sentido. Debe tener un conocimiento sobre la metodología a utilizar y las herramientas y técnicas a desarrollar en el proyecto.

### **Responsabilidades:**

- Gestionar los recursos humanos necesarios para las actividades a desarrollar, desde el punto de vista técnico, en el transcurso de la construcción del sistema.
- Garantizar el cumplimiento de las actividades solicitadas por el Equipo de Desarrollo en las áreas de desarrollo técnico.
- Tomar decisiones referentes a la parte técnica del software.
- Dirigir y definir a los especialistas en cuanto a las responsabilidades que tienen dentro del Equipo de Desarrollo.

**Nombre:** Especialista Funcional.

**Descripción:** Es un especialista capacitado en esferas determinadas del negocio. Es la persona que sirve de intermedio con el Equipo de Desarrollo para aclarar cualquier duda que pueda surgir en el negocio.

### **Responsabilidades:**

- Gestionar recursos humanos en la parte funcional que sean necesarios en la ejecución de las actividades realizadas por el cliente.
- Garantizar el cumplimiento de las actividades solicitadas por el Equipo de Desarrollo en las áreas de desarrollo funcional.
- Tomar decisiones referentes a la parte funcional del software.
- Participar en las actividades realizadas por el cliente y el Equipo de Desarrollo como intermedio, con el objetivo de aclarar cualquier duda en el tema de negocio.

**Nombre:** Grupo Especialista

**Descripción:** Es un grupo de personas integrados por especialistas funcionales y especialistas técnicos con conocimientos en la parte de tecnología del proyecto y especializadas en las diferentes áreas de organización, con un conocimiento amplio del negocio. Son las personas encargadas de validar y aceptar los documentos que se generan durante el Levantamiento de Requisitos.

### **Responsabilidades:**

- Facilitar información técnica del proyecto, así como las necesidades y características en el área de sistemas, bases de datos y seguridad.
- Facilitar información referente a las normas, políticas y procedimientos en la implementación y definición de los accesos de los usuarios.

- Facilitar información sobre el negocio a automatizar, así como de las características del sistema a desarrollar.
- Intervenir entre el Equipo de desarrollo y los trabajadores por la parte del cliente para el desarrollo de las actividades.

A continuación se presenta la definición de los roles involucrados en el proceso de Levantamiento de Requisitos por parte del Equipo de Desarrollo.

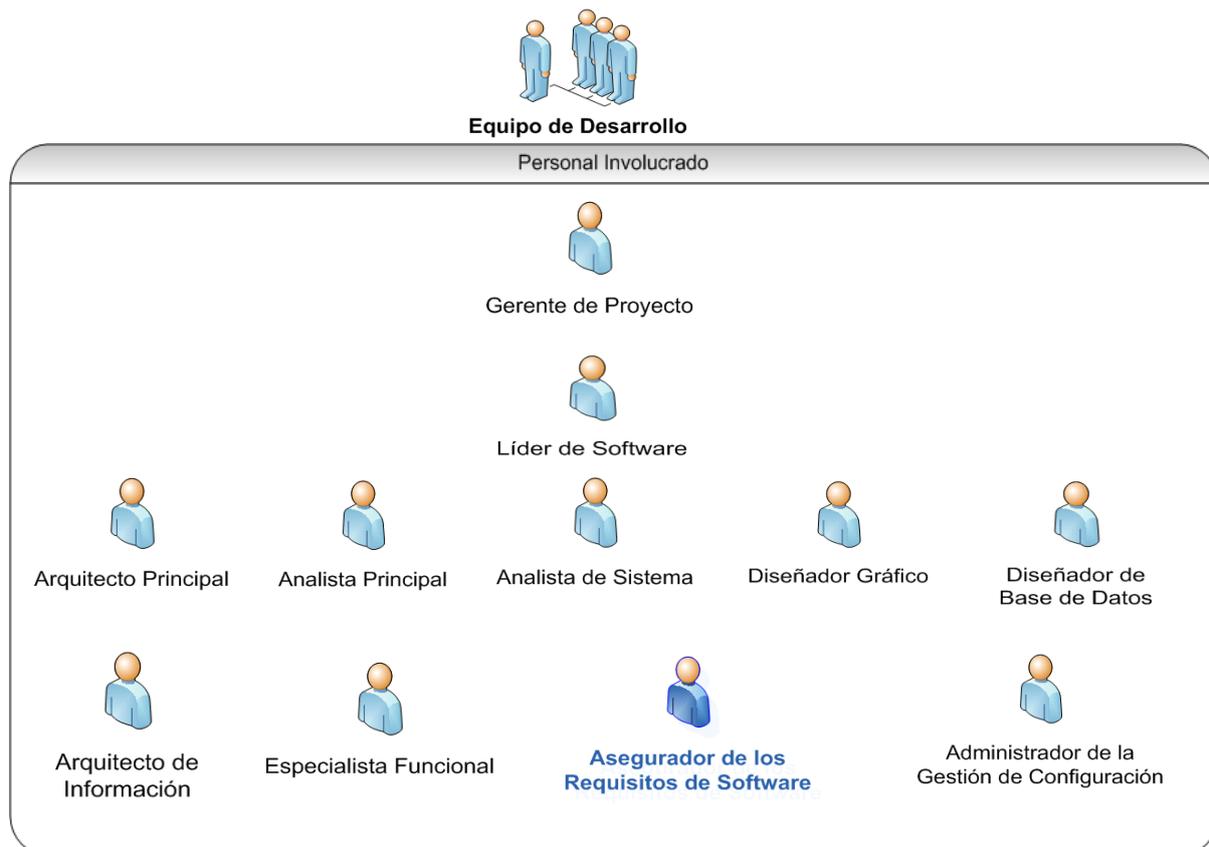


Figura 8: Roles involucrados en el Equipo de Desarrollo.

### Tareas y Responsabilidades por parte del Equipo de Desarrollo.

**Nombre:** Gerente del Proyecto.

**Descripción:** Es la persona encargada de representar al equipo de desarrollo, organizar y administrar todos los recursos con que cuenta su equipo, de tal manera que pueda terminar su trabajo en el proyecto, así como su alcance, tiempo y costes definidos. Su esfuerzo radica en mantener el progreso del desarrollo del software. No interactúa directamente en la producción de los resultados.

### **Responsabilidades:**

- Aprobar y firmar el Plan de Gestión de Levantamiento de Requisitos.
- Gestionar los recursos humanos necesarios en la ejecución de las actividades por el Equipo de Desarrollo.
- Controlar y dirigir el proyecto, evaluando todas las actividades que se realizan en el mismo.
- Velar y garantizar la organización y la disciplina del equipo de trabajo.
- Tomar todas las decisiones del proyecto, siendo responsable de que todas se cumplan en el tiempo y período establecido.

**Nombre:** Líder de Software.

**Descripción:** Se encarga de dirigir al Equipo de Desarrollo en todo el transcurso del ciclo de vida del proyecto. Está capacitado para tomar las decisiones del equipo de trabajo.

### **Responsabilidades:**

- Aprobar el Plan de Levantamiento de Requisitos.
- Mantener la gestión y el control del avance del software.
- Participar en las actividades de validación y aprobación de las propuestas del software.
- Definir la metodología a utilizar en el proyecto.
- Definir y aprobar las tecnologías a usar en el desarrollo del proyecto.
- Coordinar y organizar las tareas que se asignan a los miembros del Equipo de Desarrollo.
- Gestionar los recursos y materiales necesarios tanto para el proyecto como para el Equipo de Desarrollo.
- Llevar a cabo todo el proceso de Gestión de Proyecto.

**Nombre:** Arquitecto Principal.

**Descripción:** Es la persona responsable de priorizar los casos de uso además de la arquitectura del software, la cual incluye las decisiones técnicas claves que conllevan a la mayor parte del diseño.

### **Responsabilidades:**

- Priorizar los casos de uso. Definir la selección del conjunto de escenarios y casos de uso que serán analizados en la iteración de acuerdo a su prioridad.
- Definir una arquitectura candidata para el sistema, basada en la experiencia obtenida de sistemas similares, además de definir patrones de arquitectura y mecanismos claves.

- Participar en la elección de la metodología de desarrollo a utilizar en el software.

**Nombre:** Analista Principal.

**Descripción:** Es la persona encargada de guiar a los demás analistas del sistema. Es responsable de las actividades del especificador de requisitos, de definir los casos de uso y actores del sistema y cómo ellos interactúan; además es el responsable del recogimiento de los requerimientos de usabilidad y diseñar la interfaz de usuario candidata para conocer estos requerimientos.

**Responsabilidades:**

- Controlar y supervisar los analistas del sistema, así como el cumplimiento de la metodología a utilizar.
- Planificar las actividades de la etapa de la captura de Requisitos.
- Definir actores y casos de uso que interactúan en el sistema.
- Identificar los requisitos que el sistema debe cumplir.
- Definir los prototipos de interfaz de usuario.

**Nombre:** Analista de Sistema.

**Descripción:** Persona al frente de un módulo del proyecto, encargada de documentar y modelar los procesos del negocio y el sistema encontrados en la captura de requisitos. Debe tener dominio de la metodología que se está utilizando.

**Responsabilidades:**

- Participar en las definiciones que se realizan en el proyecto.
- Realizar modelación del negocio y del sistema.
- Detallar los requerimientos de software con el objetivo de recolectar y organizar el conjunto de artefactos que describen completamente los requerimientos de software del sistema.
- Extraer el comportamiento de los casos de usos del sistema que necesitan ser considerados como casos de uso abstractos.
- Traducir el lenguaje entre el cliente y el equipo de desarrollo.

**Nombre:** Diseñador Gráfico.

**Descripción:** Es el encargado de obtener toda la información necesaria para realizar el diseño gráfico de la aplicación del software.

**Responsabilidades:**

- Captar información del cliente para diseñar los gráficos de la aplicación de manera tal que cumplan con los requisitos del cliente.

**Nombre:** Diseñador de Base de Datos.

**Descripción:** Es el responsable de diseñar la forma óptima de almacenar los datos persistentes a ser usados por el sistema, así como identificar todas las herramientas necesarias para la modelación de la base de datos y administración de las mismas.

**Responsabilidades:**

- Diseñar la base de datos.
- Asegurar que los datos persistentes son almacenados consistente y eficientemente, además de definir el comportamiento que debe ser implementado en la base de datos.

**Nombre:** Arquitecto de Información.

**Descripción:** Esta persona estará presente para realizar toda la captura de información correspondiente en el Levantamiento de Requisitos.

**Responsabilidades:**

- Definir y establecer la audiencia y estudiar los escenarios de uso.
- Definir los Contenidos del Sistema, agrupando y etiquetando todo el contenido, así como la identificación de los requerimientos funcionales y el análisis de sistemas similares.
- Definir la Estructura del Sistema, encargándose de la creación de la estructura, mapas del sistema, definiciones y características de los sistemas de navegación.
- Definir los elementos de los Sistemas de Navegación ya sea su definición del diseño visual, diseño de las estructuras de páginas, bocetos de diseño y los borradores de página.

**Nombre:** Especialista Funcional.

**Descripción:** Es la persona encargada de intervenir entre el cliente y el equipo de desarrollo, donde será un especialista en las áreas del negocio que estará capacitado para aclarar cualquier duda entre ambas partes.

**Responsabilidades:**

- Brindar apoyo en el Levantamiento de Requisitos.
- Apoyar al equipo de desarrollo en las esferas del negocio, esclareciendo dudas que puedan surgir al respecto.
- Participar en las revisiones continuas que se realizan por cada etapa del flujo de trabajo de Requerimientos.

**Nombre:** Asegurador de Requisitos de Software.

**Descripción:** Es la persona encargada de asumir el rol de asegurador de la calidad y revisor técnico de los requisitos de software. Realizará las pruebas continuas en toda la etapa de desarrollo del Levantamiento de Requisitos, así como llevar el control de la utilización de los estándares de calidad utilizados.

**Responsabilidades:**

- Realizar un Plan de Revisiones en función del Aseguramiento de la Calidad de Software, con el fin de guiar los pasos y pruebas en dicha etapa.

- Realizar el control y revisiones de las pruebas realizadas en la etapa del Levantamiento de Requisitos.
- Aplicar los estándares de calidad a la documentación, las revisiones y pruebas realizadas durante la etapa del Levantamiento de Requisitos.
- Aplicar las Listas de Chequeo a la documentación que se estará revisando.
- Ejecutar las pruebas con el objetivo de evaluar la calidad de los documentos elaborados durante el Levantamiento de Requisitos.
- Elaborar el documento Informe de No Conformidades, con el objetivo de crear un documento con el resultado de las revisiones realizadas a la documentación generada durante el levantamiento de requisitos.

**Nombre:** Administrador de la Gestión de la Configuración.

**Descripción:** Es el responsable de suministrar la completa infraestructura de administración de la configuración y ambiente al equipo de desarrollo.

**Responsabilidades:**

- Gestionar la administración de la configuración con el objetivo de establecer un ambiente donde el producto pueda ser construido y desarrollado, esto implica la asignación de recursos de máquina (servidores y espacio en discos) e instalar las herramientas para esta configuración, además de la creación de repositorios.
- Elaborar el Plan de Administración de la Configuración, que tiene como objetivo, describir todas las actividades de configuración y control de cambio a ser ejecutadas durante el ciclo de vida del proceso de desarrollo. Documentar cómo las actividades de la configuración relacionadas con el producto son planeadas, implementadas, controladas y organizadas.

### 2.5. Documentos que se generan en la Propuesta.

En cada una de las actividades que se realizan durante las etapas del Levantamiento de Requisitos, se genera un conjunto de artefactos que son de vital importancia para su realización, de los cuales se hace una breve descripción para su mayor entendimiento.

Artefactos de entradas o salidas durante cada una de las etapas de la Propuesta:

- *Cronograma:* Documento en el cual estará registrada una planificación de las actividades a realizar durante el Levantamiento de Requisitos.
- *Plan de Revisiones:* Documento donde se registra el proceso de revisión que se realizará durante el Levantamiento de Requisitos (Ver Anexo 1).

- *Guía de preguntas*: Documento que tiene como objetivo definir un conjunto de preguntas que sirvan de guía al equipo de desarrollo durante el estudio de la base de documentación para la extracción de información.
- *Documento de Revisión Técnica Formal (RTF)*: Documento que tiene como propósito planificar las reuniones RTF y que los acuerdos, resultados de las pruebas y decisiones de la reunión queden resumidos (Ver Anexo 2).
- *Minuta de Entrega*: Documento estratégico que recoge una lista con todos los documentos entregados tanto por la parte cliente como por el equipo de desarrollo y la firma de cada una de las partes involucradas en el proyecto como constancia de que los documentos fueron recibidos. (Ver Anexo 3).
- *Base de Documentación*: Es toda la documentación que la organización pueda facilitar como por ejemplo: manuales de procedimientos y leyes.
- *Glosario de Términos*: Recoge los términos, abreviaturas, frases, palabras que son propios de la organización y del entorno del negocio que puedan resultar ambiguos a la hora de interpretarlas. (Ver Anexo 4).
- *Documento de los procesos del negocio*: Documento que representa cada uno de los procesos del negocio identificados; así como la descripción de cada uno de ellos (Ver Anexo 5).
- *Listado de Requisitos Funcionales*: Documento que refleja el listado de todos los requisitos identificados a partir de la modelación de los procesos del negocio y de la elicitación realizada con el cliente y que es actualizado en la etapa de Análisis (Ver Anexo 6).
- *Lista de Riesgos*: Documento que recoge el listado de todos los posibles riesgos identificados que pueden afectar el proceso de Levantamiento de Requisitos (Ver Anexo 7).
- *Listas de Chequeo*: Documentos que evalúan la especificación de cada artefacto generado durante el levantamiento de requisitos y que sirven como herramienta para detectar posibles defectos durante esta etapa (Ver Anexo 8).
- *Informe de No Conformidades*: Documento que recoge los defectos encontrados en las pruebas realizadas a la documentación generada durante el levantamiento de requisitos (Ver Anexo 9).
- *Documento Visión*: Define la visión del cliente sobre el producto a desarrollar, especificado en términos de las principales necesidades del cliente y características del negocio. Contiene una lista inicial de los requisitos y el alcance del proyecto (Ver Anexo 10).

- *Documento de Especificación de los Requisitos de Software*: Documento donde se realiza una especificación detallada de todos los Requisitos identificados en la etapa de Elicitación y Análisis (Ver Anexo 11).
- *Registro de Revisiones*: Documento donde se recogen todas las pruebas realizadas a los artefactos generados y los resultados de las mismas (Ver Anexo 12).
- *Minuta de Aceptación*: Documento estratégico que recoge los puntos tratados en cada encuentro y los acuerdos como resultado final. Plasma la aceptación de los clientes con la información presentada por el Equipo de desarrollo validando la misma (Ver Anexo 13).
- *Modelo de Caso de Uso del Sistema*: Presenta las funcionalidades del sistema a un alto nivel; mediante la presentación de los subsistemas o módulos, y de los casos de uso que serán implementados, además de representar los prototipos de interfaz de usuario (Ver Anexo 14).

### 2.6. Estándares y Modelo de procesos utilizados en la Propuesta.

La utilización de estándares en el proceso de Levantamiento de Requisitos es de vital importancia, lo cual no quiere decir que dará toda la solución a los problemas que se puedan presentar en esta etapa y que cubrirá todas las necesidades que se impongan en el software, pero los estándares sí se pueden implantar de manera tal que se adapten a la problemática de cada solución.

Los estándares que se proponen a utilizar son: **ITIL**, **COBIT** y **IEEE 830**.

¿Por dónde empezar e ir evolucionando en el transcurso de la etapa de Levantamiento de Requisitos?

El estándar **ITIL** debe ser utilizado para la definición y estandarización de los procesos de Gestión de Servicios en el Levantamiento de Requisitos. Principalmente se recomienda como guía esencial para crear y mantener el valor para los clientes a través de un mejor diseño, introducción y operación de los servicios. Combina los principios, prácticas y métodos de la administración de calidad. Dicha filosofía va enmarcada en las perspectivas del cliente y las contribuciones que pueden brindar al negocio del sistema; por lo tanto, es necesario para hacer el levantamiento de información, ya que sirve como guía para definir las actividades del proceso y la organización del equipo.

- Como apoyo al estándar **ITIL** es necesario unirlo con el estándar **COBIT**, ya que a través de este último, se puede definir un modelo de revisión que permita realizar el control sobre el cumplimiento de todos los procesos y el valor aportado en la etapa de Levantamiento de Requisitos.

¿Qué aporta?

Un seguimiento y control de las revisiones, así como de las auditorías, que ayudan a la mejora continua del proceso de Levantamiento de Requisitos, acompañado a las necesidades del cliente.

Conjuntamente, **COBIT** es capaz de mantener una medición de los requisitos obtenidos y la gestión de los riesgos que estos pueden traer consigo.

- Utilizar el estándar **IEEE 830** para la especificación de los requisitos de software con el fin de lograr un mayor nivel de detalle que permita a los diseñadores diseñar un sistema que satisfaga los requisitos, como también tener referencias a los documentos que los relacionen para ser identificables y tener una mayor legibilidad.

Como modelo de proceso a utilizar: **CMMi**.

Utilizar el modelo **CMMi**, para la mejora de proceso, trabajando con lo que plantea el Nivel 2 (Repetitivo) de madurez, como uno de los procesos que abarca la Gestión de Requisitos y el Aseguramiento de la Calidad de Software. Por lo tanto se podrá obtener una autoevaluación y medición que permita la mejora continua de los procesos.

### 2.7. Técnicas y Herramientas a utilizar en la Propuesta.

Como técnicas a utilizar en el Levantamiento de Requisitos se proponen las siguientes:

- **Entrevistas:**

Las entrevistas se utilizan con el objetivo de identificar los interesados en el sistema y las características generales del negocio que se desea informatizar, además de poder descubrir cuál es el alcance del proyecto y quiénes son los beneficiados. Se trata de que las preguntas conlleven a una respuesta abierta por parte del cliente para tratar de recopilar la mayor cantidad de información.

- **Lluvia de Ideas:**

Esta técnica es de vital importancia para la generación de múltiples ideas ya que intervienen diferentes personas en la actividad y además posibilita alcanzar la máxima cantidad posible de requerimientos para el sistema. No se debe pensar si la idea que surge es correcta o no, es simplemente aplicarla y de ahí derivar otras, hasta llegar a su objetivo, con la intención de ir recopilando ideas, y luego formar una, eliminando las que no son claras. Para esta técnica se recomienda la participación de personas con experiencia en el tema, para que surjan la mayor cantidad posible de ideas y que a su vez sean factibles.

- **Prototipos:**

Esta es una técnica que se utiliza en la actividad de especificación durante la elaboración del Prototipo de Interfaz de Usuario para realizar un prototipo que valide el proceso de levantamiento de requisitos.

### ➤ **Caso de Uso:**

Esta técnica se utiliza para describir las secuencias entre uno o más actores que interactúan en el sistema. Se basa en escenarios para la obtención de requerimientos donde se representa una respuesta como estímulo inicial de un actor, así como la descripción de los escenarios, cada uno de ellos comenzando por la interacción del actor con el sistema.

### ➤ **Arqueología de Documentos:**

Esta técnica es utilizada para analizar a profundidad toda la base documental (Leyes y manuales de procedimientos) dada por la organización para poder obtener información y que el trabajo sea más provechoso.

Como herramienta a utilizar en el Levantamiento de Requisitos:

### ➤ **Listas de Chequeo:**

Esta herramienta es utilizada a su vez como una técnica en las actividades de revisión en la etapa del Levantamiento de Requisitos. Son pruebas realizadas por el Equipo de Desarrollo con el objetivo de realizar una revisión a cada uno de los documentos que deben ser entregados al cliente, donde los mismos deben cumplir con los indicadores a evaluar.

## **2.8. Resolución de problemas y acciones correctivas para el Levantamiento de Requisitos.**

En la realización de un software, se está presente en la detección de un error (es) o problema (s) a la hora de su elaboración, como son los que se producen por la falta de entendimiento entre el cliente y el equipo de desarrollo.

Para la corrección de los problemas que se puedan presentar en la etapa del flujo de trabajo de Requerimientos, se conformará un documento titulado Informe de No Conformidades (NC), y conjuntamente con este, el documento de Listas de Chequeo donde quedará registrada la NC. En el desarrollo de esta revisión para obtener una mayor satisfacción será realizada en paralelo al cliente desde el primer momento en que se comienza el Levantamiento de Requisitos. Este documento pasará al responsable de la actividad de revisión y al Líder de Software.

### 2.9. Análisis de Riesgo.

Es necesario realizar un análisis de los riesgos en la etapa de definición de los Requisitos de Software, centrándose en las áreas del proceso de desarrollo, donde se revisan los requerimientos y se realiza la corrección de los mismos. Se tiene riesgos de varios tipos:

- Riesgos relacionados con la entrega de la documentación por parte del equipo de desarrollo al cliente.
- Riesgos relacionados con cronograma de actividades y tareas del equipo de desarrollo que se encuentra realizando el Levantamiento de Requisitos.
- Recursos humanos involucrados.
- Documentación generada por el Equipo de Desarrollo.

