

Universidad de las Ciencias Informáticas



Trabajo para optar por el título de Máster
en Gestión de Proyectos Informáticos

Título: Técnicas y herramientas de la
ingeniería de requisitos adecuadas para
simuladores virtuales.

Autora: Lic. Yoisy Pérez Olmos

Tutoras: Msc. Karina Pérez Teruel

Dra. Ailyn Febles Estrada

Ciudad de la Habana, Noviembre 2009

A mis hijos aunque no hayan nacido.

A mis padres y a mi esposo por su desmedido apoyo.

*A mi amigo Gerandys por su colaboración en las asignaturas y su
infinita generosidad.*

A Juan por formar parte de este trabajo y su dedicación a mí.

A mis tutoras por el apoyo y las recomendaciones.

A la amiguita Dania por su apoyo y optimismo.

A Lennita por ser compañera de estudio y su especial amistad.

A Maypher y Michael por colaborar en las revisiones de este trabajo

A Pepe por su contribución a la traducción del resumen.

A Nela por sus gestiones para la tutoría de este trabajo.

A Yusleidy por sus sugerencias y recomendaciones.

A Maydelis por dedicarme parte de su tiempo sin apenas conocerme.

Al comité de expertos por su apoyo y recomendaciones.

Abstract

The Virtual Reality Systems are used now days in several fields of industry and are even able to represent real processes in a computer. In order to reach the biggest possible realism in these applications the Virtual Simulators are specifically developed. One of the main difficulties that the development process of these systems faces today is that the work team doesn't have an exact idea of what is needed to build. The discipline of Requirements Engineering, dedicated to the study of the software requirements, is one of the proposals of the Software Engineering to improve these processes.

In this investigation, the author proposes techniques and tools to carry out the Requirements Engineering in virtual simulation projects developed in the Virtual Reality Center of the University of Computer Sciences. There were defined techniques such as that of the Apprentice to obtain the requirements that will be specified by means of the Use cases and Scenarios combined with Storyboards. The Prototypes are recommended to validate the obtained specifications also using the Check lists. The proposal also includes a group of tools to support the application of these techniques including: CASE tools, Graphic engines and RequisitePro included to achieve the automatization of the requirements management that will be executed by means of traceability matrix. The proposal was validated by experts in both areas of the knowledge, which coincided in the importance of its application. As a complement to the theoretical validation are also presented the results of applying the proposal in one of the simulators developed in the Center of Virtual Reality.

Key words: requirements engineering techniques, requirements engineering tools, virtual simulators.

Resumen

Los Sistemas de Realidad Virtual son utilizados en la actualidad en varias áreas de la industria llegando a representar procesos reales en una computadora. En aras de alcanzar el mayor realismo posible en estas aplicaciones se desarrollan específicamente los Simuladores Virtuales. Una de las principales dificultades que enfrenta hoy el proceso de desarrollo de estos sistemas radica en que el equipo de trabajo no tiene una idea exacta de lo que se necesita construir. La disciplina Ingeniería de Requisitos, dedicada al estudio de los requisitos de software, constituye una de las propuestas de la Ingeniería de Software al mejoramiento de estos procesos.

En el presente trabajo se define una propuesta de técnicas y herramientas para llevar a cabo la Ingeniería de Requisitos en proyectos de simulación virtual desarrollados en el Polo de Realidad Virtual de la Universidad de Ciencias Informáticas. En este sentido se definieron técnicas como la del Aprendiz para obtener los requisitos que serán especificados mediante los Casos de uso y Escenarios combinados con Storyboards. Los prototipos son recomendados para validar las especificaciones obtenidas utilizando además, las listas de chequeo. Se propone también un conjunto de herramientas para soportar el empleo de estas técnicas que incluye: herramientas CASE, motores gráficos y RequisitePro incluida esta última para automatizar la gestión de los requisitos que se ejecutará haciendo uso de matrices de trazabilidad. La propuesta fue validada por expertos en ambas áreas del conocimiento, los cuales coincidieron en la importancia de su aplicación. Como complemento a la validación teórica se presentan también los resultados de aplicar la propuesta en uno de los simuladores que se desarrolla en el Polo de Realidad Virtual.

Palabras claves: técnicas de la ingeniería de requisitos, herramientas de la ingeniería de requisitos, simuladores virtuales.