#### **Estrategia para la gestión de los Riesgos:**

**Evitar el riesgo:** Realizar la revisión de la entrega de la documentación al cliente en tiempo y forma, de manera tal que no sea afectado por riesgo alguno, así como la revisión de conjunto con el cliente para lograr un mayor entendimiento entre ambas partes.

**Aceptar el riesgo:** Decidir vivir con el riesgo. Monitorear los síntomas del riesgo y elaborar un plan de contingencia, en caso de que se materialice.

Si se decide aceptar el riesgo, es necesario mitigarlo, es decir tomar alguna acción inmediata que reduzca su impacto.

La estrategia que se asume es: **Aceptar el riesgo**, por lo tanto, luego de identificar los riesgos que asume el proyecto, se propone realizar un análisis de los mismos y el impacto que cada uno de ellos acarrearía para la realización exitosa de las tareas y actividades, y a partir de aquí conformar un plan para mitigarlos.

**Ponderación de los riesgos:** Luego se realizaría una ponderación de los mismos ordenándolos por orden descendente de impacto, es decir, de mayor a menor gravedad.

**Continuidad del análisis de riesgo:** En lo adelante se debe trabajar en función de crear un plan de trabajo o cronograma que ayude a gestionar los riesgos determinados en la etapa de Levantamiento de Requisitos, colocando hitos para el análisis y replanteamiento del plan de mitigación de manera continua.

### 2.10. Plan de Revisiones de la Propuesta.

El Plan de Revisiones es un documento encargado de describir de forma específica cómo el Equipo de Desarrollo realizará las revisiones en la etapa del Levantamiento de Requisitos. Su planificación, flujos de trabajo, herramientas y funcionamiento de los roles, deben quedar muy bien definidos. El

plan es elaborado a partir de lo propuesto por RUP, sólo que ha sido adaptado a las condiciones específicas de los proyectos de forma general. Por lo tanto es adaptable a cualquier proyecto que utilice la metodología de desarrollo RUP.

Hace referencia al documento de Listas de Chequeo donde se definen dos roles importantes, el Analista Principal y el Asegurador de los Requisitos de Software, cada cual con su descripción y el conjunto de conocimientos mínimos para realizar la tarea.

El Asegurador de los Requisitos de Software es el responsable de planificar las pruebas y revisiones que se realizarán en dicha etapa, así como orientar los artefactos a revisar, la entrega de la documentación y las listas de chequeo que se utilizarán para ello. En paralelo a esta actividad también se estarán controlando los cambios en que ocurran en los Requisitos de Software.

Luego conjuntamente con el Analista Principal se encargarán de aplicar las listas de chequeo registrando en ellas los resultados de cada elemento probado.

El Asegurador de los Requisitos de Software conforma el Informe de No Conformidades con todos los problemas que han sido detectados y lo hace llegar al responsable de sus soluciones.

Como última actividad el Asegurador de los Requisitos de Software realiza un resumen de los resultados y se archivan para su posterior uso, quedando así concluido el plan de revisiones (Ver Anexo 1).

### **2.11. Pruebas y Evaluaciones que se realizan en la Propuesta.**

Las pruebas que se realizan quedan registradas en un documento llamado Listas de Chequeo que a su vez servirá como un Plan de Prueba. En ese documento se describe de forma específica cómo el Equipo de Desarrollo realizará todas sus pruebas, así como una serie de preguntas que se le aplican a la documentación para obtener sus resultados y sus errores detectados si los presentan, como también si a los documentos le falta algún elemento por definir.

Es elaborado a partir de lo que propone RUP, sólo que ha sido adaptado a las condiciones específicas de forma general según las necesidades del equipo de desarrollo que vaya a efectuar las pruebas. Por tanto, su uso es muy relativo a la persona que lo realice. Para las pruebas se definen tres roles fundamentales: el Líder de Software, el Analista Principal y el Asegurador de Requisitos de Software. Se refleja su descripción y el conjunto de conocimientos mínimos para su buen desempeño. Además, se determina el flujo de trabajo que regirá la realización de dicha actividad.

### 2.12. Descripción de las Etapas de la Propuesta.

El proceso de Levantamiento de Requisitos (Requerimientos) está compuesto por 4 etapas fundamentales:

- Elicitación.
- Análisis.
- Especificación.
- Validación.

A continuación serán descritas las etapas de manera más detallada teniendo en cuenta sus objetivos, las actividades que se realizarán y los artefactos que se exigirán de entrada a cada actividad y los que se producirán para lograr las metas deseadas.

#### 2.12.1. Elicitación de los Requisitos de Software.

##### **Descripción:**

En esta etapa es donde se trata de descubrir los requisitos funcionales de software. Como primer paso se tiene el trabajo del equipo de desarrollo con los clientes y usuarios para obtener información de la organización e identificar el problema que se desea resolver. Una vez recopilada la información se realiza una comprensión del negocio que va a hacer automatizado y de las personas que interactuarán en el mismo para luego realizar una modelación de los procesos del negocio. Luego de tener toda la información se realiza una revisión técnica formal de todos los documentos generados hasta el momento. Cada actividad que se desarrolle estará registrada en un cronograma que puede variar según los defectos encontrados o problemas de otra índole. Posteriormente se hace la revisión y validación con el cliente de los procesos del negocio para poder pasar a la identificación de los requisitos funcionales de software. En esta etapa tendrán un papel importante el Líder de Software, Analista Principal, Analistas de Sistema, Asegurador de los Requisitos de Software, Arquitecto Principal, Especialista Funcional junto con el Cliente para descubrir lo que el sistema debe presentar. El Asegurador de los Requisitos de Software se apoyará en el estándar **ITIL** para la captura de información, y de este modo obtener una mayor calidad a la hora de definir lo que realmente se desea desarrollar, ya que dará una guía de cómo quedará organizada la actividad. Este grupo de personas deben hacer una extracción de la información a desarrollar para lograr el propósito del Cliente y satisfacer sus necesidades.

##### **Objetivo:**

Lograr una correcta elicitación de los requisitos, que satisfaga las necesidades del cliente y que cumpla con la calidad requerida.

### **Actividades a realizar:**

- Obtener información de la organización y del problema a resolver: Consiste en recopilar información de la organización e identificar el problema a resolver.
- Comprensión del Negocio: Se hace un estudio de la información recopilada para comprender cómo funciona el negocio de la organización que va a ser automatizada, así como las personas que interactúan con el mismo.
- Realizar la modelación de los procesos del negocio: Se realiza una modelación de los procesos del negocio, a partir de la información identificada en la actividad anterior.
- Revisión Técnica Formal: Se realiza una revisión para asegurar la calidad de los documentos realizados.
- Actualizar Cronograma: Se actualizan las tareas planificadas en el cronograma.
- Revisión y validación de los procesos del negocio por el cliente: Revisar y validar la modelación de los procesos del negocio con el cliente.
- Identificar Requisitos Funcionales: Se identifican los requisitos funcionales a partir de los requisitos de la información identificada y la modelación de los procesos del negocio.

Para ver los roles, técnicas, tareas, herramientas y estándares que se utilizan en cada una de las actividades propuestas (Ver Anexo 15).

### **Artefactos que se generan:**

#### **De Entrada:**

- Documento Visión.
- Cronograma.
- Guía de Preguntas.

#### **De Salida:**

- Minuta de entrega.
- Base de documentación.
- Glosario de Términos.
- Guía de Preguntas (Actualizada).
- Cronograma (Actualizado).
- Documento de los Procesos del Negocio.
- Documento de Revisión Técnica Formal.
- Listado de Requisitos Funcionales.

- Lista de Riesgos.
- Listas de Chequeo.
- Informe de No Conformidades.
- Documento de los Procesos del Negocio (Validado).
- Glosario de Términos (Validado).
- Minuta de aceptación.

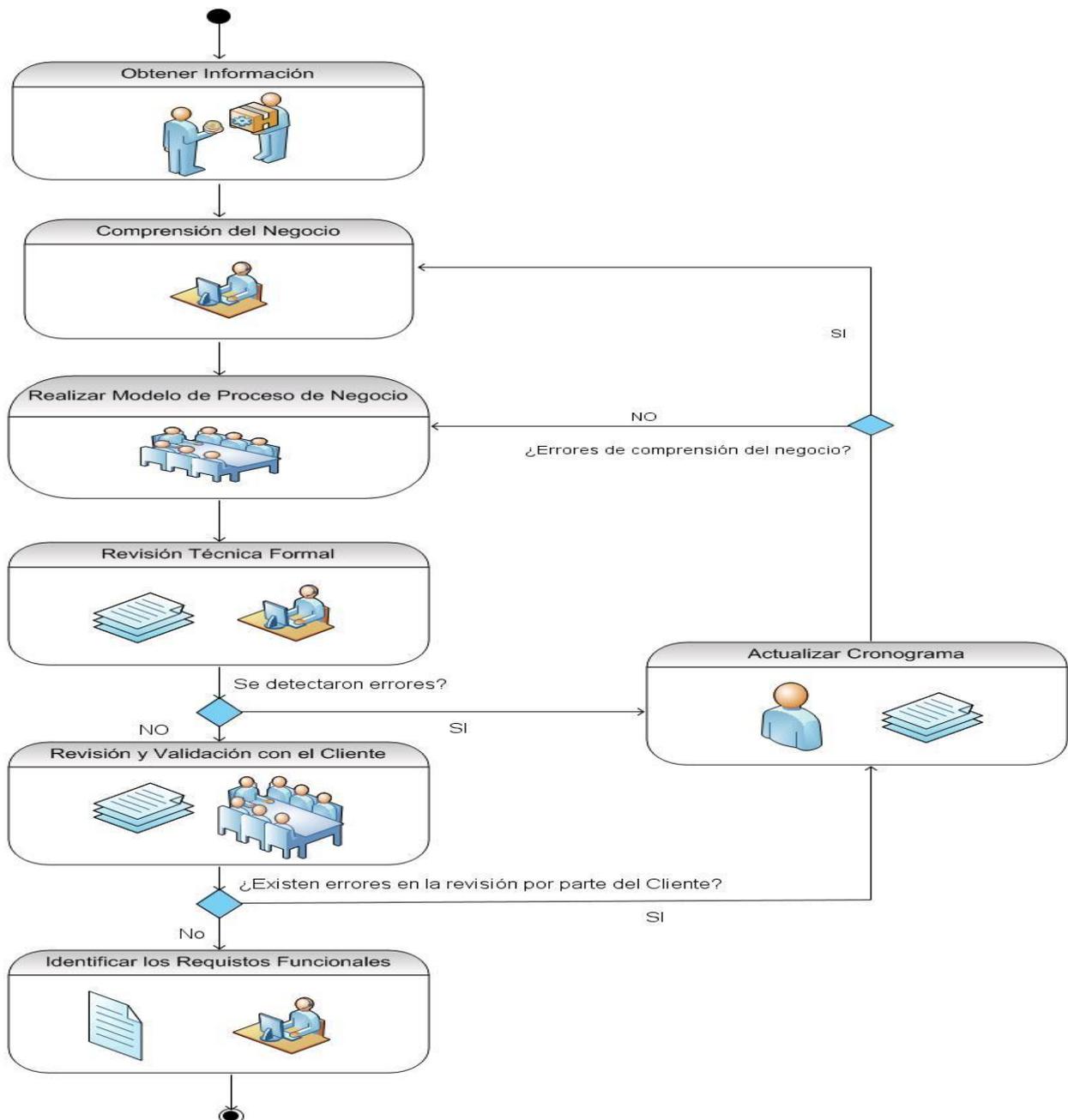


Figura 9: Imagen de las Actividades de la etapa de Elicitación

### 2.12.2. Fase de Análisis de los Requisitos de Software.

#### **Descripción:**

Una vez terminada la fase de Elicitación de Requisitos de software, esta comienza a partir de la base de información extraída en la elicitación previo de cada posible requisito a documentar. Dicha fase se realiza en conjunto con el Cliente, se van determinando una serie de Requisitos que el sistema debe cumplir, a su vez puede también aparecer un nuevo requisito. Primeramente se hace una caracterización de cada requisito identificado, logrando de ellos que no sean ambiguos, que sean concisos, completos y verificables. Luego de haber caracterizado los requisitos se les asignan una prioridad de Alto, Medio o Bajo, esto consiste en cómo será priorizado el requisito durante su desarrollo. El grupo de Analistas juegan un papel importante en esta etapa ya que son los encargados de entender básicamente lo que el Cliente desea desarrollar y son los que conformarán una serie de ideas a tratar y un debate de lo posible a documentar. Luego de terminado el análisis de lo investigado y discutido tanto el equipo de desarrollo como el equipo por parte del cliente, se podrá pasar a la fase de Especificación de Requisitos.

#### **Objetivo:**

Lograr un análisis y negociación de los requisitos para identificar posibles conflictos y caracterizar y priorizar cada requisitos, y llegar a un completo desarrollo para su especificación.

#### **Actividades a desarrollar:**

- Análisis y negociación de los requisitos: Analizar los requisitos identificados en la etapa de elicitación y negociar con el cliente nuevos requisitos.
- Revisión Técnica Formal: Realizar una revisión interna del documento de Listado de Requisitos Funcionales para asegurar la calidad del mismo.
- Actualizar Cronograma: Se actualizan las tareas planificadas en el cronograma.
- Priorizar los Requisitos: Se priorizan los Requisitos identificados de acuerdo a las necesidades de los usuarios ya sea en Alto, Medio o Bajo. Se realiza una negociación con el cliente si existe algún Requisito nuevo y se vuelve a definir.

Para ver los roles, técnicas, tareas, herramientas y estándares que se utilizan en cada una de las actividades propuestas (Ver Anexo 16).

#### **Artefactos que se generan:**

##### **De Entrada:**

- Glosario de Términos.
- Cronograma.
- Documento de los Procesos del Negocio.
- Listado de Requisitos Funcionales.

- Lista de Riesgos.

**De Salida:**

- Listado de Requisitos Funcionales (Actualizado/Modificado).
- Informe de No Conformidades.
- Registro de Revisiones.
- Cronograma (Actualizado).

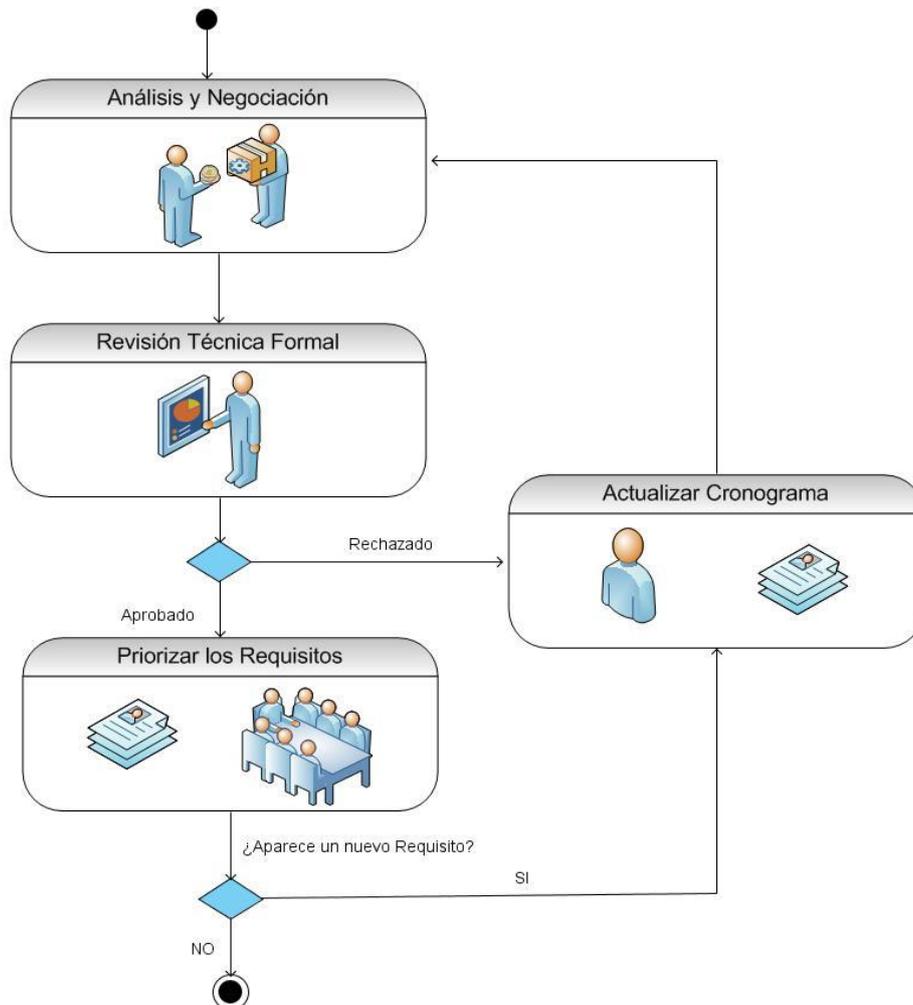


Figura 10: Imagen de las Actividades de la etapa de Análisis.

### 2.12.3. Fase de Especificación de los Requisitos de Software.

**Descripción:**

Esta fase es donde se documentan los Requisitos de Software creándose el Documento de Especificación de Requisitos de Software. Se definen todos los requerimientos funcionales, diagramas, y cualquier otra información que sirva de soporte y guía para fases posteriores. Es importante destacar que la especificación de requisitos es el resultado final de las actividades de elicitación y análisis de requerimientos; este documento será utilizado como fuente básica de comunicación entre los clientes, usuarios finales, analistas de sistema, personal de pruebas, y todo

aquel involucrado en la implementación del sistema. Esta etapa en cierto punto se va realizando en paralelo con la fase de Análisis, pero es aquí donde se pasa en limpio toda la información capturada, aplicando técnicas y estándares para desarrollar la documentación y que la misma se logre con la calidad requerida. En la Especificación de Requisitos juegan un papel importante el Analista Principal, Analista de Sistema, Asegurador de los Requisitos de Software, Diseñador Gráfico, Especialista Funcional, Cliente y el Líder de Software. El Analista Principal junto con los Analistas de Sistema se encargará de realizar la documentación conjuntamente con el Cliente para ir analizando todo lo que realmente desea. A su vez el Asegurador de los Requisitos de Software estará realizando las revisiones de la documentación aplicando listas de chequeo para la Especificación de los Requisitos. También es necesaria la presencia del Especialista Funcional porque tendrá un alto grado de conocimiento del negocio y será el intermediario entre el Equipo de Desarrollo y el Cliente. Luego de haber documentado la Especificación de Requisitos, se documenta el Modelo de Casos de Uso del Sistema y los Prototipos no Funcionales, se realizará el mismo procedimiento, donde es de vital importancia la constante revisión conjuntamente con el Cliente de los documentos elaborados, para evitar problemas a la hora de desarrollar el sistema.

### **Objetivos:**

Lograr la Especificación de cada uno de los Requisitos capturados, el Modelo de casos de uso del sistema, así como también la obtención de los Prototipos no Funcionales.

### **Actividades a desarrollar:**

- Documento de Especificación de los Requisitos de Software: Se realiza la documentación en limpio de todos los Requisitos que desea el Cliente que tenga su sistema a desarrollar.
- Revisión Formal Interna: Se realiza una revisión interna al Documento de Especificación de Requisitos para asegurar la calidad del mismo.
- Revisión con el Cliente: Realizar una revisión con el Cliente de la documentación de Especificación de los Requisitos de Software realizada por los Analista del Sistema del Equipo de Desarrollo.
- Actualizar Cronograma: Se actualizan las tareas planificadas en el cronograma de la Especificación de los Requisitos de Software.
- Modelo de Caso de Uso de Sistema: Luego de terminada la especificación se seleccionan los casos de usos a implementar y se documentan.
- Revisión Formal Interna: Se realiza una revisión interna al Modelo de Caso de Uso de Sistema para asegurar la calidad del mismo.
- Revisión con el Cliente: Realizar una revisión con el Cliente del documento de Modelación de los Casos de Uso del Sistema realizado por los Analista del Sistema del Equipo de Desarrollo.

- Actualizar Cronograma: Se actualizan las tareas planificadas en el cronograma de la Modelación de los Casos de Uso del Sistema.

Para ver los roles, técnicas, tareas, herramientas y estándares que se utilizan en cada una de las actividades propuestas (Ver Anexo 17).

### **Artefactos que se generan:**

#### **De Entrada:**

- Listado de Requisitos Funcionales.
- Glosario de Términos.
- Cronograma.

#### **De Salida:**

- Documento de Especificación de Requisitos de Software.
- Documento de Especificación de Requisitos de Software (Aprobado).
- Glosario de Términos (Actualizado/Aprobado).
- Cronograma (Actualizado).
- Modelo de Caso de Usos del Sistema.
- Modelo de Caso de Usos del Sistema (Aprobado).
- Registro de Revisiones.
- Informe de No Conformidades.
- Minuta de aceptación.

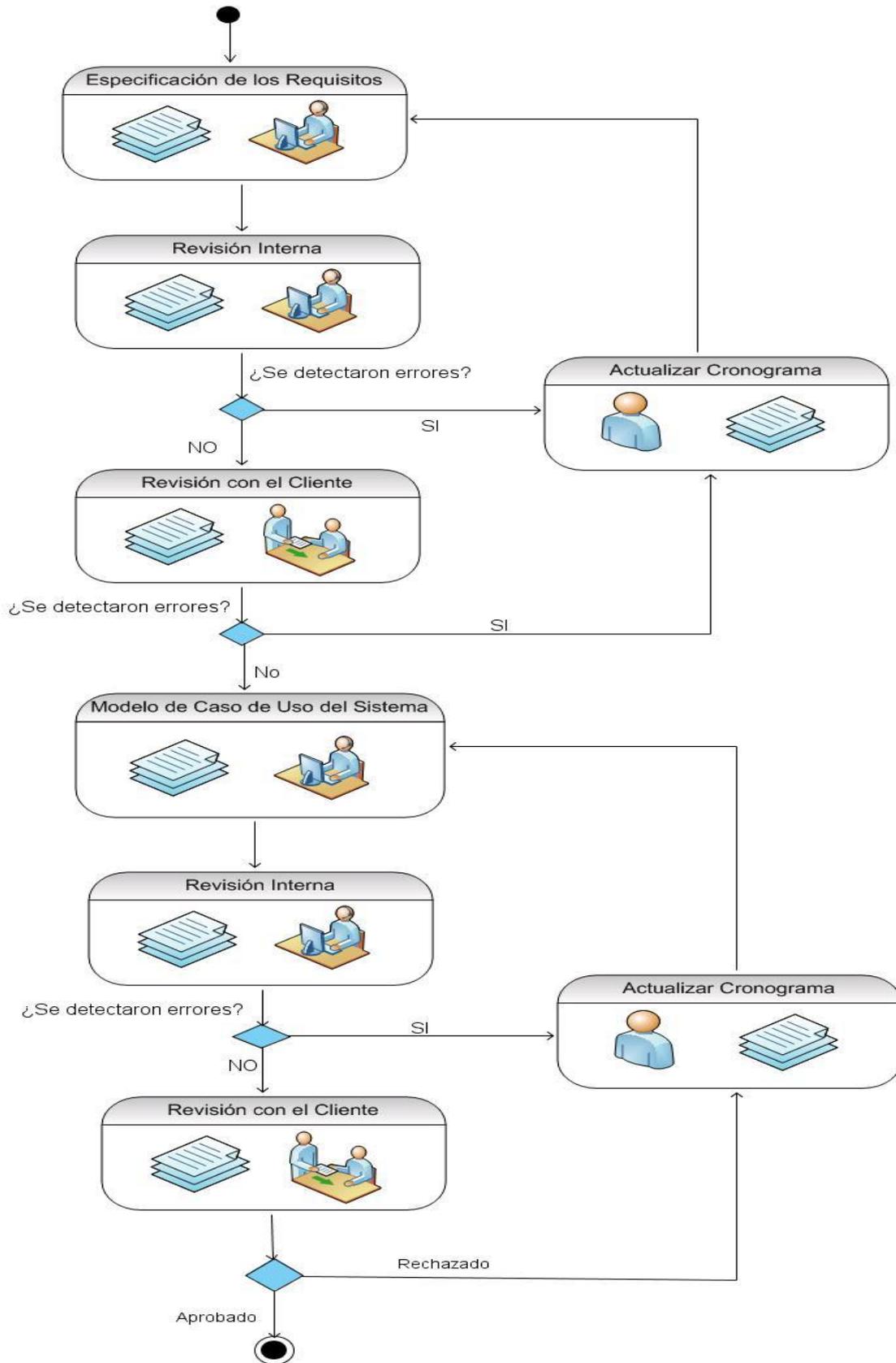


Figura 11: Imagen de las Actividades de la etapa de Especificación.

### 2.12.4. Fase de Validación de los Requisitos de Software.

#### **Descripción:**

La Validación es la actividad del Levantamiento de Requisitos que permite demostrar que los Requerimientos definidos en el sistema son los que realmente quiere el cliente; además revisa que no se haya omitido ninguno, que no sean ambiguos, inconsistentes o redundantes. En este punto es necesario recordar que el documento de especificación de requisitos debe estar libre de errores, por lo tanto, la validación garantiza que todos los requerimientos presentes sigan los estándares de calidad establecidos. En esta fase se reúne todo el Equipo de Desarrollo con el Cliente y se muestra todo lo documentado para llegar a una Validación de lo que posteriormente se va implementar. Si el Cliente no está de acuerdo con lo que está redactado en la documentación, ya sea porque no hubo un entendimiento entre el Equipo de Desarrollo y el Cliente. Terminada esta fase, se firman los documentos que serán entregados al Cliente.

#### **Objetivos:**

Lograr la Validación de la Especificación de Requisitos de Software y el Modelo de Caso de Uso del Sistema.

#### **Actividades a desarrollar:**

- Validación de la Especificación de los Requisitos de Software: Se realiza una validación en conjunto con el Cliente de la Especificación de los Requisitos de Software.
- Validación del Modelo de Caso de Uso del Sistema: Se realizar una validación en conjunto con el Cliente del Modelo de Caso de Uso del Sistema.

Para ver los roles, técnicas, tareas, herramientas y estándares que se utilizan en cada una de las actividades propuestas (Ver Anexo 18).

#### **Artefactos que se generan:**

##### **De Entrada:**

- Documentos de Especificación de Requisitos de Software.
- Modelo de Caso de Usos del Sistema.
- Glosario de Términos.

##### **De Salida:**

- Modelo de Caso de Uso del Sistema (Aprobado).
- Documento de Especificación de los Requisitos de Software (Aprobado).
- Registro de Revisiones.
- Informe de No Conformidades.

- Glosario de términos (Aprobado).
- Minuta de Aceptación.

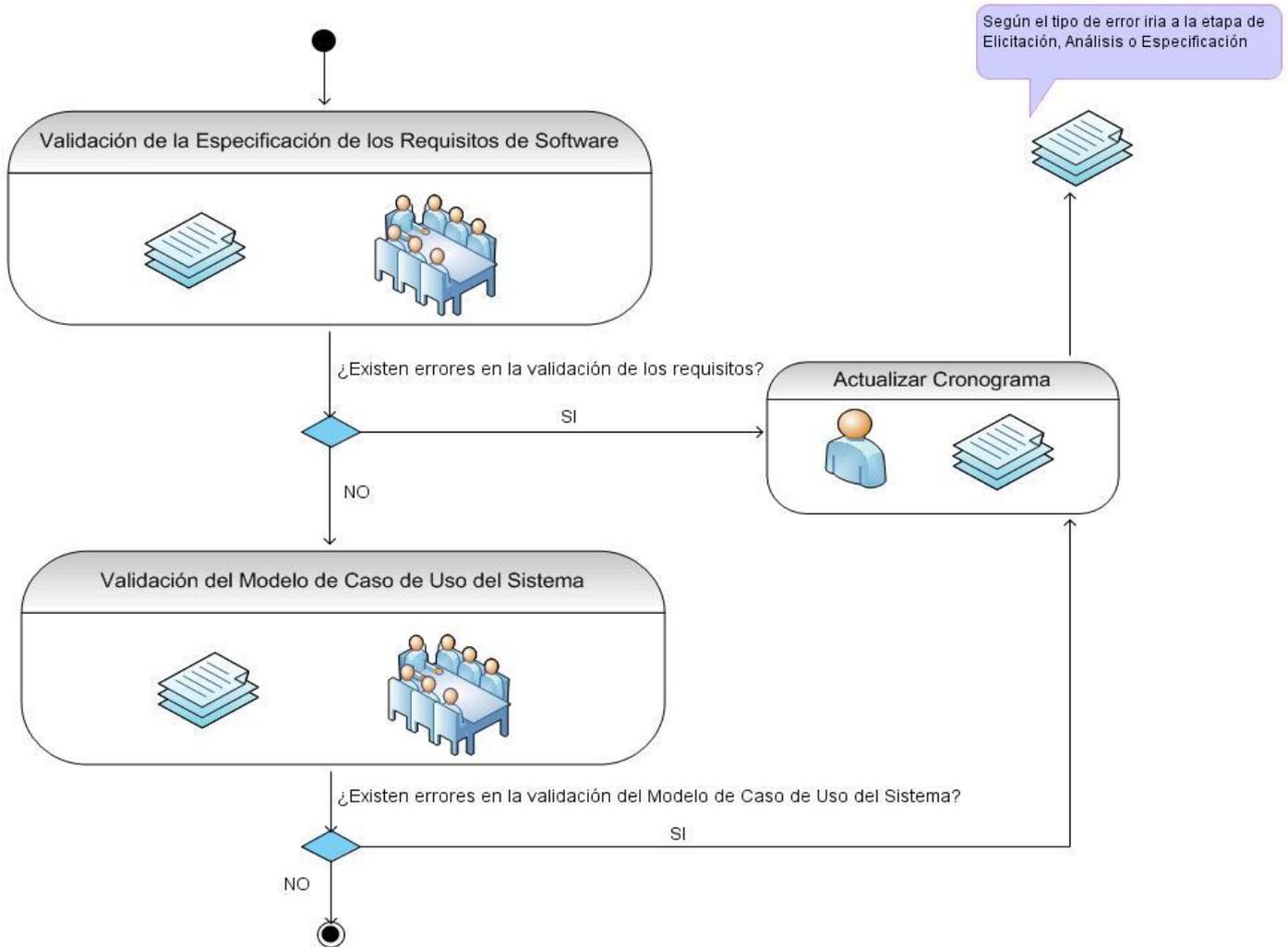


Figura 12: Imagen de las Actividades de la etapa de Validación.

### 2.13. Conclusiones

Como resultado de la propuesta que se realizó en base a las etapas de la Ingeniería de Requisitos ya sea Elicitación, Análisis, Especificación y Validación que se debe llevar a cabo en el Levantamiento de Requisitos se obtuvo un conjunto de resultados que pueden ser aplicables a cualquier proyecto que use la Metodología de Desarrollo RUP.

La Propuesta que se realizó estuvo basada en determinar el objetivo y la descripción del proceso que se propone para el Levantamiento de Requisitos en función del Aseguramiento de la Calidad de Software, donde se identificaron las etapas que componen dicho proceso. De cada una de las etapas se definieron las tareas, técnicas, los roles involucrados y los artefactos que se generan durante cada una de estas actividades.

La Propuesta está basada en toda una investigación que se realizó en el Capítulo 1 donde se hace referencia a las buenas prácticas que deben ser utilizadas en la etapa de Levantamiento de Requisitos que tributen a la calidad del producto final, ya sea las herramientas y las técnicas para verificar que los requisitos cumplan con las necesidades del Cliente, además de un estudio de la Ingeniería de Requisitos y un análisis de la Calidad de Software.

Luego de haber culminado un análisis de todo el proceso se ha podido llegar a la conclusión de que todos los roles definidos en la propuesta son importantes durante el Levantamiento de Requisitos, aunque no todos participen durante todo el proceso, sino en algunas actividades del flujo de Requerimientos que es necesaria su participación para obtener un mayor resultado. Esto no significa que tenga que existir una persona para cada rol, sino que hay personas que pueden cumplir más de un rol de acuerdo al grado de responsabilidad en cada una de las actividades y los conocimientos.

En el capítulo también se definieron un conjunto de artefactos que deben ser utilizados como entradas a una actividad y otros que son salidas que se generan con un resultado valioso para ambas partes involucradas en el proceso o solamente para el Equipo de desarrollo. Para la mayoría de estos artefactos se definieron un grupo de plantillas que tienen como objetivo organizar y formalizar dicho proceso. También se realizaron un conjunto de artefactos que son para probar los Requisitos y asegurar la calidad de los mismos a la hora de hacer entrega al Cliente.

Se hace uso de un conjunto de Estándares y Modelo como guía para obtener una mayor calidad en el proceso de Levantamiento de Requisitos, así como la aplicación de Herramientas y Técnicas para la captura de información.

La elaboración de esta propuesta, en su conjunto, tiene los elementos que se consideran importantes para lograr que el proceso de Levantamiento de Requisitos Funcionales en función del Aseguramiento de la Calidad de Software cumpla con las necesidades del Cliente.

---

### CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Se realiza la revisión, valoración y evaluación de la propuesta para el levantamiento de requisitos en función del aseguramiento de la calidad de software, con el objetivo de validar la solución, y de conocer el criterio de especialistas en el tema que de alguna forma u otra han participado y adquirido conocimientos sobre la etapa de levantamiento de requisitos y el aseguramiento de la calidad de software. Se aplicó un cuestionario que permitió conocer dichos criterios y analizar los resultados de forma cuantitativa, generando estadísticas que pueden ilustrar de forma general la aceptación, adaptabilidad, importancia, necesidad, completitud, entre otros parámetros de evaluación definidos y evaluados por los especialistas.

#### 3.1. Tipos de Evaluación

Existen varios tipos de evaluaciones que fueron analizados para escoger los factibles y aplicarlos a la propuesta.

Criterio de expertos. Delphy:

- El Método de Expertos Delphy desde sus inicios en los años 50 ha sido utilizado frecuentemente como sistema para obtener información sobre las ocurrencias de un fenómeno en el futuro. Consiste en la selección de un grupo de expertos a los que se les encuesta su opinión sobre cuestiones referidas a sucesos del futuro. El método se basa en la utilización sistemática de un juicio intuitivo emitido por un grupo de expertos obtenido encuestando a este grupo mediante un cuestionario. Es un método fiable y muy utilizado actualmente pero necesita del estudio de competencias en los participantes del panel.
- No existe posibilidad de aplicarlo a la propuesta debido a la poca existencia de personal con experiencia en el tema en la UCI. Existían posibilidades de contactarlos en otras universidades o posibilidad de contactarlos en Internet, pero no se creyó fiable pues no se estaría respetando el ajuste a las necesidades de la universidad pues estos no conocerían el modelo de producción de la misma. Siendo este uno de los principales objetivos que persigue la propuesta, o sea la estrategia de alineación al modo de levantamiento de requisitos de la UCI.

**Criterio de especialistas:**

- El criterio de especialistas es un instrumento rápido y eficaz por el potencial que contiene para conformar, valorar y enriquecer criterios, concepciones, modelos, estrategias y metodologías. Existen varias técnicas:
  - ✓ Encuestas.

- ✓ Cuestionarios.
- ✓ Entrevistas.
- ✓ Estados de opinión.
- ✓ Positivo-Negativo–Interesante.
- ✓ Sugerencias.

### Validación Práctica:

- No es más que la obtención, comparación y análisis de resultados obtenidos al aplicar prácticamente el procedimiento en varios proyectos.
- No es posible aplicarlo pues no existía(n) en estos momentos ningún proyecto dispuesto a realizar el levantamiento de requisitos que se pudiera(n) utilizar para aplicarle la propuesta.

### Recopilación de información:

- Se basa en recoger estados de opinión, encuestas, cuestionarios o entrevistas a los clientes o a las personas que tengan que ver de una forma u otra con la propuesta, o con la puesta en práctica de esta de forma general.
- Tampoco se puede aplicar porque depende de la realización práctica de la propuesta.

### Triangulación:

- La triangulación entendida como técnica de confrontación y herramienta de comparación de diferentes tipos de análisis de datos (triangulación analítica) con un mismo objetivo puede contribuir a validar un estudio de encuesta y potenciar las conclusiones que de él se derivan. Se hace a través de datos estadísticos recogidos sobre resultados de la puesta en práctica. O de los estados de opinión.
- Es imposible de aplicar pues los datos deben ser recogidos aplicando este. O recogiendo estados de opinión de los involucrados la propuesta, cosa que tampoco se hace posible por el momento, dado a lo explicado previamente.

### Grupo Focal:

- Básicamente es la selección de un grupo de personas con conocimientos sobre el tema, deben ser especialistas, expertos, de distintos niveles y categorías, que se reúnen en un lugar a una hora determinada, donde se discute en forma de grupo debate sobre la propuesta, siendo este debate dirigido por los autores, y centrado en lo que se quiere conocer sobre la misma.

- Imposibilidad de realizar esta actividad por el desconocimiento sobre el tema.

Se decidió realizar por el criterio de especialistas pues es el que más se ajusta a las necesidades y de esta propuesta, según lo analizado previamente. Este fue el seleccionado para la validación de la propuesta por las ventajas y oportunidades que ofrecía.

### **3.2. Criterio de Evaluación**

Para realizar la validación de la propuesta se tuvo en cuenta primeramente los criterios que serían evaluados por los especialistas. Estos son:

Importancia y necesidad de la aplicación de la propuesta.

Adaptabilidad a los proyectos que utilicen una metodología diferente a RUP.

Posibilidades de aplicación.

Eficacia de la propuesta.

Nivel de complejidad de la propuesta.

Grado de completitud de la propuesta.

Orden lógico y estructura de las actividades.

Satisfacción de las necesidades de los proyectos.

### **3.3. Diseño del cuestionario**

Se diseñó un cuestionario que tuviera en cuenta los aspectos anteriormente mencionados y que sirviera de guía para la evaluación por parte de los especialistas (Ver anexo 19).

### **3.4. Selección de los especialistas**

Para seleccionar los especialistas que participarían en la validación se tuvo en cuenta los siguientes requisitos:

1. Poseer grado científico de ingeniero o superior.
2. Tener conocimiento y experiencia en el tema.
3. Haber participado como mínimo en una etapa de levantamiento de requisitos de un proyecto con resultados satisfactorios.
4. Ser jefe de proyecto, gerente, analista principal, analista de sistema, jefe de calidad u otro rol importante y con influencia directa dentro del levantamiento de requisitos o del equipo de este.
5. Pertenecer a la UCI y conocer el modelo de producción de la universidad, para valorar correctamente el ajuste a las necesidades de esta, que contiene la propuesta.

### 3.5. Especialistas seleccionados

De los especialistas seleccionados se hacía necesario conocer a fondo sus características para que cumplieran con lo establecido para participar en la validación, por lo que se les aplicó un cuestionario (Ver Anexo 19).

Este recoge una serie de datos personales y de conocimientos adquiridos que nos brindó la información que se cita a continuación.

#### **Especialista # 1**

*Nombre:* Ing. Roig Calzadilla Díaz.

*Cargo:* Especialista de la Dirección de Calidad de Software.

*Categoría Científica:* Ingeniero en Ciencias Informáticas.

*Vinculación a proyecto:* Laboratorio de Pruebas.

*Roles:* Especialista de Calidad.

*Experiencia:* Cuenta con cuatro años de labor en la producción de software (dos como estudiante, y dos como profesional). Ha adquirido conocimientos y experiencias en el tema de calidad de software. Presenta dos años de experiencia en la dirección de proyectos con resultados exitosos.

#### **Especialista # 2**

*Nombre y Apellidos:* Ing. Geiser Arcio Pérez Rivas.

*Cargo:* Especialista de la Dirección de Calidad.

*Categoría Científica:* Ingeniero en Ciencias Informáticas.

*Vinculación a proyecto:* Proyectos de Desarrollo del Grupo Central de Calidad.

*Roles:* Ha desempeñado varios roles como:

- Jefe de Proyecto.
- Programador.
- Jefe del Grupo de Aseguramiento.
- Analista.

*Experiencia:* Cuenta con cuatro años de experiencia en el desarrollo de proyectos de software exitosos y un año en la dirección de proyectos de software. Ha adquirido experiencia y conocimiento en el tema de calidad de software y como analista de sistema por su labor desempeñada a lo largo de su carrera y vinculación laboral.

### **Especialista # 3**

*Nombre y Apellidos:* Ing. Yudisbel Pérez Moreno.

*Cargo:* Especialista de la Dirección de Calidad.

*Categoría Científica:* Ingeniero en Ciencias Informáticas.

*Vinculación a proyecto:* Ha participado en varios proyectos como:

- Proyectos de Desarrollo del Grupo Central de Calidad.
- Cuerpo de Investigaciones Científicas, Penales y Criminalísticas (CICPC).

*Roles:* Gestión de la Configuración.

*Experiencia:* Cuenta con cuatro años de experiencia en el desarrollo de proyectos de software exitosos (dos como estudiante y dos como profesional) y dos años en la dirección de proyectos de software. Ha adquirido experiencia y conocimiento en el tema de calidad de software la rama de gestor de la configuración de software a lo largo de su trabajo.

### **Especialista # 4**

*Nombre y Apellidos:* Ing. Danaysa Macías Hernández.

*Cargo:* Profesora de Ingeniería de Software.

*Categoría Científica:* Ingeniero en Ciencias Informáticas.

*Vinculación a proyecto:* Ha participado en varios proyectos como:

- Oficina Nacional de Estadísticas (ONE).
- Convenio Integral de Cooperación Cuba – Venezuela (CCV).

*Roles:* Ha desempeñado varios roles como:

- Analista.
- Jefe de módulo.

*Experiencia:* Cuenta con cuatro años de experiencia en el desarrollo de proyectos de software exitosos (dos como estudiante y dos como profesional) y un año en la dirección de proyectos de software. Ha adquirido experiencia y conocimiento en el tema de analista de software y como jefa de modulo.

### **Especialista # 5**

*Nombre y Apellidos:* Ing. Diana Valdés González.

*Cargo:* Profesora de Ingeniería de Software.

*Categoría Científica:* Ingeniero en Ciencias Informáticas.

*Vinculación a proyecto:* Servicio Autónomo de Registros y Notarías.

*Roles:* Ha desempeñado varios roles como:

- Analista Principal.
- Analista de Sistema.

*Experiencia:* Cuenta con tres años de experiencia en el desarrollo de proyectos de software exitosos (uno como estudiante y dos como profesional) y un año en la dirección de proyectos de software. Ha adquirido experiencia y conocimiento en el tema de analista de software.

### 3.6. Entrega de los cuestionarios y de la propuesta

Una vez realizado el estudio previo sobre los tipos de evaluación, se escogió el que se usaría y se seleccionaron los criterios a evaluar. Se diseñó el cuestionario y una vez seleccionados los especialistas, y analizadas sus características se les entregó la propuesta, y el cuestionario de evaluación para que procedieran con la validación.

### 3.7. Resultados de los cuestionarios

Una vez que los especialistas analizaron a fondo la propuesta, evaluaron la misma emitiendo los siguientes criterios:

**Especialista # 1:** Considera lo siguiente de la propuesta.

Una propuesta que puede ayudar a mejorar la etapa de levantamiento de requisitos para lograr un producto final sin lamentar errores en esta etapa. Cumple con todas las actividades necesarias para el levantamiento de requisitos. Todas las actividades están dispuestas correctamente, es decir orden lógico y estructural satisfactorio. La propuesta no es compleja, es bastante abarcadora, lo que implica que puede hacerse más ligera acorde con el levantamiento de requisitos y el aseguramiento de la calidad en toda la etapa. Básicamente puede poseer dualidad de roles y reducción de artefactos en el caso de soluciones sencillas. En general es un excelente trabajo, con alto nivel técnico y profundidad tanto en definición de artefactos, roles y actividades.

**Especialista # 2:** Considera lo siguiente de la propuesta.

Interesante y que puede tener grandes resultados, todo en dependencia del grado de aceptación de los proyectos y su utilización. Es importante y necesaria para desarrollar la etapa de Levantamiento de Requisitos en la UCI. Esta cumple con las actividades a desarrollar dentro de un Levantamiento de Requisitos y están ordenadas correctamente. Puede que no sea aplicable a todos los proyectos que utilicen otra metodología diferente a RUP, pues estos presentan disímiles características, de

tamaño y complejidad, y para ello habría que hacerle adaptaciones a la propuesta, aunque sí se puede usar como base para ello. En cuanto a la utilización de los estándares de calidad tiene unacorrecta utilización aunque es necesario destacar la utilización de la IEEE 830 para una mayor completitud. De manera general la propuesta no es compleja, es bastante simple y abarcadora.

**Especialista # 3:** Considera lo siguiente de la propuesta.

Creo que es de gran importancia y que da un paso más de avance en todo el proceso en el Levantamiento de Requisitos en función del Aseguramiento de la Calidad de Software. La propuesta desde mi punto de vista minimizará los problemas relacionados al desarrollo de software asegurando la calidad del mismo en todas las etapas del flujo de Requerimiento, veo acertada la metodología seleccionada (RUP) como metodología rectora en la actualidad en el mundo del SW y previendo el aumento de proyectos de gran envergadura en nuestra universidad. Sugiero abordar los flujos pendientes en el proceso Levantamiento de Requisitos y poner en práctica lo más pronto posible dicha investigación en los proyectos de nuestra universidad.

**Especialista # 4:** Considera lo siguiente de la propuesta.

La propuesta cumple con las necesidades y es de gran importancia para el levantamiento de requisitos. Desde mi punto de vista minimizará los problemas a lo largo de las etapas de elicitación, análisis, especificación y validación. Creo muy necesario las revisiones que se realizan en la propuesta, ya que es lo que va a garantizar la calidad del producto final. Se ha realizado una correcta selección de técnicas para el levantamiento de requisitos como también las herramientas a utilizar por parte de calidad. Al igual que otros especialistas visualiza la importancia y necesidad de esta propuesta, así como el orden lógico de las actividades, y su aplicabilidad. La propuesta de forma general se considera bastante completa, la selección de las fases y las actividades por cada una, son explicadas de manera sencilla y proveen una guía útil para la planificación, control y seguimiento de las revisiones durante las etapas. Esta propuesta abarca de manera general el “qué” hacer para la captura de información en el levantamiento de requisitos y la revisión de todas las actividades y artefactos generados, el “como” es un aspecto altamente dependiente de las particularidades de cada proyecto aunque pudieran existir aspectos generales comunes para el tipo de aplicación. Sin dudas la propuesta permitirá mejorar la planificación y revisión de las actividades en la etapa de levantamiento de requisitos. De forma general toda la propuesta para el Levantamiento de Requisitos en función del aseguramiento de la Calidad de Software es hasta cierto nivel compleja, en este caso particular se trazan pautas generales que son fácilmente aplicables y entendibles por los involucrados en la propuesta.

**Especialista # 5:** Considera lo siguiente de la propuesta.

Es una propuesta muy buena y completa, ya que propone realizar revisiones en cada una de las etapas de la IR, garantizando así que no se comience una etapa si en la anterior se detectan errores. Esto contribuye a la obtención de requisitos completos, correctos, sin ambigüedades y que satisfagan las necesidades de los clientes.

### 3.8. Análisis de los resultados

A continuación se analizarán los resultados vistos desde el punto de vista estadístico, para arribar a conclusiones sobre el nivel de aceptación de los especialistas seleccionados.

Criterio de evaluaciones	Valores
Alta	5 puntos
Media	4 puntos
Baja	3 puntos
Ninguna	2 puntos

Figura 13: Tabla de Criterio de Evaluaciones.

Aspectos a evaluar	Especialistas	1	2	3	4	5	Promedio
		Importancia y necesidad	5	5	5	5	
Adaptabilidad		4	5	3	5	5	4.4
Posibilidades de aplicación.		5	5	5	5	5	5.0
Eficacia de la propuesta.		5	5	5	5	5	5.0
Nivel de complejidad de la propuesta.		4	4	4	4	4	4.0
Grado de completitud de la propuesta.		4	5	5	5	5	4.6
Orden lógico y estructura de las actividades.		5	5	5	5	5	5.0
Satisfacción de las necesidades de los proyectos.		5	5	5	4	5	4.8
<b>Promedio</b>		4.63	4.88	4.5	4.75	4.75	<b>4.71</b>

Figura 14: Tabla de los aspectos a evaluar.

### 3.9. Análisis de la concordancia en la valoración de aspectos (coeficiente de Kendall).

Después de obtener los resultados de las evaluaciones en la consulta a los especialistas se necesita demostrar su confiabilidad, se debe probar el nivel de acuerdo entre ellos para otorgar mayor autenticidad al estudio, es preciso comprobar el grado de coincidencia de las valoraciones realizadas. Se puede utilizar entonces el Coeficiente de Concordancia de Kendall, que constituye un estadígrafo muy útil en estudios de confiabilidad entre especialistas y/o especialistas de una materia, al determinar la asociación entre distintas variables. Es una medida de coincidencia entre ordenaciones que pueden ser objetos o individuos. En este caso el coeficiente de concordancia (W) será un índice de la divergencia o del acuerdo efectivo entre los especialistas, mostrado en los datos del máximo acuerdo posible (perfecto) (Hurtado de Mendoza, y otros).

Para la aplicación del Coeficiente de Concordancia de Kendall (W), se utilizan los valores que se muestran en la figura 15. Para agilizar el cálculo de W se introdujeron los datos en el software SPSS que permite realizar análisis y gráficos estadísticos sin tener que conocer la mecánica de los cálculos ni la sintaxis de los comandos del sistema. Comparado con otros programas, es más intuitivo y fácil de aprender. Su desventaja es que es menos flexible y con menos procedimientos avanzados que otros programas comerciales.

SPSS es bueno a la hora de organizar y analizar datos. Se puede ordenar datos, calcular nuevos datos y realizar una gran variedad de análisis estadísticos. En teoría el tamaño de los ficheros de datos que SPSS puede manejar no está limitado por lo que puede trabajar con ficheros grandes. Esta versión también permite el manejo cómodo de ficheros, la personalización de los informes, y el cortar y pegar en otros programas (Kessler).

Luego de haber realizado el cálculo, la herramienta arrojó los siguientes resultados:

Parámetros	N	Media	Desviación típica	Mínimo	Máximo
Importancia y Necesidad	5	4.8000	.44721	4.00	5.00
Aplicación	5	5.0000	.00000	5.00	5.00
Eficacia	5	5.0000	.00000	5.00	5.00
Complejidad	5	4.0000	.00000	4.00	4.00
Complejitud	5	4.8000	.44721	4.00	5.00
Estructura	5	5.0000	.00000	5.00	5.00
Satisfacción de las Necesidades	5	4.8000	.44721	4.00	5.00

Figura 15: Tabla Estadística Descriptiva.

En la prueba estadística el Coeficiente de Concordancia de Kendall (W) ofrece el valor que posibilita decidir el nivel de concordancia entre los especialistas. El valor de W oscila entre 0 y 1. El valor de 1 significa una concordancia de acuerdos total y el valor de 0 una concordancia de desacuerdo total. Obviamente la tendencia a 1 es lo deseado, pudiéndose realizar nuevas rondas si en la primera no es alcanzada la significación en la concordancia.

Finalmente el coeficiente calculado fue de  $W = 0.629$ , por lo que queda aceptada la concordancia debido a que este coeficiente es mayor que 0.5.

### 3.9.1. Gráficas estadísticas de resultados de las evaluaciones.

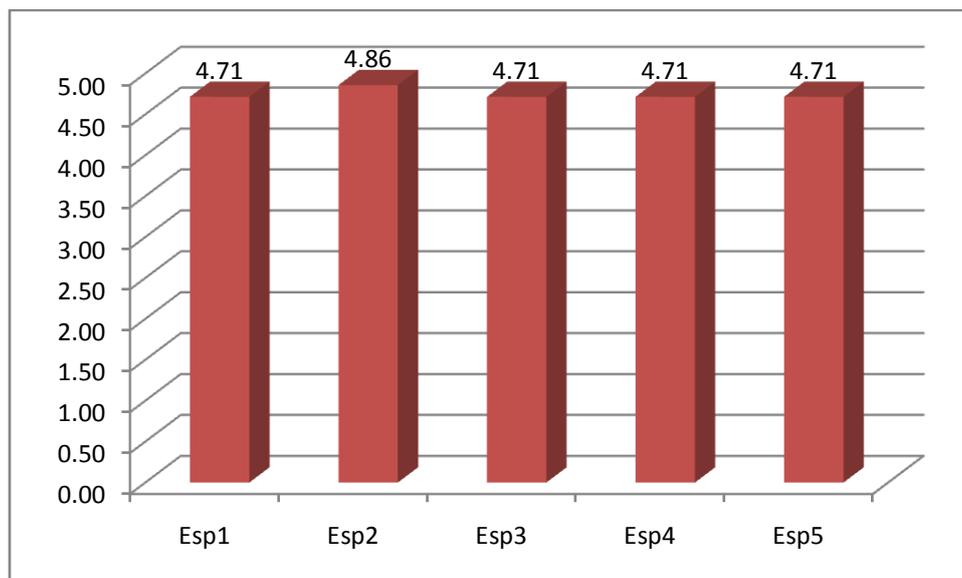


Figura 16: Promedio de evaluaciones por parámetros.

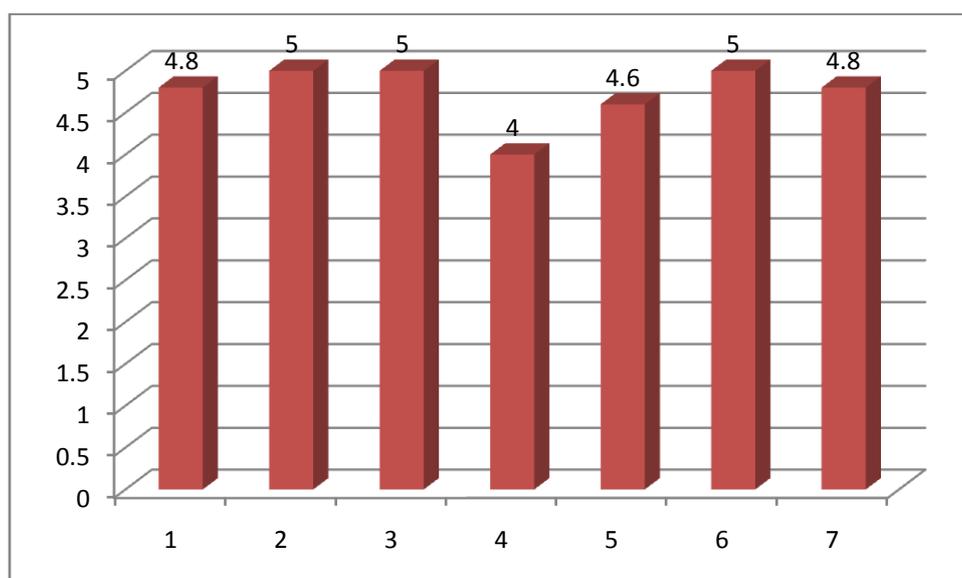


Figura 17: Promedio de evaluaciones por especialistas.

### **3.10. Conclusiones**

Existen gran variedad de criterios, que provocan que el Levantamiento de Requisitos de cada proyecto se realice de una manera diferente y no se realicen las revisiones correspondientes a cada actividad, lo que provoca fallos en algunos casos. Además de que no todos los proyectos consolidan un procedimiento establecido que sirva de guía para la ejecución del Levantamiento de Requisitos a pesar de que ponen en prácticas muchas de las actividades. De ahí la necesidad de desarrollar una propuesta que sirva para la ejecución del Levantamiento de Requisitos logrando la calidad del producto final.

Se diseñó una propuesta para facilitar el correcto desarrollo del Levantamiento de Requisitos, haciéndose énfasis en las revisiones de calidad por cada una de las etapas que integran la Ingeniería de Requisitos. La aplicación de esta podrá contribuir a reducir los fallos identificados en dicha etapa. Como también a estandarizar las actividades del Levantamiento de Requisitos en la Universidad.

La propuesta fue evaluada por un comité de especialistas con experiencias en el tema, que validaron la importancia y necesidad de la aplicación de esta en los proyectos de la universidad, entre otros parámetros que se tuvieron en cuenta. Ellos a su vez proporcionaron recomendaciones que sirvieron para mejorar la propuesta.

### CONCLUSIONES

Como resultado de la investigación y el desarrollo del presente trabajo de diploma se llegaron a las siguientes conclusiones:

Con el desarrollo de este trabajo de diploma se logró desarrollar una propuesta para el Levantamiento de Requisitos Funcionales en función del Aseguramiento de la Calidad de Software. Para lograrlo se realizó un análisis de las etapas de la Ingeniería de Requisitos, donde se propone las revisiones de calidad por parte del equipo de desarrollo, donde el encargado de ponerlas en prácticas es el Asegurador de los Requisitos de Software.

Durante el proceso de desarrollo de software en la fase de Levantamiento de Requisitos es necesario las revisiones continuas de calidad por cada una de las actividades, desarrolladas en la Ingeniería de Requerimientos como son: Elicitación, Análisis, Especificación y Validación, con el objetivo de identificar los requisitos del sistema a desarrollar y culminar con un resultado satisfactorio donde tribute a la calidad del producto final.

Algunas de las buenas prácticas identificadas durante la realización de este trabajo y que deben ser aplicadas en cualquier proyecto durante la fase de Levantamiento de Requisitos son las siguientes:

- Aplicar revisiones de calidad en las actividades de la Ingeniería de Requerimientos como son: Elicitación, Análisis, Especificación y Validación.
- Utilizar técnicas para la captura de información que permita lograr un entendimiento del negocio a informatizar y una buen Levantamiento de Requisitos como: Las Entrevistas, Lluvia de Ideas, Prototipos, Caso de Uso y Archeología de Documentos.
- Utilizar la herramienta Listas de Chequeo para cada una de las actividades de la Ingeniería de Requisitos logrando la validación y aceptación del cliente.
- Utilizar los estándares de calidad ITIL, COBIT y la IEE830 para la mejora continua del proceso de Levantamiento de Requisitos.
- La propuesta define un conjunto de artefactos que son generados durante el Levantamiento de Requisitos que se utilizan como entradas a cada una de las actividades de las etapas de Elicitación, Análisis, Especificación y Validación de los requisitos y otros que son generados como salidas de dichas actividades.

En la propuesta que se realizó se identificaron varios aspectos que no pueden dejar de realizarse durante esta etapa:

- Debe existir un entendimiento del negocio que se desea informatizar entre los clientes y el equipo de desarrollo.
- Las etapas fundamentales que deben realizarse durante el Levantamiento de Requisitos son:
  - Elicitación, con el objetivo de obtener una base de documentación para la identificación de los requisitos funcionales.

- Análisis para llegar a un estudio y negociación con el cliente si surge un nuevo requisito y priorizar los requisitos de acuerdo a las necesidades del cliente.
- Especificación, donde su objetivo es lograr la calidad de los documentos entregados al cliente y la realización de los prototipos no funcionales.
- Validación, para obtener la aceptación del cliente de los documentos que serán entregados.

Es necesaria la participación de los roles definidos por parte del cliente y el equipo de desarrollo, para un mayor entendimiento en el proceso de Levantamiento de Requisitos. Esto no quiere decir que cada uno deba estar presente durante el proceso, sino solo en aquellas actividades que sean necesarios su participación para obtener un mayor resultado.

### RECOMENDACIONES

Como se ha observado los objetivos trazados para este trabajo han sido logrados, sin embargo la propuesta está basada en una investigación realiza sobre la Ingeniería de Requisitos y la Calidad de Software las por lo que se hacen las siguientes recomendaciones:

- Continuar el desarrollo de esta propuesta con la fase de Administración de Requisitos de Software para optimizar los procesos del Levantamiento de Requisitos.
- Continuar el desarrollo de la propuesta con el análisis de los Requisitos no Funcionales para una mayor completitud.
- Aplicar la Propuesta en los proyectos productivos y realizar las mejoras pertinentes a partir de su aplicación para validar su fiabilidad.
- Utilizarla como guía para los proyectos de desarrollo de software de la Universidad.

### REFERENCIAS BIBLIOGRÁFICAS

**20000-2:2005**; **BS ISO/IEC 20000-1:2005** y **BS ISO/IEC. 2008**. [www.bsi-global.com/ICT/Service/bs15000-1.xalter](http://www.bsi-global.com/ICT/Service/bs15000-1.xalter). [En línea] 2008.

**8402, ISO. UNE 66-001-92.**

**9000:2000, ISO. 2000.** *Organización Internacional de Normalización.* 2000.

**9001:2000, ISO.** *Organización Internacional de Normalización.* 2001.

**Andersen, Arthur.** *Calidad de Software.*

**Antonio, Angelica de.** *GESTIÓN, CONTROL Y GARANTÍA.*

**Barzanallana. 2008.** *Rafael- Metodologías de desarrollo de software.* 2008.

**Boehm. 1979.** *Procesos De La Ingenieria De Requerimientos.* 1979.

**Brian, Burke Coyner. 2005.** 2005.

**BROOKS. 2003.** *Ingenieria de Requisitos.* 2003.

**Burke y Coyner. 2005.**

**2008.** *Capability Maturity Model .* 2008.

**Christel, Kang.** *La Ingenieria de Requisitos.*

**Couner, Brian y Burke, Eric. 2005.** *Extreme Programing.* 2005.

**Coyner, Brian y Burke, Eric. 2005.** *Extreme Programing.* 2005.

**Coyner., Brian Burke. 2005.** 2005.

**Eric, Burke. 2005. y Brian., Coyner. 2005.** 2005.

**Feigenbaun, Deming y Juran. 2007.** *Calidad de Software.* 2007.

**Gracia, Joaquin. 2005.** *CMM.* 2005.

**Guide, STARTS. 1987.** *Ingenieria de Requerimientos.* 1987.

**Herrera, J y Lizka, Johany. 2003.** *“Ingeniería de Requerimientos, Ingeniería de Software”.* 2003.

**IEEE-STD-830-1998. 2000..** *Especificaciones de los Requisitos del Software.* 1998. 8. *ISO 9001:2000.* *Organización Internacional de Normalización.* . 2000.

**Ishikawa, Dr. Kaoru.** *La Calidad de Software.*

**ISO, 8402.** *UNE 66-001-92.*

**ISO, 9000:2000.** *Una introducción a ISO 9000:2000.*

**ISO, BS. 2008.** [www.bsi-global.com/ICT/Service/bs15000-1.xalter](http://www.bsi-global.com/ICT/Service/bs15000-1.xalter). [En línea] 2008.

**ITIL. 2008.** [www.itil.co.uk/](http://www.itil.co.uk/). [En línea] 2008.

**Jacobson, Ivar. 2002.** *El Proceso Unificado de Desarrollo de Software. S.I.* 2002.

**Jacobson, y otros. 2002.** *El Proceso Unificado de Desarrollo de Software. S.I.: Pearson Education, S.* 2002.

**Kessler, Mathieu.** *Práctica 0. Introducción al programa SPSS, Departamento de Matemática Aplicada y Estadística.*

**Leite. 1987.** *La ingeniería de requerimientos y sus principales actividades.* 1987.

**Lovelle, Juan Manuel Cueva.** *Calidad del Software.*

**Model, Capability Maturity. 2008.** *Capability Maturity Model .* 2008.

**Pressman, Roger S. 2002.** *Ingeniería del software. Un enfoque práctico. Quinta edición. S.I.: McGraw-Hill/Interamericana de España. S.A.* 2002.

—. **2005.** *Ingeniería del software: Un enfoque práctico. Félix Varela.* 2005.

**Pressman., R. S. 1998.** *Ingeniería del software. Un enfoque práctico. 4ª.* 1998.

**Process, Rational Unified. 2005.** *El Proceso Unificado de Desarrollo de Software .* 2005.

—. **2005.** *El Proceso Unificado de Desarrollo de Software.* 2005.

**Process, Rational Unified. 2005.** *El Proceso Unificado de Desarrollo de Software.* 2005.

**Ralph, R. Young. 2004.** *The Requirements Engineering Handbook. Norwood: ARTECH HOUSE, INC.* 2004.

**Rational. 2003.** *Rational Software Corporation.* 2003.

**Schwaber. 2005.** *Metodología SCRUM.* 2005.

**Torres, Francisco. 2006.** *Micorsoft Solution Framewor.* 2006.

## BIBLIOGRAFÍA

**20000-2:2005**., **BS ISO/IEC 20000-1:2005 y BS ISO/IEC. 2008.** [www.bsi-global.com/ICT/Service/bs15000-1.xalter](http://www.bsi-global.com/ICT/Service/bs15000-1.xalter). [En línea] 2008.

**9000:2000, ISO. 2000.** *Organización Internacional de Normalización.* 2000.

**9001:2000, ISO.** *Organización Internacional de Normalización.* 2001.

**Andersen, Arthur.** *Calidad de Software.*

**Antonio, Angelica de.** *GESTIÓN, CONTROL Y GARANTÍA.*

**Barzanallana. 2008.** *Rafael- Metodologías de desarrollo de software.* 2008.

**Boehm. 1979.** *Procesos De La Ingenieria De Requerimientos.* 1979.

**Brian, Burke Coyner. 2005.** 2005.

**BROOKS. 2003.** *Ingenieria de Requisitos.* 2003.

**2008.** *Capability Maturity Model .* 2008.

**Christel, Kang.** *La Ingenieria de Requisitos.*

**Couner, Brian y Burke, Eric. 2005.** *Extreme Programing.* 2005.

**Coyner, Brian y Burke, Eric. 2005.** *Extreme Programing.* 2005.

**Feigenbaun, Deming y Juran. 2007.** *Calidad de Software.* 2007.

**Gracia, Joaquin. 2005.** *CMM.* 2005.

**Guide, STARTS. 1987.** *Ingenieria de Requerimientos.* 1987.

**Herrera, J y Lizka, Johany. 2003.** *“Ingeniería de Requerimientos, Ingeniería de Software”,.* 2003.

**IEEE-STD-830-1998. 2000..** *Especificaciones de los Requisitos del Software. 1998. 8. ISO 9001:2000. Organización Internacional de Normalización. .* 2000.

**Ishikawa, Dr. Kaoru.** *La Calidad de Software.*

- ISO, 9000:2000.** *Una introducción a ISO 9000:2000.*
- ISO, BS. 2008.** [www.bsi-global.com/ICT/Service/bs15000-1.xalter](http://www.bsi-global.com/ICT/Service/bs15000-1.xalter). [En línea] 2008.
- ITIL. 2008.** [www.itil.co.uk/](http://www.itil.co.uk/). [En línea] 2008.
- Jacobson, Ivar. 2002.** *El Proceso Unificado de Desarrollo de Software. S.I.* 2002.
- Jacobson, y otros. 2002.** *El Proceso Unificado de Desarrollo de Software. S.I.: Pearson Education, S.* 2002.
- Kessler, Mathieu.** *Práctica 0. Introducción al programa SPSS, Departamento de Matemática Aplicada y Estadística.*
- Leite. 1987.** *La ingeniería de requerimientos y sus principales actividades.* 1987.
- Lovelle, Juan Manuel Cueva.** *Calidad del Software.*
- Model, Capability Maturity. 2008.** *Capability Maturity Model .* 2008.
- Pressman, Roger S. 2002.** *Ingeniería del software. Un enfoque práctico. Quinta edición. S.I.: McGraw-Hill/Interamericana de España. S.A.* 2002.
- 2005.** *Ingeniería del software: Un enfoque práctico. Félix Varela.* 2005.
- Pressman., R. S. 1998.** *Ingeniería del software. Un enfoque práctico. 4ª.* 1998.
- Process, Rational Unified. 2005.** *El Proceso Unificado de Desarrollo de Software .* 2005.
- 2005.** *El Proceso Unificado de Desarrollo de Software.* 2005.
- Process, Rational Unified. 2005.** *El Proceso Unificado de Desarrollo de Software.* 2005.
- Ralph, R. Young. 2004.** *The Requirements Engineering Handbook. Norwood: ARTECH HOUSE, INC.* 2004.
- Rational. 2003.** *Rational Software Corporation.* 2003.
- Schwaber. 2005.** *Metodología SCRUM.* 2005.
- Torres, Francisco. 2006.** *Micorsoft Solution Framewor.* 2006.

## ANEXOS

### Anexo 1: Plantilla Plan de Revisiones



## Plan de Revisiones

### Nombre del proyecto

Versión 1.x

### Plantilla de Plan de Revisiones

Versión 1.x Control de versiones:

#### Control de versiones

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

#### Reglas de Confidencialidad

Clasificación: <<Clasificación>>

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

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

Estas reglas son aplicables a las 3 páginas de este documento.

### Tabla de Contenidos

#### Plan de Revisiones

##### 1. Introducción.

###### 1.1. Propósito.

Este documento es el encargado de describir de forma específica, cómo el Equipo de Calidad realizará las revisiones. Su planificación, flujos de trabajo, herramientas y funcionamiento de los roles, deben quedar muy bien definidos.

###### 1.2. Alcance.

Este Plan es elaborado a partir del propuesto por RUP, sólo que ha sido adaptado a las condiciones específicas del proyecto de forma general. Por tanto, su uso es muy particular del proyecto.

###### 1.3. Definiciones, Acrónimos y Abreviaturas.

No existen Definiciones, Acrónimos ni Abreviaturas en este documento.

###### 1.4. Referencias.

Plan de Listas de Chequeo.

###### 1.5. Resumen.

[Resumen de los aspectos del plan]

## 2. Organización.

Rol	Descripción	Conocimientos mínimos
Asegurador de los Requisitos de Software	Es la persona encargada de asumir el rol de asegurador de la calidad y revisor técnico de los requisitos de software. Realizará las pruebas continuas en toda la etapa de desarrollo del Levantamiento de Requisitos, así como llevar el control de la utilización de los estándares de calidad utilizados.	Conocimientos de Gestión de Calidad de Software. Ingeniería de Software. Posee al menos conocimientos básicos del negocio.
Analista Principal	Es la persona encarga de guiar a los demás analistas del sistema. Es responsable de las actividades del especificador de requisitos, de definir los casos de uso y actores del sistema y cómo ellos interactúan; además es el responsable del recogimiento de los requerimientos de usabilidad y diseñar la interfaz de usuario candidata para conocer estos requerimientos.	

Plan de elaboración de Listas de Chequeo.

Esta planificación la hemos recogido en un documento al cual hacemos referencia. Plan de Listas de Chequeo.

## 3. Flujo de Trabajo.

#### 4. Metodología de aplicación.

El Asegurador de los Requisitos de Software del equipo es la persona que se encarga de planificar y organizar las revisiones, orientando los artefactos a revisar así como su entrega y las lista de chequeo que se utilizarán para ello.

Aplica las listas de chequeo orientadas, registrando los resultados de cada elemento.

El Asegurador de los Requisitos de Software conforma el Informe de No Conformidades con todos los problemas que se hayan detectado y las hace llegar al responsable de sus soluciones.

Como última actividad el de los Requisitos de Software realiza un resumen de los resultados y se archivan los resultados para su uso posterior. Queda así concluido el flujo de trabajo de las revisiones.

#### 5. Herramientas.

Inicialmente se usarán las Listas de Chequeo.

## Anexo 2: Plantilla Revisión Técnica Formal



## Revisión Técnica Formal

## Nombre del proyecto

Versión 1.x

## Plantilla de Revisión Técnica Formal

Versión 1.x Control de versiones:

## Control de versiones

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

## Reglas de Confidencialidad

Clasificación: &lt;&lt;Clasificación&gt;&gt;

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

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

Estas reglas son aplicables a las 3 páginas de este documento.

1.Introducción.....	3
1.1.Propósito.....	3
1.2.Alcance .....	3
1.3.Definiciones, Acrónimos y Abreviaturas.....	3
1.4.Referencias.....	3
2.Informe de revisión .....	3
2.1.¿Qué fue lo revisado? .....	3
2.2.¿Quién lo revisó?.....	3
2.3.¿Qué se descubrió? .....	3
3.Conclusiones .....	4
3.1.Lista de sucesos de revisión .....	4
3.2. Procedimiento.....	4

## Introducción

### Propósito

*Este documento tiene como propósito planificar las Reuniones RTF y que los acuerdos y decisiones de la reunión queden resumidos.*

### Alcance

### Definiciones, Acrónimos y Abreviaturas

### Referencias

*Documento a los que se hace referencia*

### Informe de revisión

*En el artefacto de NO conformidades esta el resumen de todos los defectos encontrados, los cuales son el objetivo fundamental de la reunión.*

### ¿Qué fue lo revisado?

### ¿Quién lo revisó?

*Equipo de calidad interna y equipo de calidad de la facultad.*

### ¿Qué se descubrió?

*Cantidad de defectos detectados en la revisión realizada*

### Conclusiones

[Hacer un resumen de todas las pegas y tener en cuenta que los participantes deben decidir si:

1. Aceptan el producto sin posteriores modificaciones, o sea que lo presentado esta correcto
2. Rechazan el producto debido a los errores encontrados, o sea los errores fueron tantos que es mejor rehacer el producto
3. Aceptan el producto provisionalmente, o sea se han encontrado errores menores que deben ser corregidos, pero sin necesidad de otra revisión.]

### Lista de sucesos de revisión

1. Identificar áreas problemáticas dentro del producto
2. Servir como lista de comprobación de puntos de acción que guíe al productor para hacer las correcciones.]

**Procedimiento**

*Procedimiento de RTF.*

**Prueba**

*Tipo de prueba realizada.*

### Anexo 3: Plantilla Minuta de entrega



## Minuta de entrega

### Nombre del proyecto

Versión 1.x

## Plantilla de Minuta de Entrega

Versión 1.x Control de versiones:

### Control de versiones

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

### Reglas de Confidencialidad

Clasificación: <<Clasificación>>

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

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

Estas reglas son aplicables a las **3** páginas de este documento.

---

**Minuta de entrega**

Se hace entrega de los documentos y aplicaciones que se relacionan en la lista. Las aplicaciones y manuales forman parte de entregas anteriormente efectuadas siendo las últimas en las fechas 6 de febrero de 2007 y 6 de marzo de 2007.

<b>Referencia</b>	<b>Archivo</b>
-------------------	----------------

<b>XX-XX-00</b>	XXXXXXXXXX.XXX
-----------------	----------------

---

**Entrega**

NOMBRE Y APELLIDO

CARGO

FIRMA

---

**Recibe**

NOMBRE Y APELLIDO

CARGO

FIRMA

FECHA

---

**Anexo 4: Plantilla Glosario de Terminos.**

## Glosario de términos

### Nombre del proyecto

Versión 1.x

## Plantilla de Glosario de Término

**Entregable**

Versión 1.xControl de versiones:

### Control de versiones

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

Reglas de Confidencialidad

Clasificación: &lt;&lt;Clasificación&gt;&gt;

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

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

Estas reglas son aplicables a las 3 páginas de este documento.

---

## Introducción

### Propósito

*[Definir términos Generales establecidos pro la Dirección de Calidad y Normas de la UCI para los proyectos Productivos]*

### Alcance

*[Todos los proyectos de la UCI]*

### Referencias

*[Lista de documentos a los que se hace referencia en el Plan]*

CÓDIGO	TÍTULO
[1]	DOCUMENTO 1
[2]	DOCUMENTO 2
[3]	MODELO DE DISEÑO - MÓDULO DE ADMINISTRACIÓN V0.0

## Resumen

*[En el glosario aparecen un grupo de términos básicos para los proyectos productivos de la UCI]*

## Definiciones

### Caso de Uso

Secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias. [RUP]

### Caso de Uso arquitectónicamente significativos

Casos de uso que ayudan a mitigar los riesgos más importantes, aquellos que son los más importantes para los usuarios del sistema y aquellos que ayudan a cubrir todas las funcionalidades significativas [RUP]

Caso de uso que, dada las funcionalidades que agrupa, constituye pieza clave para la arquitectura del sistema. Esto puede estar dado por la dependencia que exista de otros casos de uso con él, que

por su complejidad pueda constituir un riesgo para el sistema o por la cantidad de clases que puedan generarse del mismo. [DCS]

### **Caso de Uso significativos para el negocio**

Caso de uso que representa procesos de gran importancia en la línea del negocio. [DCS]

### **No Conformidades**

Problema detectado en un artefacto según:

- Error con respecto a lo definido en artefactos anteriores y/o en lo pactado con el cliente.
- No concordancia con Normas internacionales que deben ser cumplidas por el artefacto.
- Insatisfacción del cliente con el resultado final de un Elemento de Configuración según lo pactado con anterioridad en el proyecto.

La resolución de una no conformidad siempre genera una Orden de Trabajo para el proyecto. [DCS]

### **Solicitud de Cambios**

Para el proceso de desarrollo

Documento en el que queda reflejado un cambio que se desee realizar al sistema, para su aprobación este debe pasar por el Comité de Control de Cambio y deberá tenerse en cuenta el impacto que traerá este cambio sobre el sistema (complejidad del cambio, número de elementos de configuración que afecta, tiempo de implementación, importancia). El cambio puede generarse por un error del sistema o por una mejora al mismo. [DCS]

En el proceso de aceptación

Aparición de nuevos requisitos o de modificaciones a los mismos. [DCS]

### **Prototipo**

Maqueta visual funcional o no de la futura aplicación. Este puede ser una imagen o una aplicación software que simule funcionalidades del software. [DCS]

### **Prueba de Liberación**

<Prueba>

<Liberación>

### **Prueba de Aceptación**

<Prueba>

<Liberación>

### **Expediente del proyecto**

Conjunto de documentos y plantillas estructura siguiendo una jerarquía que constituyen referencia para la documentación de los proyectos de la Universidad de las Ciencias Informáticas

### **Lineamientos Mínimos**

Lineamientos básicos que deben cumplir los proyectos de la Universidad de las Ciencias Informáticas con el fin de obtener un producto final con calidad.

---

**Anexo 5: Plantilla Documento de los procesos del negocio.**

## Documento de los Procesos del Negocio.

### Nombre del proyecto

Versión 1.x

## Plantilla de Procedo del Negocio

**Entregable**

Versión 1.xControl de versiones:

### Control de versiones

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

Reglas de Confidencialidad

Clasificación: &lt;&lt;Clasificación&gt;&gt;

Introducción

Propósito

[Resumen del propósito de este documento]

Alcance

[Breve descripción del alcance del modelo del negocio]

Definiciones, Acrónimos y Abreviaturas

Referencias

[Lista de documentos a los que se hace referencia]

Reglas del negocio a considerar

<Primera regla del negocio>

Actores del negocio

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

Actor	Descripción

Trabajadores del negocio

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

Trabajador	Descripción

## Diagrama de Casos de Uso del Negocio

### Casos de Uso del Negocio

[Descripción literal del caso de uso del negocio y Diagrama de actividades]

<Primer Caso de Uso del Negocio>

<Descripción literal del Caso de Uso del Negocio>

<b>Caso de uso del negocio</b>	<Nombre del caso de uso>
Actores	<Actores del negocio que interactúan con el caso de uso>
Resumen	<Explicar cuándo y cómo se inicia el caso de uso. Describir brevemente el proceso y cuales son sus resultados>
Casos de uso asociados	<En el caso de tener algún caso de uso asociado a este especificar el nombre y tipo de asociación>
<b>Acción del actor</b>	<b>Respuesta del proceso de negocio</b>
<1. Numerar y describir las acciones que realiza el actor.>	<2. Numerar y describir el procesamiento que se realiza en el caso de uso como respuesta a la acción del actor.>
Otras secciones	<Descripción de flujos alternativos al procesamiento básico detallado anteriormente.>

Mejoras propuestas	<Describir las mejoras que se pueden introducir en el procesamiento del caso de uso una vez que se tenga el sistema informático. Si se pretende modificar prácticamente todo el procesamiento actual se debe mostrar un diagrama de actividades del proceso mejorado>
--------------------	---

<Diagrama de actividades del Caso de Uso del Negocio>

Modelo de objetos

---

**Anexo 6: Plantilla Listado de Requisitos Funcionales.**

## Documento de Listado de Requisitos Funcionales.

### Nombre del proyecto

Versión 1.x

## Plantilla de Listado de Requisitos

**Entregable**

Versión 1.x Control de versiones:

### Control de versiones

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

Reglas de Confidencialidad

Clasificación: &lt;&lt;Clasificación&gt;&gt;

Introducción

Propósito

*[Resumen del propósito de este documento]*

Alcance

*[Breve descripción del alcance del modelo del negocio]*

Definiciones, Acrónimos y Abreviaturas

Referencias

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

### **Funcionalidad**

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

#### **<Requisito Funcional 1..n>**

*[Descripción del requisito]*

**Anexo 7: Plantilla de Lista de Riesgos**



**Documento de Lista de Riesgo**

**Nombre del proyecto**

Versión 1.x

**Plantilla de Lista de Riesgos**

**Interno**

Versión 1.xControl de versiones:

---

**Control de versiones**

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

**Reglas de Confidencialidad****Clasificación:** <<Clasificación>>

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

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

Estas reglas son aplicables a las **3** páginas de este documento.

---

**Tabla de Contenidos**

1.	Introducción.....	107
1.1	Propósito .....	107
1.2	Alcance .....	107
1.3	Referencias .....	107
1.4	Resumen.....	
2.	Riesgos.....	108
2.1	<Identificador de riesgo—un nombre o número descriptivo> .....	108
2.1.1	Indicadores .....	108
2.1.2	Estrategia de Mitigación.....	108
2.1.3	Plan de Contingencia.....	108
2.2	<Identificador de riesgo—un nombre o número descriptivo > .....	108
3.	Gestión de Riesgos.....	108

## Introducción

### Propósito

*[Definir términos Generales establecidos pro la Dirección de Calidad y Normas de la UCI para los proyectos Productivos]*

### Alcance

*[Todos los proyectos de la UCI]*

### Referencias

*[Lista de documentos a los que se hace referencia en el Plan]*

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

### Glosario

*[En el glosario aparecen un grupo de términos básicos para los proyectos productivos de la UCI]*

## Riesgos

Riesgo	Tipo de Riesgo	Impacto	Descripción	Probabilidad	Efectos
	<i>[Los tipos de riesgos pueden ser:</i> Tecnológico Personal Organización Herramientas Requerimientos Estimación]	<i>[Lista de impactos en el proyecto o producto.]</i>		<i>[La probabilidad puede ser:</i> Alta Media Baja Muy alta ]	<i>[Los efectos pueden ser:</i> Catastrófico Serias Tolerable Insignificante ]

### <Identificador de riesgo—un nombre o número descriptivo>

Indicadores

*[Describe como monitorear o detectar que el riesgo ha ocurrido o está próximo. Incluye cosas como métricas y umbrales, resultados de prueba, eventos específicos, y así sucesivamente.]*

Estrategia de Mitigación

*[Describe que se hace actualmente en el proyecto para reducir el impacto del riesgo.]*

Plan de Contingencia

*[Describe que curso seguirán las acciones si el riesgo se materializa: solución alternativa, reducción de su efecto, y así sucesivamente.]*

### <Identificador de riesgo—un nombre o número descriptivo >

## Gestión de Riesgos

- *Estimar la probabilidad de ocurrencia*
- *Estimar el impacto sobre el proyecto en una escala del 1 al 5, donde*
  - *1 = bajo impacto sobre el éxito del proyecto*
  - *5= impacto catastrófico sobre el éxito del proyecto*
- *ordenar la tabla por probabilidad e impacto*

Riesgo	Probabilidad	Impacto	Mitigación del riesgo	Monitoreo del riesgo	Administración del riesgo

Nota:

### **Mitigación**

- *¿Cómo se puede evitar el riesgo?*

### **Monitoreo**

- *¿Qué factores podemos vigilar que nos permitan ser capaces de determinar si el riesgo es más o menos probable?*

### **Administración**

- *¿Con qué planes de contingencia contamos si el riesgo se vuelve realidad?*

## Anexo 8: Plantilla Listas de Chequeo.



## Lista de Chequeo de Elicitación

### Nombre del proyecto

Versión 1.x

## Plantilla de Lista de Chequeo

Interno

Versión 1.x Control de versiones:

### Control de versiones

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

### Reglas de Confidencialidad

**Clasificación:** <<Clasificación>>

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

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

Estas reglas son aplicables a las **3** páginas de este documento.

*[Documento para la confección de Listas de Chequeo]*

### *Introducción*

*Esta lista de chequeo cuenta con varios puntos, los cuales serán clasificados antes de ser aplicados, teniendo en cuenta su pertinencia, además de tener un punto donde se referencia las observaciones en cuanto al aspecto a evaluar.*

### *Propósito y objetivos*

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

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

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

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

### *Alcance*

*Esta plantilla es aplicable a cada una de las Revisiones de la actividad de elicitación de [nombre del artefacto] que se desarrollen.*

### *Resumen*

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

### *Forma de Uso:*

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

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

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

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

*Estructura del Documento: Abarca todos los aspectos definidos por el expediente de proyecto o el formato establecido por el proyecto.*

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

*Semántica del documento: Contempla todos los indicadores a evaluar respecto a la ortografía, redacción y demás. N.P. (No Procede): Se usa para especificar que el indicador a evaluar no se puede aplicar en ese caso.*

### Estructura de la lista de chequeo

Estructura del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad elementos afectados	de Comentarios
crítico	1. ¿Está el documento acorde con a la plantilla estándar del proyecto o del expediente de proyecto?				
crítico	2. ¿Contiene las secciones obligatorias definidas en el expediente? (Ver Expediente de Proyecto)				
Elementos definidos por la metodología					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad elementos afectados	de Comentarios
crítico	1. ¿Están todos los requisitos identificados claramente en el documento?				

	2. ¿Debería identificarse algún requisito con más detalle?				
	3. ¿Debería identificarse algún requisito con menos detalles?				
	4. ¿Todos los requisitos identificados se centran en lo que el sistema debe hacer y no como el sistema debe hacerlo?				
crítico	5. ¿Han sido abordadas e identificadas los valores de entradas y salidas?				
	6. ¿Se han enumerado los requisitos identificados incluso los que se derivan de otros requisitos?				
	7. ¿Se puede trazar cada requisito al origen en el entorno del problema, (caso de uso del negocio)?				
	8. ¿No aparece un mismo requisito en más de un lugar del documento de identificación de requisitos?				
crítico	9. ¿No existe contradicción entre lo identificado por un requisito y lo identificado por otro?				

Semántica del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad elementos afectados	de Comentarios
Crítico	1. ¿Ha identificado errores ortográficos?				
Crítico	2. ¿Se entiende claramente lo que se ha identificado en el documento?				
	3. ¿El número de página que aparece en el índice coincide con el contenido que se refleja realmente en dicha página?				
	4. ¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento?				

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Significativa	No Significativa	Recomendación	Estado NC	Respuesta del Equipo Desarrollo
<Nombre del Elemento >	< 1 >	<Descripción de la No Conformidad >	<Descripción del Aspecto correspondiente >	<Etapas de detección del error >	<X>	<X>	<X>	<p>[Se coloca el estado de la NC y la fecha, cada vez que se revise se deja el estado anterior y se coloca el nuevo con la fecha en que se revisó.]</p> <p>RA: Resuelta</p> <p>PD: Pendiente</p> <p>NP: No Procede</p>	<p>[Esta columna se comienza a llenar a partir de la 2da iteración, y es responsabilidad del equipo de desarrollo, quien especifica la conformidad con lo encontrado o no y en caso de no proceder la no conformidad explica por qué.]</p>

### Evaluación del Artefacto

#### **Se aborta la revisión del artefacto revisado si:**

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

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

- ✓ Existen al menos dos indicadores críticos evaluados de mal.
- ✓ Existe más de una falta de ortografía por página o pantalla en caso de ser una interfaz.
- ✓ Incumple con más del 50 % de los indicadores a evaluar de la sección Estructura del Documento que posee la lista de chequeo.
- ✓ Se mantienen las No Conformidades de una revisión a otra.

**Se evalúa de regular la calidad del artefacto revisado** si el artefacto no cumple los criterios para ser abortado y:

- ✓ Existe una No Conformidad crítica.
- ✓ La cantidad de elementos afectados de un indicador evaluado de mal es superior a tres.

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

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

- ✓ No existe ninguna No Conformidad relacionada con indicadores con peso crítico.
- ✓ Si la cantidad de elementos afectados de un indicador que no sea crítico no es mayor que dos.

**Evaluación:** \_\_\_\_\_

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



## Lista de Chequeo de Análisis

### Nombre del proyecto

Versión 1.x

## Plantilla de Lista de Chequeo

**Interno**

Versión 1.x Control de versiones:

### Control de versiones

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

### Reglas de Confidencialidad

**Clasificación:** <<Clasificación>>

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

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

Estas reglas son aplicables a las 211 páginas de este documento.

## Estructura de la lista de chequeo

Estructura del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad elementos afectados	de Comentarios
crítico	1. ¿Está el documento acorde con a la plantilla estándar del proyecto o del expediente de proyecto?				
crítico	2. ¿Contiene las secciones obligatorias definidas en el expediente? (Ver Expediente de Proyecto)				
Elementos definidos por la metodología					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad elementos afectados	de Comentarios
crítico	10. ¿Están incluidas todas las funciones requeridas por el cliente? (Completa)				
	11. ¿Existen conflictos en los requerimientos? (Consistencia)				

<p>12. ¿Tiene alguno de los requerimientos más de una interpretación? (no ambigua)</p>				
<p>13. ¿Está cada requerimiento claramente representado? (entendible)</p>				
<p>14. ¿Pueden los requerimientos ser implementados con la tecnología y el presupuesto disponible? (factible)</p>				
<p>15. ¿Está el documento de Requisitos identificados escrito en un lenguaje apropiado? (clara)</p>				
<p>16. ¿Existe facilidad para hacer cambios en los requerimientos? (modificable)</p>				
<p>17. ¿Está claramente definido el origen de cada requisito? (rastreable)</p>				
<p>18. ¿Pueden los requerimientos ser sometidos a medidas cuantitativas? (verificable)</p>				

Semántica del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	1. ¿Ha identificado errores ortográficos?				
Crítico	2. ¿Se entiende claramente lo que se ha identificado en el documento?				
	3. ¿El número de página que aparece en el índice coincide con el contenido que se refleja realmente en dicha página?				
	4. ¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento?				

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Significativa	No Significativa	Recomendación	Estado NC	Respuesta del Equipo o Desarrollo
<Nombre del Elemento >	< 1 >	<Descripción de la No Conformidad >	<Descripción del Aspecto correspondiente >	<Etapas de detección del error >				<p>[Se coloca el estado de la NC y la fecha, cada vez que se revise se deja el estado anterior y se coloca el nuevo con la fecha en que se revisó.]</p> <p>RA: Resuelta</p> <p>PD: Pendiente</p> <p>NP: No Procede</p>	<p>[Esta columna se comienza a llenar a partir de la 2da iteración, y es responsabilidad del equipo o de desarrollo, quien especifica la conformidad con lo encontrado o no y en caso de no proceder la no conformidad explica por qué.]</p>
					<X>	<X>	<X>		

### Evaluación del Artefacto

#### **Se aborta la revisión del artefacto revisado si:**

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

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

- ✓ *Existen al menos dos indicadores críticos evaluados de mal.*
- ✓ *Existe más de una falta de ortografía por página o pantalla en caso de ser una interfaz.*
- ✓ *Incumple con más del 50 % de los indicadores a evaluar de la sección Estructura del Documento que posee la lista de chequeo.*
- ✓ *Se mantienen las No Conformidades de una revisión a otra.*

**Se evalúa de regular la calidad del artefacto revisado si el artefacto no cumple los criterios para ser abortado y:**

- ✓ *Existe una No Conformidad crítica.*
- ✓ *La cantidad de elementos afectados de un indicador evaluado de mal es superior a tres.*

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

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

- ✓ *No existe ninguna No Conformidad relacionada con indicadores con peso crítico.*
- ✓ *Si la cantidad de elementos afectados de un indicador que no sea crítico no es mayor que dos.*

**Evaluación:** \_\_\_\_\_

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



## Lista de Chequeo de Especificación de Requisitos

### Nombre del proyecto

Versión 1.x

## Plantilla de Lista de Chequeo

Interno

Versión 1.x Control de versiones:

### Control de versiones

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

### Reglas de Confidencialidad

**Clasificación:** <<Clasificación>>

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

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

Estas reglas son aplicables a las 211 páginas de este documento.

*[Documento para la confección de Listas de Chequeo]*

### *Introducción*

*Esta lista de chequeo cuenta con varios puntos, los cuales serán clasificados antes de ser aplicados, teniendo en cuenta su pertinencia, además de tener un punto donde se referencia las observaciones en cuanto al aspecto a evaluar.*

### *Propósito y objetivos*

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

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

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

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

### *Alcance*

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

### *Resumen*

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

### *Forma de Uso:*

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

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

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

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

*Estructura del Documento: Abarca todos los aspectos definidos por el expediente de proyecto o el formato establecido por el proyecto.*

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

### Estructura de la lista de chequeo

Estructura del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
crítico	1. ¿Está el documento acorde con a la plantilla estándar del proyecto o del expediente de proyecto?				
crítico	2. ¿Contiene las secciones obligatorias definidas en el expediente? (Ver Expediente de Proyecto)				
Elementos definidos por la metodología					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
crítico	19. ¿Están todos los requisitos redactados de forma simple y clara para aquellos que vayan a consultarlo en un futuro?				

---

	20. ¿Debería especificarse algún requisito con más detalle?				
--	---	--	--	--	--

	21. ¿Debería especificarse algún requisito con menos detalles?				
	22. ¿Todos los requisitos identificados se centran en lo que el sistema debe hacer y no como el sistema debe hacerlo?				
crítico	23. ¿Han sido abordadas e identificadas los valores de entradas y salidas?				
	24. ¿Han sido incluidos las respuestas válidas y no válidas de los valores de entrada?				
	25. ¿Se han identificado los requerimientos de software?				
	26. ¿Han sido identificadas las restricciones de diseño e implementación?				
	27. ¿Se puede verificar cada requisito? (Un requisito se dice que es verificable si existe algún proceso no excesivamente costoso por el cual una persona o una máquina pueda chequear que el software satisface dicho requerimiento, ejemplo la especificación del caso de uso).				

	28. ¿Se han enumerado los requisitos incluso los que se derivan de otros requisitos?				
	29. ¿Se puede trazar cada requisito al origen en el entorno del problema, (caso de uso del negocio)?				
	30. ¿Se han especificado todos los posibles cambios en los requisitos, incluyendo la probabilidad de cambio?				
	31. ¿No aparece un mismo requisito en más de un lugar del documento de especificación?				
crítico	32. ¿No existe contradicción entre lo especificado por un requisito y lo especificado por otro?				
	33. ¿Existe correspondencia entre el modelo de caso de uso, las Especificaciones Suplementarias y las especificaciones de requerimientos?				

Semántica del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	1. ¿Ha identificado errores ortográficos?				
Crítico	2. ¿Se entiende claramente lo que se ha especificado en el documento?				
	3. ¿El número de página que aparece en el índice coincide con el contenido que se refleja realmente en dicha página?				
	4. ¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento?				

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Significativa	No Significativa	Recomendación	Estado NC	Respuesta del Equipo o Desarrollo
<Nombre del Elemento >	< 1 >	<Descripción de la No Conformidad >	<Descripción del Aspecto correspondiente >	<Etapas de detección del error >				<p>[Se coloca el estado de la NC y la fecha, cada vez que se revise se deja el estado anterior y se coloca el nuevo con la fecha en que se revisó.]</p> <p>RA: Resuelta</p> <p>PD: Pendiente</p> <p>NP: No Procede</p>	<p>[Esta columna se comienza a llenar a partir de la 2da iteración, y es responsabilidad del equipo o de desarrollo, quien especifica la conformidad con lo encontrado o no y en caso de no proceder la no conformidad explica por qué.]</p>

### Evaluación del Artefacto

#### **Se aborta la revisión del artefacto revisado si:**

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

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

- ✓ *Existen al menos dos indicadores críticos evaluados de mal.*
- ✓ *Existe más de una falta de ortografía por página o pantalla en caso de ser una interfaz.*
- ✓ *Incumple con más del 50 % de los indicadores a evaluar de la sección Estructura del Documento que posee la lista de chequeo.*
- ✓ *Se mantienen las No Conformidades de una revisión a otra.*

**Se evalúa de regular la calidad del artefacto revisado si el artefacto no cumple los criterios para ser abortado y:**

- ✓ *Existe una No Conformidad crítica.*
- ✓ *La cantidad de elementos afectados de un indicador evaluado de mal es superior a tres.*

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

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

- ✓ *No existe ninguna No Conformidad relacionada con indicadores con peso crítico.*
- ✓ *Si la cantidad de elementos afectados de un indicador que no sea crítico no es mayor que dos.*

**Evaluación:** \_\_\_\_\_

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



## Lista de Chequeo de Modelo de Caso de Uso del Sistema

### Nombre del proyecto

Versión 1.x

### Plantilla de Lista de Chequeo

Interno

Versión 1.x Control de versiones:

#### Control de versiones

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

#### Reglas de Confidencialidad

**Clasificación:** <<Clasificación>>

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

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

Estas reglas son aplicables a las 211 páginas de este documento.

*[Documento para la confección de Listas de Chequeo]*

### *Introducción*

*Esta lista de chequeo cuenta con varios puntos, los cuales serán clasificados antes de ser aplicados, teniendo en cuenta su pertinencia, además de tener un punto donde se referencia las observaciones en cuanto al aspecto a evaluar.*

### *Propósito y objetivos*

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

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

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

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

### *Alcance*

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

### *Resumen*

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

### *Forma de Uso:*

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

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

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

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

*Estructura del Documento: Abarca todos los aspectos definidos por el expediente de proyecto o el formato establecido por el proyecto.*

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

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

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

## Estructura de la lista de chequeo

Estructura del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
crítico	1. ¿Está el documento acorde con a la plantilla estándar del proyecto o del expediente de proyecto?				
crítico	2. ¿Contiene las secciones obligatorias definidas en el expediente? (Ver Expediente de Proyecto)				
Elementos definidos por la metodología					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Aspectos Generales					
	1. ¿Cada caso de uso registra claramente lo que el sistema debe hacer?				
crítico	2. ¿Están clasificado los casos de uso que definen la arquitectura básica del sistema?				
	3. ¿Se ha identificado los casos de uso que darán soporte y mantenimiento al sistema?				

crítico	4. ¿Se ha descrito con precisión todas las alternativas o excepciones?				
	5. ¿Están clasificados los casos de uso que sirven de apoyo a los caso de uso que cubren las principales funciones que el sistema debe realizar?				
	6. ¿Están clasificados los casos de uso que no son claves para la arquitectura?				

**Nombre del Caso de Uso**

crítico	1. ¿Está en infinitivo y refleja de manera clara el objetivo del usuario sobre el sistema?				
crítico	2. ¿El nombre del caso de uso es único?				
	3. ¿El nombre del caso de uso es intuitivo?				

**Descripción**

	1. ¿El resumen dice como se inicia, como termina y las operaciones principales que realiza el caso de uso?				
--	--	--	--	--	--

**Precondición**

	1. ¿Se escribe una precondición si y solo si a partir de la ocurrencia de un suceso determinado comienza el caso de uso?				
	2. ¿La precondición es válida tanto para flujos básicos como flujos alternativos?				

**Poscondición**

	1. ¿La poscondición plasma cambios que suceden en el sistema al terminarse de ejecutar el caso de uso?				
--	--	--	--	--	--

**Complejidad del CU**

	1. ¿Se especifica la complejidad del caso de uso?				
--	---	--	--	--	--

**Forma de presentar la información**

crítico	1. ¿Está descrito el caso de uso en presente?				
---------	---	--	--	--	--

crítico	2. ¿Se describe de manera comprensible y detallada las acciones del actor frente al sistema? ¿Está lo más parecido a un manual de ayuda?				
---------	---	--	--	--	--

**Actores del CU**

crítico	1. ¿El Caso de Uso está relacionado con al menos un actor?				
---------	--	--	--	--	--

crítico	2. ¿Si hay dos actores interactuando con el caso de uso está generalizado en uno solo?				
---------	--	--	--	--	--

crítico	3. ¿Si el caso de uso es abstracto (include, extende, generalización-especialización), no lo inicializa ningún actor?				
---------	---	--	--	--	--

**Flujo Básico**

	1. ¿Comienza diciendo "El caso de uso se inicia cuando el actor..."?				
--	--	--	--	--	--

	2. ¿Termina diciendo en un evento independiente "El caso de uso termina"?				
	3. ¿No existen abreviaturas?				
crítico	4. ¿Las partes del flujo de eventos que se repiten en otro caso de uso se especifican como un Caso de Uso incluido?				
crítico	5. ¿Si las alternativas que se describen casi nunca ocurren o son alternativas comunes a otros casos de uso se especifican como un Caso de Uso extendido?				
	6. ¿Si existe un proceso general y a partir de él se especializan otros se especifican como una generalización/especialización?				

### Flujo Alternativo

crítico	1. ¿Las alternativas o excepciones se reflejan como flujos alternos?				
	2. ¿En todos los CU que se introducen datos tienen un flujo alternativo donde el sistema valida la integridad de los datos que se introducen y muestra un mensaje en caso de que los datos estén incompletos?				
	3. ¿Los flujos alternativos se nombran con el número del paso que lo generó en el flujo básico, una letra, ordenados alfabéticamente que lo produjo?				

crítico	4. ¿En la sección flujos alternativos se describen todas las excepciones que existan por muy evidentes que parezcan?				
<b>Casos de uso incluidos y extendidos</b>					
crítico	1. ¿Al describir el caso de uso base se mencionan todos los casos de Uso que Extienden, se incluyen o se generalizan del Caso de Uso?				
crítico	2. ¿La descripción de los Casos de Uso incluidos, extendidos y especializados se realiza aparte?				
<b>Navegabilidad</b>					
crítico	1. ¿La navegabilidad en los caso de uso de inclusión se inicia desde el caso uso base hasta el caso de uso incluido?				
crítico	2. ¿La navegabilidad en los caso de uso de extensión se inicia desde el caso uso extendido hasta el caso de uso base?				
	3. ¿La navegabilidad en la generalización/especialización se inicia desde el caso de uso especializado a al generalizado y se representa con una relación de herencia?				

Relaciones					
crítico	1. ¿Las relaciones de inclusión y extensión entre los caso de uso se han representado con línea discontinua?				
Información General					
crítico	1. ¿El diagrama de casos de uso expresa en detalles y claramente lo que debe hacer el sistema?				
	2. Si la modelación de las interacciones con el sistema es muy extensa ¿ha empleado los paquetes de caso de uso?				
Semántica del documento					
Peso	Indicadores a Evaluar	Eval	(NP)	Cantidad de elementos afectados	Comentarios
Crítico	5. ¿Ha identificado errores ortográficos?				
Crítico	6. ¿Se entiende claramente lo que se ha especificado en el documento?				
	7. ¿El número de página que aparece en el índice coincide con el contenido que se refleja realmente en dicha página?				
	8. ¿El total de páginas que aparecen en las reglas de confidencialidad coincide con el total de páginas que tiene el documento?				

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Significativa	No Significativa	Recomendación	Estado NC	Respuesta del Equipo Desarrollo
<Nombre del Elemento>	< 1 >	<Descripción de la No Conformidad >	<Descripción del Aspecto correspondiente>	<Etapas de detección del error>	<X>	<X>	<X>	<p>[Se coloca el estado de la NC y la fecha, cada vez que se revise se deja el estado anterior y se coloca el nuevo con la fecha en que se revisó.]</p> <p>RA: Resuelta</p> <p>PD: Pendiente</p> <p>NP: No Procede</p>	<p>[Esta columna se comienza a llenar a partir de la 2da iteración, y es responsabilidad del equipo de desarrollo, quien especifica la conformidad con lo encontrado o no y en caso de no proceder la no conformidad explica por qué.]</p>

### Evaluación del Artefacto

#### **Se aborta la revisión del artefacto revisado si:**

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

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

- ✓ Existen al menos dos indicadores críticos evaluados de mal.
- ✓ Existe más de una falta de ortografía por página o pantalla en caso de ser una interfaz.
- ✓ Incumple con más del 50 % de los indicadores a evaluar de la sección Estructura del Documento que posee la lista de chequeo.
- ✓ Se mantienen las No Conformidades de una revisión a otra.

**Se evalúa de regular la calidad del artefacto revisado** si el artefacto no cumple los criterios para ser abortado y:

- ✓ Existe una No Conformidad crítica.
- ✓ La cantidad de elementos afectados de un indicador evaluado de mal es superior a tres.

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

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

- ✓ No existe ninguna No Conformidad relacionada con indicadores con peso crítico.
- ✓ Si la cantidad de elementos afectados de un indicador que no sea crítico no es mayor que dos.

**Evaluación:** \_\_\_\_\_

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

---

**Anexo 9: Plantilla Informe de No Conformidades.**

## Informe de No Conformidades

### Nombre del proyecto

Versión 1.x

## Plantilla de No Conformidades

Elemento a probar &lt;Nombre del artefacto a probar&gt;

Versión 1.xControl de versiones:

Control de Versiones

Fecha	Versión	Descripción	Autor
<00/00/0000>	<1.0>	<Descripción>	<Autor>

### **Descripción General**

*[Descripción de Aspectos Generales a tener en cuenta a la hora de realizar el diseño de las pruebas, incidencias en el momento de su desarrollo y otros aspectos relevantes.]*

### **Elementos probados**

*[Descripción general o lista de los Elementos Probados, y otros aspectos importantes a tener en cuenta a la hora analizar las No Conformidades Detectadas.]*

### **Elementos no probados y causas**

*[Descripción de Aspectos Generales a tener en cuenta a la hora de realizar el diseño de las pruebas, incidencias en el momento de su desarrollo y otros aspectos relevantes.]*

Registro de defectos y dificultades detectados

Elemento	No	No conformidad	Aspecto correspondiente	Etapas de detección	Clasificación	Estado NC	Respuesta del Equipo Desarrollo
<Nombre del Elemento >	< 1 >	<Descripción de la No Conformidad>	<Descripción del Aspecto correspondiente>	<Etapas de detección del error>	<p>S: Significativa</p> <p>NS: No Significativa</p> <p>R: Recomendación</p>	<p>[Se coloca el estado de la NC y la fecha, cada vez que se revise se deja el estado anterior y se coloca el nuevo con la fecha en que se revisó.]</p> <p>RA: Resuelta</p> <p>PD: Pendiente</p> <p>NP: No Procede</p>	<p>[Esta columna se comienza a llenar a partir de la 2da iteración, y es responsabilidad del equipo de desarrollo, quien especifica la conformidad con lo encontrado o no y en caso de no proceder la no conformidad explica por qué.]</p>

[La NC puede tener solo una de las tres clasificaciones: Significativa, No Significativa o Recomendación]

---

**Anexo 10: Plantilla Documento Visión.**

## Documento Visión

### Nombre del proyecto

Versión 1.x

## Plantilla de Documento Visión

Versión 1.xControl de versiones:

#### Control de Versiones

Fecha	Versión	Descripción	Autor
<00/00/0000>	<1.0>	<Descripción>	<Autor>

---

**Tabla de contenidos**

1.	Introducción.....	148
1.1	Alcance.....	148
1.2	Definiciones, Acrónimos, y Abreviaturas1.1 Alcance.....	148
1.3	Referencias.....	148
2.	Posición.....	148
2.1	Oportunidad de Negocio.....	148
2.2	Establecimiento del problema.....	148
2.3	Declaración de posición del producto.....	149
3.	Descripción de los usuarios y todos los involucrados.....	149
3.1	Estudio de mercado.....	149
3.2	Entorno de usuario.....	149
3.3	Perfil de los involucrados .....	150
3.3.1	<Nombre del involucrado>.....	150
3.4	Perfil de los usuarios.....	150
3.4.1	<Nombre del usuario>.....	150
3.5	Principales necesidades de los Involucrados y los usuarios .....	150
3.6	Alternativas y competencias.....	151
3.6.1	<Competidor 1>.....	151
3.6.2	<Competidor 2>.....	151
4.	Visión general del producto.....	151
4.1	Perspectivas del producto .....	151
4.2	Resumen de capacidades .....	152
4.3	Suposiciones y dependencias.....	154
4.4	Costo y precio.....	154
5.	Características básicas del proyecto.....	154

**Introducción**

*[La introducción a la **visión** facilita una vista general de todo el documento. Incluye el propósito, alcance, definiciones, acrónimos, abreviaturas, referencias, y una vista general de esta **Visión**.]*

**Alcance****Definiciones, Acrónimos, y Abreviaturas****Referencias**

*[Lista de documentos a los que se hace referencia en el Plan]*

<b>Código</b>	<b>Título</b>
[1]	Documento 1
[2]	Documento 2
[3]	Modelo de Diseño - Módulo de Administración v0.0

**Posición****Oportunidad de Negocio**

*[Breve descripción de las oportunidades de negocio con el proyecto. Donde se expresen los elementos que hacen atractivo al proyecto, las coyunturas políticas, económicas o específicas del sector del mercado a que pudiera estar dirigida.]*

**Establecimiento del problema**

*[Resumen de los posibles problemas que pueden ser resueltos con el proyecto. Puede utilizar la siguiente tabla]*

<b>Problema</b>	<i>[descripción del problema]</i>
<b>Afectados</b>	<i>[involucrados afectados con el problema]</i>
<b>Impacto que provoca</b>	<i>[Cuál es el impacto del problema?]</i>
<b>Una solución satisfactoria pudiera implicar</b>	<i>[lista de los beneficios que pudiera tener]</i>

**Declaración de posición del producto**

*[Resumen de la única área en la que el producto intenta incidir. Pudiera utilizarse el siguiente formato:]*

<b>Para</b>	<i>[usuario apuntado]</i>
<b>Quien</b>	<i>[necesidades y oportunidades]</i>
<b>El nombre del producto</b>	<i>[cuál es el impacto del problema?]</i>
<b>Que</b>	<i>[resumen de los principales beneficios]</i>
<b>A diferencia</b>	<i>[alternativa competitiva fundamental]</i>
<b>Nuestro producto</b>	<i>[principales características que lo diferencia]</i>

*[Pretende comunicar de forma resumida porque es importante el proyecto]*

**Descripción de los usuarios y todos los involucrados****Estudio de mercado**

*[Breve resumen de las principales características del Mercado que motivan el proyecto. Da respuesta a las siguientes preguntas estratégicas:*

- *¿Cuál es la reputación de tu organización en este mercado?*
- *¿Qué quieres lograr?*
- *¿Desarrollar este producto o dar este servicio influye en el Cumplimiento de tus metas?]*

**Entorno de usuario**

*[Destalles del entorno de trabajo del usuario principal. Pudieran seguirse estas sugerencias:*

- *Número de personas involucradas en resolver una tarea. ¿Esto está cambiando?*
- *¿Cuán largo es el ciclo de una tarea? ¿Cuánto tiempo consume en cada actividad? ¿Esto está cambiando?*
- *¿Existe un solo entorno de trabajo?*
- *¿Qué plataforma de software se usan hoy? ¿Cuáles consideras Plataformas futuras?*

- *¿Qué otras aplicaciones están en uso? ¿La aplicación a realizar debe integrarse con estas? ]*

### Perfil de los involucrados

*[Detalles de cada involucrado]*

<Nombre del involucrado>

<b>Descripción</b>	<i>[Breve descripción]</i>
<b>Tipo</b>	<i>[Clasificar al involucrado como experto, gurú, accionista, etc]</i>
<b>Responsabilidades</b>	<i>[Principales responsabilidades relacionadas con el proyecto]</i>
<b>Criterio de éxito</b>	<i>[Cómo define el criterio de éxito?]</i>
<b>Implicación</b>	<i>[Qué rol juega el interesado en el proyecto.]</i>
<b>Entregable</b>	<i>[Requiere algún entregable adicional?.]</i>
<b>Comentarios</b>	

### Perfil de los usuarios

*[Detalles de cada usuario]*

<Nombre del usuario>

<b>Descripción</b>	<i>[Breve descripción]</i>
<b>Tipo</b>	<i>[Clasificar al involucrado como experto, gurú, accionista, etc]</i>
<b>Responsabilidades</b>	<i>[Principales responsabilidades relacionadas con el proyecto]</i>
<b>Criterio de éxito</b>	<i>[Cómo define el criterio de éxito?]</i>
<b>Implicación</b>	<i>[Qué rol juega el interesado en el proyecto.]</i>
<b>Entregable</b>	<i>[Requiere algún entregable adicional?.]</i>
<b>Comentarios</b>	

### Principales necesidades de los Involucrados y los usuarios

*[Lista de problemas claves con las soluciones existentes y como son percibidas por los involucrados .Aclarar los siguientes asuntos para cada problema:*

*Razones del problema*

*¿Cómo se resuelve ahora el problema?*

*¿Que solución el involucrado quiere?*

*Esto es importante para entender la importancia relativa que da el involucrado sobre la solución de cada problema. Clasificando y acumulando votos técnicos que indiquen la relación problemas a ser resueltos vs asuntos que ellos quieren dirigir.*

*Completar la siguiente tabla –Si usa el RequisitePro para capturar las necesidades esto puede ser extraído de la herramienta.]*

<b>Necesidad</b>	<b>Prioridad</b>	<b>Inquietud</b>	<b>Solución actual</b>	<b>Solución propuesta</b>
------------------	------------------	------------------	------------------------	---------------------------

### **Alternativas y competencias**

*[Identificar las alternativas que perciben los involucrados están disponibles. Esto puede incluir comprar un producto competidor, una solución hecha por ellos mismos o simplemente el mantenimiento del estatus. Lista cualquier opción de competencia que exista y se conozca, o pueda convertirse en disponible. Incluye las principales fortalezas y debilidades de cada competidor percibido por el involucrado.]*

<Competidor 1>

<Competidor 2>

### **Visión general del producto**

*[Esta sección proporciona en alto nivel una visión de las capacidades del producto, interfaces con otras aplicaciones y configuración del sistema. Esta sección usualmente se divide en tres sub secciones tal como sigue:*

*Perspectivas del producto*

*Funciones del producto*

*Suposiciones y dependencias]*

### **Perspectivas del producto**

*[Esta sub sección puede poner la perspectiva de este producto relativo a otros productos o ambientes de usuarios. Si el producto es independiente y totalmente auto controlado*

*declárelo aquí. Si el producto es un componente de un gran sistema, entonces esta subsección puede relatar cómo estos sistemas interactúan y puede identificar interfaces relevantes entre los sistemas. Una forma fácil para mostrar los principales componentes del gran sistema, interconexiones e interfaces externas, son los diagramas de bloque.]*

### **Resumen de capacidades**

*[Resumen de los principales beneficios y características que el producto debe tener. Por ejemplo, para un sistema de soporte al cliente se puede usar esta parte para enunciar los problemas de documentación, camino y estado reportando con el nivel de detalle que cada función requiere.*

*Organizar las funciones así como la lista es entendible para el cliente o cualquier otra persona que lea el documento por primera vez. Una simple tabla lista las claves para el beneficio y soporta las características.*

---

**SISTEMA DE SOPORTE AL CLIENTE**

<b>Beneficios del cliente</b>	<b>Características soportadas</b>
Nuevo equipo de soporte que pueda acelerar la atención.	Base de conocimiento asistida por el personal de soporte reconociendo rápidamente los errores conocidos y trabajando sobre ellos.
La satisfacción del cliente es mejorada porque nada lo obtiene a través de los cracks.	Los problemas son únicamente detallados, clasificados y seguidos a través del proceso de resolución. Ocurre una notificación automática ante cualquier proceso que esté envejeciendo.
La administración puede identificar las áreas de problemas y graduar las cargas de trabajo de los equipos.	Los reportes de tendencias y distribución permiten un alto nivel de revisión del estado de los problemas.
Equipos de soporte distribuido pueden trabajar juntos en la solución de un problema.	Los servidores de réplicas permiten que el flujo de información sea compartido a través de la empresa.
Los clientes pueden ayudarse ellos mismos, bajando el costo y mejorando el tiempo de respuesta.	Las bases de conocimientos pueden estar disponibles en Internet. Incluye hipertexto, capacidades de búsqueda.

---

### **Suposiciones y dependencias**

*[Lista cada factor que afecta el estado de las características. La lista de suposiciones que, si es cambiada, alterará el documento. Por ejemplo, una suposición puede ser que estará disponible el sistema operativo y el hardware requerido para el software. Si el sistema operativo no está disponible el documento de la visión debe ser cambiado.]*

### **Costo y precio**

*[Para comercializar el producto a clientes externos y para muchas aplicaciones en casa, los asuntos de costo y precio pueden directamente impactar la definición e implementación del producto. En esta sección, se registran las restricciones de costo y precio que son relevantes. Por ejemplo: costo de distribución (# de discos, #-de CD, etc) u otros costos para restricciones de buena comercialización (manuales,) otros elementos de costo (manuales, ensamblaje packaging)*

### **Características básicas del proyecto**

*[Lista y breve descripción de las características básicas del producto. Estas características básicas son las capacidades elementales con las que debe cumplir el producto para satisfacer al cliente. Cuando se desarrolle el modelo de Casos de uso se describen el resto de las especificaciones necesarias.*

*Se recomienda no tener menos de 25 ni más de 99 características.]*

---

**Anexo 11: Plantilla Documento de Especificación de los Requisitos de Software****Especificación de Requisitos de Software****Nombre del proyecto**

Versión 1.x

**Plantilla de Especificación de Requisitos de Software**

Versión 1.x Control de versiones:

**Control de versiones**

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

**Reglas de Confidencialidad****Clasificación:** <<Clasificación>>

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

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

Estas reglas son aplicables a las **3** páginas de este documento.

**Tabla de Contenidos**

1. Introducción ..... 158

    1.1 Alcance ..... ¡Error! Marcador no definido.

    1.2 Definiciones, Acrónimos y Abreviaturas ..... ¡Error! Marcador no definido.

    1.3 Referencias ..... ¡Error! Marcador no definido.

2. Funcionalidad ..... 158

    2.1 <Requisito Funcional 1..n>..... 159

13. Estándares Aplicables..... 159

---

## Introducción

## Alcance

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

## Definiciones, Acrónimos y Abreviaturas

## Referencias

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

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

## Funcionalidad

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

### <Requisito Funcional 1..n>

*[Descripción del requisito]*

### Restricciones de diseño

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

### Estándares Aplicables

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

Anexo 12 Plantilla Registro de Revisiones



**Registro de Revisiones**

**Nombre del proyecto**

Versión 1.x

**Plantilla de Registro de Revisiones**

Versión 1.xControl de versiones:

---

**Control de versiones**

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

**Reglas de Confidencialidad**

**Clasificación:** <<Clasificación>>

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

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

Estas reglas son aplicables a las **3** páginas de este documento.

Tabla de Contenido

1.Introducción..... 3

    1.1.Propósito..... 3

    1.2.Alcance ..... 3

    1.3.Definiciones, Acrónimos y Abreviaturas..... 3

    1.4.Referencias..... 3

2.Informe de revisión ..... 3

    2.1.¿Qué fue lo revisado? ..... 3

    2.2.¿Quién lo revisó?..... 3

    2.3.¿Qué se descubrió? ..... 3

3.Conclusiones ..... 4

    3.1.Lista de sucesos de revisión ..... 4

    3.2. Procedimiento..... 4

## Introducción

### Propósito

### Alcance

### Definiciones, Acrónimos y Abreviaturas

### Referencias

#### ➤ Informe de revisión

Realizar un informe de las revisiones realizadas que se encuentran registradas en el Informe de No Conformidades.

- ¿Qué fue lo revisado?
- ¿Quién lo revisó?
- ¿Qué se descubrió?

## Conclusiones

[Hacer un resumen de todas las pegs y tener en cuenta que los participantes deben decidir si:

- Aceptan el producto sin posteriores modificaciones, o sea que lo presentado esta correcto
- Rechazan el producto debido a los errores encontrados, o sea los errores fueron tantos que es mejor rehacer el producto
- Aceptan el producto provisionalmente, o sea se han encontrado errores menores que deben ser corregidos, pero sin necesidad de otra revisión.]

## Lista de sucesos de revisión

- Identificar áreas problemáticas dentro del producto
- Servir como lista de comprobación de puntos de acción que guíe al productor para hacer las correcciones.]

## Procedimiento

Procedimiento de RTF.

---

**Anexo 13 Plantilla Minuta de Aceptación****MINUTA DE ACEPTACION**

En la Ciudad de Caracas, a los \_\_\_\_ días del mes de mayo del 2007

**De una parte**, el Ministerio del Poder Popular para ....., de la República Bolivariana de Venezuela, representado en este acto por el ciudadano venezolano **[Nombres y Apellidos]**, mayor de edad, portador de la cédula de identidad **Nº. 000000**, quien a los fines y efectos derivados del presente documento se denominará como **“La Parte Venezolana”**, y de otra Parte, la **Sociedad Mercantil ALBET Ingeniería y Sistemas S.A.**, conocida de forma abreviada como **ALBET, S.A.** de la República de Cuba, representada en este acto por la ciudadana cubana **[Nombres y Apellidos]**, mayor de edad, titular del Pasaporte **Nº. 00000**, que a los fines y efectos derivados del presente documento se denominará **La Parte Cubana**.

**Primero:** Que en cumplimiento del **Contrato.....**; han sido efectuadas las actividades..... que se describen , **Las Partes DECLARAN:**

**CONSIDERANDO:** Que se han efectuado las actividades siguientes:

1. \_

2. \_

**CONSIDERANDO:** Que las actividades realizadas han sido desarrolladas con la calidad requerida y bajo las condiciones pactadas y aprobadas por **Las Partes**.

**CONSIDERANDO:** Que las actividades que se han ejecutado cumplen con los requerimientos de **La Parte Venezolana**.

**CONSIDERANDO:** Que **La Parte Cubana** ha entregado la documentación que avala la ejecución de este acto a **La Parte Venezolana**.

**POR TANTO: Las Partes** acuerdan formalizar mediante la presente Acta, **Aceptadas** las actividades que han sido ejecutadas en esta fecha.

Y para que así conste, se extiende la presenta **Acta** en dos (3) ejemplares, rubricados por **Las Partes**.

**Por la PARTE VENEZOLANA**

**Por la PARTE CUBANA**

---

**Nombres y Apellidos**

Gerente General de Proyecto

---

**Nombres y Apellidos**

Gerente General de Proyecto

---

**Anexo 14 Plantilla Modelo de Caso de Uso del Sistema**

## Modelo de Caso de Uso del Sistema

### Nombre del proyecto

Versión 1.x

## Plantilla de Modelo de Caso de Uso del Sistema

Versión 1.x Control de versiones:

### Control de versiones

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

**Tabla de contenidos**

1. Introducción..... 168

    1.1 Propósito ..... 168

    1.2 Alcance ..... 168

    1.3 Definiciones, Acrónimos y Abreviaturas..... 168

    1.4 Referencias ..... 168

2. Actores del Sistema ..... 168

3. Diagrama de Casos de Uso del Sistema ..... 168

4. Especificación de los Casos de Uso..... 168

    4.1 <Primer Caso de Uso del Sistema>..... 168

        4.1.1 Descripción de Casos de Uso 168

---

## Introducción

### Propósito

[Resumen del propósito de este documento]

### Alcance

[Breve descripción del alcance del modelo del negocio]

### Definiciones, Acrónimos y Abreviaturas

### Referencias

[Lista de documentos a los que se hace referencia]

### Actores del Sistema

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

Actor	Descripción

### Diagrama de Casos de Uso del Sistema

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

### Especificación de los Casos de Uso

#### <Primer Caso de Uso del Sistema>

##### *Descripción de Casos de Uso*

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

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

<b>Caso de Uso:</b>	
<b>Actores:</b>	
<b>Resumen:</b>	
<b>Precondiciones:</b>	
<b>Referencias</b>	
<b>Prioridad</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección “”</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
<i>Prototipo de Interfaz</i>	
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
<i>Prototipo de Interfaz</i>	
<b>Poscondiciones</b>	

## Anexo 15 Elicitación de los Requisitos de Software.

**Actividad:** Obtener información de la organización y del problema a resolver.

<b>Objetivos</b>	Recopilar información de la organización e identificar el problema a resolver.
<b>Responsables</b>	Analista Principal (Equipo de Desarrollo).
<b>Otros Participantes</b>	Líder de Software (Equipo de Desarrollo).  Analistas de Sistema (Equipo de Desarrollo).  Asegurador de los Requisitos de Software (Equipo de Desarrollo).  Especialista Funcional (Equipo de Desarrollo, Cliente).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Cronograma.</li> <li>• Documento Visión.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Minuta de Entrega.</li> <li>• Base de Documentación.</li> <li>• Lista de Riesgos.</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Definir la documentación que se debe solicitar (manuales de procedimientos y leyes).</li> <li>• Identificar las características del entorno de la organización.</li> <li>• Solicitar información de la organización que se requiere para el entendimiento del negocio.</li> <li>• Comprobar la totalidad de la información solicitada.</li> <li>• Realizar una minuta de entrega como constancia de la documentación recibida.</li> </ul>
<b>Técnicas a Utilizar</b>	Entrevistas.
<b>Estándar</b>	ITIL.

**Actividad:** Comprensión del Negocio.

<b>Objetivos</b>	Estudiar la información recopilada para comprender como funciona el negocio de la organización que va a ser automatizada, así como las personas que interactúan con el mismo.
<b>Responsable</b>	Analista Principal.
<b>Otros Participantes</b>	Analistas de Sistema (Equipo de Desarrollo).  Asegurador de los Requisitos de Software (Equipo de Desarrollo).  Arquitecto Principal (Equipo de Desarrollo).  Especialista Funcional (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Base de Documentación.</li> <li>• Guía de preguntas.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Glosario de Términos.</li> <li>• Guía de Preguntas (Actualizada).</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Revisar toda la documentación entregada por los clientes para un mejor entendimiento del negocio por ejemplo: leyes y manuales de procedimientos.</li> <li>• Realizar una guía de preguntas con la información que debe ser identificada en el estudio de la base de documentación.</li> </ul>
<b>Técnicas a Utilizar</b>	Arqueología de documento.
<b>Herramienta</b>	

**Actividad:** Realizar la modelación de los procesos del negocio.

<b>Objetivos</b>	Realizar la modelación de los procesos del negocio, a partir de la información identificada en la actividad anterior.
<b>Responsable</b>	Analista Principal.
<b>Otros Participantes</b>	Analistas de Sistema (Equipo de Desarrollo). Especialista Funcional (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Glosario de Términos.</li> <li>• Guía de Preguntas (Actualizada).</li> <li>• Base de documentación.</li> <li>• Documento visión.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Glosario de Términos (Actualizado).</li> <li>• Documento de los procesos del negocio.</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Realizar entrevistas para identificar la información, a partir de las necesidades no identificadas en la documentación estudiada.</li> <li>• Modelar los procesos del negocio.</li> <li>• Identificar reglas del negocio.</li> </ul>
<b>Técnicas a Utilizar</b>	Entrevistas y Arqueología de documento.
<b>Estándar</b>	ITIL.

**Actividad:** Revisión Técnica Formal.

<b>Objetivos</b>	Asegurar la calidad de los documentos realizados.
<b>Responsable</b>	Asegurador de los Requisitos de Software.
<b>Otros Participantes</b>	Analista Principal (Equipo de Desarrollo).  Analistas de Sistema (Equipo de Desarrollo).  Especialista Funcional (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Documento de los procesos del negocio.</li> <li>• Glosario de términos.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Documento de Revisión técnica formal.</li> <li>• Informe de No Conformidades.</li> <li>• Documento de los procesos del negocio (Actualizado).</li> <li>• Glosario de términos (Actualizado).</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Revisión de la documentación.</li> <li>• Revisión de estándares y formatos.</li> <li>• Revisión del contenido basado en listas de chequeos.</li> </ul>
<b>Técnicas a Utilizar</b>	Listas de Chequeo.
<b>Estándar</b>	COBIT

**Actividad:** Actualizar Cronograma

<b>Objetivos</b>	Actualizar las tareas planificadas en el cronograma.
<b>Responsable</b>	Líder de software (Equipo de Desarrollo).
<b>Otros Participantes</b>	Analista Principal (Equipo de Desarrollo). Analistas de Sistema (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"><li>• Cronograma.</li></ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"><li>• Cronograma (Actualizado).</li></ul>
<b>Tareas</b>	<ul style="list-style-type: none"><li>• Actualización del cronograma de acuerdo a los defectos encontrados.</li></ul>
<b>Técnicas a Utilizar</b>	
<b>Estándar</b>	

**Actividad:** Revisión y validación de los procesos del negocio por el cliente.

<b>Objetivos</b>	Revisar y validar la modelación de los procesos del negocio con el cliente.
<b>Responsable</b>	Gerente del proyecto (Cliente).
<b>Otros Participantes</b>	Líder de Software (Equipo de Desarrollo). Analista Principal (Equipo de Desarrollo). Grupo de Especialistas (Equipo de Desarrollo). Gerente del proyecto (Equipo de desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Documento de los procesos del negocio.</li> <li>• Glosario de términos.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Documento de los procesos del negocio (Validado).</li> <li>• Glosario de Términos (Validado).</li> <li>• Informe de No conformidades.</li> <li>• Minuta de aceptación.</li> <li>• Documento de revisión técnica formal.</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Revisión de la documentación por el cliente.</li> </ul>
<b>Técnicas a Utilizar</b>	Tormentas de ideas, Arqueología de Documentos.
<b>Estándar</b>	COBIT

**Actividad:** Identificar Requisitos Funcionales.

<b>Objetivos</b>	Identificar requisitos funcionales a partir de la modelación de los procesos del negocio.
<b>Responsable</b>	Analista Principal (Equipo de Desarrollo).
<b>Otros Participantes</b>	Analistas de Sistema (Equipo de Desarrollo).  Asegurador de los Requisitos de Software (Equipo de Desarrollo).  Especialista Funcional (Equipo de Desarrollo, Cliente).  Arquitecto Principal (Equipo de Desarrollo).  Diseñador de Base de Datos (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Documento de los procesos del negocio (Validado).</li> <li>• Glosario de Términos (Validado).</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Listado de Requisitos Funcionales.</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Identificar los posibles requisitos del sistema a partir de la modelación de los procesos del negocio.</li> <li>• Realizar entrevistas a los expertos de la parte cliente para identificar los posibles requisitos funcionales de la aplicación a desarrollar.</li> </ul>
<b>Técnicas a Utilizar</b>	Arqueología de Documento, Entrevistas.

## Anexo 16 Análisis de los Requisitos de Software.

**Actividad:** Análisis y negociación de los requisitos.

<b>Objetivos</b>	Analizar los requisitos identificados en la etapa de elicitación y negociar con el cliente nuevos requisitos.
<b>Responsable</b>	Analista Principal (Equipo de desarrollo).
<b>Otros Participantes</b>	Analista de sistemas (Equipo de Desarrollo). Asegurador de requisitos de software (Equipo de desarrollo). Líder de Software (Equipo de Desarrollo). Arquitecto Principal (Equipo de Desarrollo). Diseñador de Base de Datos (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Documento de los procesos del negocio.</li> <li>• Listado de Requisitos Funcionales.</li> <li>• Lista de riesgos.</li> <li>• Glosario de Términos.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Listado de Requisitos Funcionales (Actualizado).</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Analizar el listado de requisitos funcionales identificados en la etapa de elicitación.</li> <li>• Identificar posibles contradicciones entre los requisitos identificados.</li> <li>• Negociar con el cliente nuevos requisitos.</li> </ul>
<b>Técnicas a Utilizar</b>	Arqueología de Documento y Lluvia de Ideas.
<b>Estándar</b>	

**Actividad:** Revisión Técnica Formal.

<b>Objetivos</b>	Realizar una revisión interna del documento de Listado de Requisitos Funcionales para asegurar la calidad del mismo.
<b>Responsable</b>	Asegurador de los Requisitos de Software (Equipo de Desarrollo).
<b>Otros Participantes</b>	Analista Principal (Equipo de Desarrollo).  Analistas de Sistema (Equipo de Desarrollo).  Especialista Funcional (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>Listado de Requisitos Funcionales.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>Listado de Requisitos Funcionales (Actualizado).</li> <li>Informe de No Conformidades.</li> <li>Registro de Revisiones.</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>Realizar una revisión del Documento Listado de Requisitos Funcionales, aplicando listas de chequeo a la documentación.</li> <li>Identificar que cada uno de los requisitos cumplan con las siguientes características: <ul style="list-style-type: none"> <li>✓ Requisito consistente.</li> <li>✓ Requisito no ambiguo.</li> <li>✓ Requisito completo.</li> </ul> </li> <li>Actualización del listado de requisitos funcionales.</li> <li>Realizar un Informe de No Conformidades de los defectos encontrados.</li> </ul>
<b>Técnicas a Utilizar</b>	Listas de Chequeo, Revisión Formal.

<b>Estándar</b>	COBIT
<b>Modelo de Proceso</b>	CMMi.

**Actividad:** Actualizar Cronograma

<b>Objetivos</b>	Actualizar las tareas planificadas en el cronograma.
<b>Responsable</b>	Líder de software (Equipo de Desarrollo).
<b>Otros Participantes</b>	Analista Principal (Equipo de Desarrollo). Analistas de Sistema (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Cronograma.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Cronograma (Actualizado).</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Actualización del cronograma de acuerdo a los defectos encontrados.</li> </ul>
<b>Técnicas a Utilizar</b>	
<b>Estándar</b>	

**Actividad:** Priorizar los requisitos.

<b>Objetivos</b>	Priorizar los requisitos identificados de acuerdo a las necesidades de los usuarios.
<b>Responsable</b>	Analista Principal (Equipo de Desarrollo).
<b>Otros Participantes</b>	Analista de Sistemas (Equipo de Desarrollo). Asegurador de requisitos de software (Equipo de Desarrollo). Especialistas funcionales (Equipo de Desarrollo, Cliente). Líder de Software (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"><li>• Listado de Requisitos Funcionales (Actualizado).</li></ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"><li>• Listado de Requisitos Funcionales (Modificado).</li></ul>
<b>Tareas</b>	<ul style="list-style-type: none"><li>• Priorizar cada uno de los requisitos de acuerdo a las necesidades de los clientes en: Alto, Medio y Bajo.</li></ul>
<b>Técnicas a Utilizar</b>	Entrevistas, Lluvia de Ideas.

## Anexo 17 Especificación de los Requisitos de Software.

**Actividad:** Especificar los Requisitos de Software.

<b>Objetivos</b>	Especificar el Listado de los Requisitos Funcionales definidos en la etapa de Análisis.
<b>Responsable</b>	Analista Principal (Equipo de Desarrollo).
<b>Otros Participantes</b>	Analista de Sistema (Equipo de Desarrollo). Asegurador de los Requisitos de Software (Equipo de Desarrollo). Arquitecto Principal (Equipo de desarrollo). Diseñador de Base de Datos (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>Listado de los Requisitos Funcionales.</li> <li>Glosario de Términos.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>Documento de Especificación de los Requisitos de Software.</li> <li>Glosario de Términos (Actualizado).</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>Especificar los Requisitos de Software a partir del Listado de los Requisitos Funcionales.</li> </ul>
<b>Técnicas a Utilizar</b>	Arqueología de Documento. Entrevistas.
<b>Estándar</b>	IEEE 830

**Actividad:** Revisión Técnica Formal.

<b>Objetivos</b>	Realizar una revisión interna del documento de Especificación de los Requisitos de Software para asegurar la calidad del mismo.
<b>Responsable</b>	Asegurador de los Requisitos de Software (Equipo de Desarrollo).
<b>Otros Participantes</b>	Analista Principal (Equipo de Desarrollo).  Analistas de Sistema (Equipo de Desarrollo).  Especialista Funcional (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Documento de Especificación de los Requisitos de Software.</li> <li>• Glosario de Términos.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Documento de Especificación de los Requisitos de Software (Actualizado).</li> <li>• Glosario de Términos (Actualizado).</li> <li>• Informe de No Conformidades.</li> <li>• Registro de Revisiones.</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Realizar una revisión del Documento de Especificación de los Requisitos de Software, aplicando listas de chequeo a la documentación.</li> <li>• Realizar un Informe de No Conformidades de los defectos encontrados.</li> </ul>
<b>Técnicas a Utilizar</b>	Listas de Chequeo, Revisión Formal.
<b>Estándar</b>	COBIT.
<b>Modelo de Proceso</b>	CMMi.

**Actividad:** Actualizar Cronograma

<b>Objetivos</b>	Restablecer los tiempos de duración de las actividades tras un cambio en la planificación inicial.
<b>Responsable</b>	Líder de software (Equipo de Desarrollo).
<b>Otros Participantes</b>	Analista Principal (Equipo de Desarrollo).  Analistas de Sistema (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"><li>• Cronograma.</li></ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"><li>• Cronograma (Actualizado).</li></ul>
<b>Tareas</b>	<ul style="list-style-type: none"><li>• Adicionan y modifican las actividades, recursos o tiempos programados con anterioridad.</li><li>• Identificar las tareas que se tienen que realizar.</li><li>• Calcular los riesgos y retrasos de las actividades programadas con anterioridad.</li></ul> <p>Analizar las estrategias de mitigación de riesgos y retrasos.</p>
<b>Técnicas a Utilizar</b>	
<b>Estándar</b>	

**Actividad:** Revisión y validación del Documento Especificación de los Requisitos de Software con el Cliente.

<b>Objetivos</b>	Revisar y validar el Documento de Especificación de los Requisitos de Software con el cliente.
<b>Responsable</b>	Gerente del proyecto (Cliente).
<b>Otros Participantes</b>	Líder de Software (Equipo de Desarrollo).  Analista Principal (Equipo de Desarrollo).  Asegurador de los Requisitos de Software (Equipo de Desarrollo).  Grupo de Especialistas (Equipo de Desarrollo).  Gerente del proyecto (Equipo de desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Documento de Especificación de los Requisitos de Software.</li> <li>• Glosario de Términos.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Minuta de Aceptación.</li> <li>• Documento de Especificación de los Requisitos de Software (Aprobado).</li> <li>• Glosario de Terminos (Validado).</li> <li>• Informe de No Conformidades.</li> <li>• Registro de Revisiones.</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Exponer la interpretación de los Requisitos de Software encontrados.</li> <li>• Realizar un encuentro con el Cliente donde se valide la Especificación de los Requisitos de Software.</li> <li>• Realizar un Informe de No Conformidades de los defectos encontrados.</li> </ul>
<b>Técnicas a Utilizar</b>	Arqueología de Documento, Revisión Formal.
<b>Estándar</b>	

**Actividad:** Realizar el Modelo de Caso de Uso del Sistema.

<b>Objetivos</b>	Modelar los Casos de Uso del Sistema a partir del Documento de Especificación de los Requisitos de Software.
<b>Responsable</b>	Analista Principal (Equipo de Desarrollo).
<b>Otros Participantes</b>	Analista de Sistema (Equipo de Desarrollo).  Asegurador de los Requisitos de Software (Equipo de Desarrollo).  Especialista Funcional (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Documento de Especificación de los Requisitos de Software.</li> <li>• Glosario de Términos.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Modelo de Caso de Uso del Sistema.</li> <li>• Glosario de Términos (Actualizado).</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Realizar la modelación de los Casos de Uso del Sistema a partir del Documento de Especificación de Requisitos de Software.</li> <li>• Realizar los Prototipos no Funcionales.</li> </ul>
<b>Técnicas a Utilizar</b>	Listas de Chequeo, Caso de Uso, Prototipo.
<b>Estándar</b>	

**Actividad:** Revisión Formal Interna.

<b>Objetivos</b>	Realizar una revisión interna del Modelo de Caso de Uso del Sistema para asegurar la calidad de los documentos que serán entregados al Cliente.
<b>Responsable</b>	Asegurador de los Requisitos de Software (Equipo de Desarrollo).
<b>Otros Participantes</b>	Analista Principal (Equipo de Desarrollo). Especialista Funcional (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Modelo de Caso de Uso del Sistema.</li> <li>• Glosario de Términos.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Modelo de Caso de Uso del Sistema (Actualizado).</li> <li>• Glosario de Términos (Actualizado).</li> <li>• Informe de No Conformidades.</li> <li>• Registro de Revisiones.</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Realizar una revisión del Modelo de Caso de Uso del Sistema.</li> <li>• Aplicar la Listas de Chequeo de Modelo de Caso de Uso del Sistema a la documentación.</li> <li>• Realizar un Informe de No Conformidades de los defectos encontrados.</li> </ul>
<b>Técnicas a Utilizar</b>	Listas de Chequeo, Revisión Formal.
<b>Estándar</b>	COBIT
<b>Modelo de Proceso</b>	CMMi.

**Actividad:** Actualizar Cronograma

<b>Objetivos</b>	Actualizar las tereas planificadas en el cronograma de la Modelación de los Casos de Uso.
<b>Responsable</b>	Líder de software (Equipo de Desarrollo).
<b>Otros Participantes</b>	Analista Principal (Equipo de Desarrollo). Analistas de Sistema (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"><li>• Cronograma.</li></ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"><li>• Cronograma (Actualizado).</li></ul>
<b>Tareas</b>	<ul style="list-style-type: none"><li>• Actualización del cronograma de acuerdo a los defectos encontrados.</li></ul>
<b>Técnicas a Utilizar</b>	
<b>Estándar</b>	

**Actividad:** Revisión y validación del Documento de Modelo de Caso de Uso del Sistema con el Cliente.

<b>Objetivos</b>	Revisar y validar el Documento de Modelo de Caso de Uso del Sistema con el cliente.
<b>Responsable</b>	Gerente del proyecto (Cliente).
<b>Otros Participantes</b>	Líder de Software (Equipo de Desarrollo). Analista Principal (Equipo de Desarrollo). Asegurador de los Requisitos de Software (Equipo de Desarrollo). Grupo de Especialistas (Equipo de Desarrollo). Gerente del proyecto (Equipo de desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Modelo de Caso de Uso del Sistema.</li> <li>• Glosario de Términos.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Minuta de Aceptación.</li> <li>• Modelo de Caso de Uso del Sistema (Aprobado).</li> <li>• Glosario de Términos (Aprobado).</li> <li>• Informe de No Conformidades.</li> <li>• Registro de Revisiones.</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Exponer la interpretación de los Casos de Usos encontrados.</li> <li>• Realizar un encuentro con el Cliente donde se valide el Modelo de Caso de Uso del Sistema.</li> <li>• Realizar un Informe de No Conformidades de los defectos encontrados.</li> </ul>
<b>Técnicas a Utilizar</b>	Arqueología de Documento, Revisión Formal.
<b>Estándar</b>	
<b>Modelo de Proceso</b>	

## Anexo 18 Validación de los Requisitos de Software.

**Actividad:** Validación de la Especificación de los Requisitos de Software.

<b>Objetivos</b>	El objetivo es que el Cliente valide la Especificación de los Requisitos de Software realizados por el Analista de Sistemas del Equipo de Desarrollo.
<b>Responsable</b>	Gerente de Proyecto (Cliente).
<b>Otros Participantes</b>	Líder de Software (Equipo de Desarrollo). Analista Principal (Equipo de Desarrollo). Especialistas Funcionales (Equipo de Desarrollo, Cliente). Asegurador de los Requisitos de Software (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Documento de Especificación de los Requisitos de Software.</li> <li>• Glosario de Términos.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Documento de Especificación de los Requisitos de Software (Aprobado).</li> <li>• Glosario de Términos (Aprobado).</li> <li>• Minuta de Aceptación.</li> <li>• Registro de Revisiones.</li> <li>• Informe de No Conformidades.</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Realizar una validación en conjunto con el Cliente de la Especificación de los Requisitos de Software.</li> </ul>
<b>Técnicas a Utilizar</b>	Entrevista.

**Actividad:** Validación del Modelo de Caso de Uso del Sistema.

<b>Objetivos</b>	El objetivo es que el Cliente valide el Modelo de Caso de Uso de Sistema realizado por el Analista de Sistemas del Equipo de Desarrollo.
<b>Responsable</b>	Gerente de Proyecto (Cliente).
<b>Otros Participantes</b>	Líder de Software (Equipo de Desarrollo).  Analista Principal (Equipo de Desarrollo).  Especialistas Funcionales (Equipo de Desarrollo, Cliente).  Asegurador de los Requisitos de Software (Equipo de Desarrollo).
<b>Artefactos de Entrada</b>	<ul style="list-style-type: none"> <li>• Modelo de Caso de Uso del Sistema.</li> <li>• Glosario de Términos.</li> </ul>
<b>Artefactos de Salida</b>	<ul style="list-style-type: none"> <li>• Modelo de Caso de Uso del Sistema (Aprobado).</li> <li>• Glosario de Términos (Aprobado).</li> <li>• Minuta de Aceptación.</li> <li>• Registro de Revisiones.</li> <li>• Informe de No Conformidades.</li> </ul>
<b>Tareas</b>	<ul style="list-style-type: none"> <li>• Realizar una validación en conjunto con el Cliente del Modelo de Caso de Uso del Sistema.</li> </ul>
<b>Técnicas a Utilizar</b>	Entrevistas.

## Anexo 19 Cuestionario para la evaluación de la propuesta

### **Criterio de Evaluación de la propuesta:**

Propuesta para el levantamiento de requisitos en función del aseguramiento de la calidad de software.

### **Cuestionario aplicable a los especialistas seleccionados.**

**Objetivos del cuestionario:** Evaluación de la propuesta mediante el Criterio de Especialista.

Es necesaria la lectura y análisis de la propuesta para dar respuesta al cuestionario.

### **Datos del Especialista**

*Nombre y Apellidos:* \_\_\_\_\_

*Cargo:* \_\_\_\_\_

*Categoría Científica:* \_\_\_\_\_

*Vinculación a proyecto:* \_\_\_\_\_

*Roles:* \_\_\_\_\_

*Años de experiencia en el desarrollo de proyectos de software exitosos:* \_\_\_\_\_

*Años de experiencia en la dirección de proyectos de software exitosos:* \_\_\_\_\_

---

1. Preguntas sobre la propuesta.

- a. Considera esta propuesta:
- i. completa \_\_\_\_
  - ii. incompleta \_\_\_\_
- b. ¿Considera que las fases de la Ingeniería de Requisitos que se contemplan en la propuesta, son suficientes para desarrollar un Levantamiento de Requisitos en el que se asegure la calidad?
- i. Si \_\_\_\_ No \_\_\_\_ ¿Cuál o Cuáles considera que se debe incluir? \_\_\_\_
- c. ¿Son suficientes las revisiones de calidad que se realizan durante las etapas de elicitación, análisis, especificación y validación contempladas dentro de cada fase de la Ingeniería de Requisitos?
- i. Si \_\_\_\_ No \_\_\_\_
- d. ¿Logrará la correcta planificación y gestión de riesgos durante las etapas de elicitación, análisis, especificación y validación?
- i. Si \_\_\_\_ No \_\_\_\_
- e. ¿Están correctamente identificados los roles por la parte Cliente?
- i. Si \_\_\_\_ No \_\_\_\_ ¿Cuál o Cuáles considera que se debe incluir? \_\_\_\_
- f. ¿Están correctamente identificados los roles por la parte del Equipo de Desarrollo?
- i. Si \_\_\_\_ No \_\_\_\_ ¿Cuál o Cuáles considera que se debe incluir? \_\_\_\_
- g. ¿Cree necesario la utilización de los estándares de calidad ITIL y COBIT?
- i. Si \_\_\_\_ No \_\_\_\_ ¿Cuál o Cuáles considera que se debe incluir? \_\_\_\_\_
- h. ¿Cree necesarias la utilización de las técnicas: Entrevistas y Cuestionarios, Lluvia de Ideas, Prototipos, Caso de Uso, Arqueología de Documento para el Levantamiento de Requisitos?
- i. Si \_\_\_\_ No \_\_\_\_ ¿Cuál o Cuáles considera que se debe incluir? \_\_\_\_\_

i. ¿Cree necesarias las herramientas de Listas de Chequeo a utilizar para cada una de las etapas del Levantamiento de Requisitos, para asegurar la calidad?

i. Si \_\_\_ No \_\_\_ ¿Cuál o Cuáles considera que se debe incluir? \_\_\_\_\_

j. ¿Cree necesario la elaboración de un plan de revisiones durante la etapa de levantamiento de requisitos?

i. Si \_\_\_ No \_\_\_

k. ¿Detecta algo incorrecto en la propuesta?

i. Si \_\_\_ No \_\_\_ ¿Qué? \_\_\_\_\_

l. ¿Cree que la propuesta es tan compleja como para que no se use?

i. Si \_\_\_ No \_\_\_

2. Marque cuál o cuáles roles usted cree que deban estar presentes en un Levantamiento de Requisitos y considera que son más eficiente.

\_\_\_ Gerente de Proyecto

\_\_\_ Jefe de Proyecto

\_\_\_ Arquitecto

\_\_\_ Programador

\_\_\_ Analista

\_\_\_ Diseñador

\_\_\_ Diseñador de bases de datos

\_\_\_ Arquitecto de Información

\_\_\_ Probador

\_\_\_ Diseñador gráfico

\_\_\_ Gestor de configuración y cambios

Asegurador de los Requisitos de Software

Especialista de seguridad

Especialistas Funcionales

Documentador

Marque cuáles roles prefiere evitar

Gerente de Proyecto

Jefe de Proyecto

Arquitecto

Programador

Analista

Diseñador

Diseñador de bases de datos

Arquitecto de Información

Probador

Diseñador gráfico

Gestor de configuración y cambios

Asegurador de los Requisitos de Software

Especialista de seguridad

Especialistas Funcionales

Documentador

3. Emita brevemente su opinión sobre la propuesta de forma general.

4. Evaluación cuantitativa de la propuesta:

<b>Criterio de evaluaciones</b>	<b>Valores</b>
Alta	5 puntos
Media	4 puntos
Baja	3 puntos
Ninguna	2 puntos

<b>Aspectos a evaluar</b>	<b>Eval.</b>
Importancia y necesidad de la aplicación de la propuesta.	
Adaptabilidad a los proyectos que utilicen una metodología diferente a RUP.	
Posibilidades de aplicación.	
Eficacia de la propuesta.	
Nivel de complejidad de la propuesta.	
Grado de completitud de la propuesta.	
Orden lógico y estructura de las actividades.	
Satisfacción de las necesidades de los proyectos.	

### GLOSARIO DE TÉRMINOS

#### A

**Artefactos:** Una parte de la información que es producida, modificada, o usada por un proceso, define un área de responsabilidad, y está sujeta al control de versión. Un artefacto puede ser un modelo, un elemento del modelo, o un documento. Un documento puede adjuntar otros documentos.

**Arqueología de documentos:** Es el estudio de todos los documentos que debe tener la organización desde su inicio, es la revisión detallada de los documentos de la organización que se desea informatizar.

#### C

**Captura de Requisitos (CR):** Proceso durante el cual se identifica un problema y se especifica los requisitos que debe cumplir un producto de software.

**Caso de Uso:** Notación utilizada para representar los requerimientos funcionales de un sistema basada en el esquema propuesto por Ivar Jacobson a principios de la década del '90.

**Cliente:** Una persona u organización, interna o externa a la organización productora, que toma responsabilidad financiera por el sistema. El cliente es el último destinatario del producto desarrollado y sus artefactos. Beneficiado con el producto o servicio.

**CMM:** *Capacity Maturity Model fue desarrollado por el Software Engineering Institute (SEI), su principal premisa es la calidad de un producto determinada por la calidad del proceso utilizado para desarrollarlo y mantenerlo. Es un modelo de referencias sobre buenas prácticas, consolidadas y probadas para el desarrollo y mantenimiento de productos y servicios, cubriendo todo el ciclo de vida, desde la concepción a la entrega y mantenimiento. Integra la Ingeniería de Software, la Ingeniería de Sistemas y la Adquisición de Productos y Servicios.*

**Contrato:** Es un documento legalmente obligatorio y en el estarán de acuerdo las partes del cliente y proveedor. Esto incluye los requisitos técnicos y requerimientos de la organización, costo y tiempo para un producto. Un contrato también puede contener la información informal pero útil como los compromisos o expectativas de las partes involucradas.

#### D

**Defecto:** Cualquier requerimiento, elemento de diseño o de implementación que si no es cambiado, causará un diseño, implementación, prueba, uso, o mantenimiento inapropiado del producto.

### E

**Entrevista:** Acción y efecto de entrevistar o entrevistarse. Vista, concurrencia y conferencia de dos o más personas en lugar determinado, para tratar o resolver un negocio.

**Equipo:** Equipo de software. Dos personas o más que trabajan para lograr una meta, objetivo o misión común, donde cada individuo tiene asignado un rol específico y donde el completamiento de la misión depende de los miembros del equipo.

**Especificaciones de Requisitos:** Documento que describe lo que hace un sistema de software: sus funciones y sus atributos. Generalmente escritas desde el punto de vista del usuario.

### F

**Funcionalidad:** (Functionality, ISO 9126): grado en que las necesidades asumidas o descritas se satisfacen. Se divide en las subcaracterísticas idoneidad, precisión, interoperabilidad, seguridad.

### I

**IEEE:** (Institute of Electrical and Electronics Engineers) Asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, científicos de la computación e ingenieros en telecomunicación. Se creó en 1884.

**Ingeniería de Software:** La Ingeniería de Software es una tecnología multicapa en la que, según Pressman, se pueden identificar: los métodos, el proceso y las herramientas.

**Ingeniería de Requerimiento (IR):** Se define, como un conjunto de actividades en las cuales, utilizando técnicas y herramientas, se analiza un problema y se concluye con la especificación de una solución (a veces más de una).

**ISO:** (International Organization for Standardization) Es la Organización Internacional para la Normalización; responsable para la normalización a escala mundial. ISO está formado por distintos comités técnicos, cada uno de los cuales es responsable de la normalización para cada área de especialidad. El propósito de ISO es promover el desarrollo de la normalización para fomentar a nivel internacional el intercambio de bienes y servicios y para el desarrollo de la cooperación en actividades económicas, intelectuales, científicas y tecnológicas. El resultado del trabajo técnico dentro de ISO se publica en forma final como normas internacionales.

### L

**Lista de Chequeo:** Se entiende por lista de chequeo (cheks-list) a un listado de preguntas, en forma de cuestionario que sirve para verificar el grado de cumplimiento de determinadas reglas establecidas

aprioris con un fin determinado. El uso de estas listas está generalizado en rubros muy diversos que van desde verificar y determinar el potencial de mercados extranjeros hasta medir la confiabilidad y seguridad de sistemas informáticos, incluyendo ítems tales como la evaluación de criterios de usabilidad de un sitio de Internet, como así también la verificación de un plan.

### M

**Madurez:** (*Maturity*, ISO 9126) Subcaracterística de fiabilidad, que indica la frecuencia con que ocurren los fallos.

**Mantenimiento:** (*Maintainability*, ISO 9126) Esfuerzo requerido para implementar cambios. Se divide en las subcaracterísticas capacidad de ser analizado, confiabilidad, estabilidad, facilidad de prueba.

**Proceso:** (ISO-15504) Proceso de Software, es el proceso (o procesos), usado por una organización (o proyecto), para planificar, administrar, ejecutar, monitorear, controlar y mejorar sus actividades, relacionadas con el software.

### N

**No Conformidad:** Según la definición de ISO 9000: 2000 (3.6.2), una no conformidad es el “incumplimiento de un requisito”.

### O

**Observación:** Es la acción de mirar detenidamente una cosa para asimilar en detalle la naturaleza investigada, su conjunto de datos, hechos y fenómenos.

**Organización:** La organización es una unidad social coordinada, consciente, compuesta por dos personas o más, que funciona con relativa constancia a efecto de alcanzar una meta o una serie de metas comunes. Según esta definición, las empresas productoras, los organismos de los gobiernos locales, estatales y federal son organizaciones.

### P

**Proceso de Desarrollo de Software:** Un Proceso de Desarrollo de Software es la definición del conjunto de actividades que guían los esfuerzos de las personas implicadas en el proyecto, a modo de plantilla que explica los pasos necesarios para terminar el proyecto. Un PROCESO define:

“QUIEN”, “QUE”, “CUANDO”, y “COMO” hay que realizar las cosas para alcanzar un determinado producto de software.

**Prototipo:** Es un modelo a escala o imitación de lo real, pero no tan funcional para que equivalga a un producto final, ya que no lleva a cabo la totalidad de las funciones necesarias del sistema final. Proporcionando una retroalimentación temprana por parte de los usuarios acerca del Sistema.

### R

**Revisión:** Reuniones de un grupo definido de personas cuyo objetivo es encontrar errores en un artefacto de software. Con revisiones para testear requisitos, diseño, planes, manuales y software. Participantes de las revisiones son: los autores que han escrito el artefacto; los revisores que tienen que detectar errores; el secretario que documenta los errores encontrados; el presentador que expone/explica el artefacto bajo testeo; el líder que dirige la reunión, elige la fecha para la reunión e invita a los participantes.

**Requerimiento:** Una condición o capacidad necesaria para un usuario para resolver un problema o alcanzar un objetivo.

**Requerimiento funcional:** Alguna actividad o servicio que debe realizar el software y que puede ser comprobada.

**Rol:** Papel, cometido o función que tiene o desempeña un actor.

**RUP:** (Racional Unified Process) El Proceso Unificado de Desarrollo Software o simplemente Proceso Unificado es un marco de desarrollo software iterativo e incremental. El refinamiento más conocido y documentado del Proceso Unificado es el Proceso Unificado de Rational o simplemente RUP.

### S

**Seguridad:** (Security, ISO 9126) Subcaracterística de funcionalidad, que indica el grado en que un acceso no autorizado (accidental o deliberado) se prevenga y se permita un acceso autorizado.

### T

**Tormenta de ideas:** Es una herramienta de planeamiento que se puede utilizar para obtener ideas a partir de la creatividad de un grupo y con ello resolver un problema. El fundamento del brainstorming es la generación de ideas, en modo individual o en grupo, evitando evaluaciones inmediatas: la investigación científica ha demostrado que este principio es altamente productivo tanto en el esfuerzo individual como en el trabajo de grupo.

### U

**UCI:** Universidad de las Ciencias Informáticas.

**Usuario:** Aquella persona o grupo de personas dentro de la empresa que utilizarán el software desarrollado, incluido el tomador de decisiones.

**Usuario Final:** Toda aquella persona u organización siendo influenciada o ejerciendo influencia sobre el software que está siendo construido.

### X

**XP:** Extreme Programming (Programación Extrema), es la más destacada de los procesos ágiles de desarrollo de software. Utilizada para proyectos de corto plazo. Consisten desarrollos rápidos e iterativos, cuya particularidad es tener como parte del equipo al usuario final.