INDICE

INTRODUCCIÓN..... 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA INGENIERÍA DE REQUISITOS EN SIMULADORES VIRTUALES..... 8

1.1 ESTADO DEL ARTE DE LOS SIMULADORES VIRTUALES 8

 1.1.1 *Desarrollo de Simuladores Virtuales en Cuba* 9

1.2 REQUISITOS DE SOFTWARE 11

1.3 IMPORTANCIA DE LOS REQUISITOS EN LA INDUSTRIA DEL SOFTWARE 12

1.4 LA INGENIERÍA DE REQUISITOS 15

1.5 LA INGENIERÍA DE REQUISITOS SEGÚN ESTÁNDARES DE CALIDAD..... 16

 1.5.1 *Ingeniería de requisitos en CMMI* 17

 1.5.2 *Ingeniería de requisitos en las prácticas recomendadas por la IEEE* 19

 1.5.3 *Ingeniería de requisitos en la ISO* 20

1.6 PROCEDIMIENTOS USADOS EN IR 20

1.7 TÉCNICAS UTILIZADAS EN INGENIERÍA DE REQUISITOS..... 22

 1.7.1 *Técnicas para la obtención de requisitos*..... 22

 1.7.2 *Técnicas para el análisis de requisitos*..... 25

 1.7.3 *Técnicas para la especificación de requisitos* 27

 1.7.4 *Técnicas para la validación de requisitos* 29

 1.7.5 *Técnicas para la gestión de requisitos* 30

1.8 HERRAMIENTAS UTILIZADAS EN INGENIERÍA DE REQUISITOS 32

 1.8.1 *Herramientas para el desarrollo de requisitos* 32

 1.8.2 *Herramientas para la gestión de requisitos*..... 33

1.9 CALIDAD DE LAS ESPECIFICACIONES DE REQUISITOS 35

1.10 LA IR EN SIMULADORES VIRTUALES 37

CONSIDERACIONES DEL CAPÍTULO 40

CAPÍTULO 2: TÉCNICAS Y HERRAMIENTAS DE LA IR PARA SV.....41

2.1 CARACTERÍSTICAS DE LOS SIMULADORES VIRTUALES..... 41

2.2 PROPUESTA DE PROCESOS O ETAPAS PARA LA IR EN SV..... 43

2.3 TÉCNICAS EN CADA ETAPA O PROCESO DE LA INGENIERÍA DE REQUISITOS 45

2.4 TÉCNICAS PARA LA OBTENCIÓN DE REQUISITOS..... 47

 2.4.1 *Entrevista*..... 48

 2.4.2 *Tormenta de ideas* 49

 2.4.3 *Aprendiz*..... 50

2.5 TÉCNICAS PARA EL ANÁLISIS DE REQUISITOS 50

 2.5.1 *Priorización de los requisitos mediante el método AHP* 51

2.6 TÉCNICAS PARA LA ESPECIFICACIÓN DE REQUISITOS 56

 2.6.1 *Casos de uso* 57

 2.6.2 *Storyboards*..... 58

 2.6.3 *Escenarios* 59

2.7	TÉCNICAS PARA LA VALIDACIÓN DE REQUISITOS	60
2.7.1	<i>Construcción de prototipos</i>	61
2.7.2	<i>Listas de chequeo</i>	62
2.7.3	<i>Generación de casos de prueba</i>	63
2.8	TÉCNICAS PARA LA GESTIÓN DE REQUISITOS	64
2.8.1	<i>Trazabilidad de requisitos</i>	64
2.9	HERRAMIENTAS PROPUESTAS PARA LA IR EN SV.....	67
2.9.1	<i>Herramientas para el desarrollo de requisitos</i>	70
2.9.2	<i>Herramientas para la gestión de requisitos</i>	72
	CONSIDERACIONES DEL CAPÍTULO	73
CAPÍTULO 3: VALIDACIÓN Y APLICACIÓN PRÁCTICA DE LA PROPUESTA.....		74
3.1	VALIDACIÓN DEL MODELO PROPUESTO POR UN MÉTODO TEÓRICO.....	74
1.1.1	<i>Selección de los expertos</i>	74
1.1.2	<i>Determinación de la validez de la propuesta</i>	76
3.2	APLICACIÓN PRÁCTICA DE LA PROPUESTA EN EL PROYECTO SIMULADOR QUIRÚRGICO.	79
3.2.1	<i>Estructura del proyecto Simulador Quirúrgico</i>	79
3.2.2	<i>Situación inicial del simulador quirúrgico</i>	81
3.2.3	<i>Técnicas utilizadas en el simulador quirúrgico</i>	82
3.2.4	<i>Resultados arrojados en el proyecto simulador quirúrgico</i>	89
	CONSIDERACIONES DEL CAPÍTULO	93
CONCLUSIONES		94
RECOMENDACIONES		95
REFERENCIAS BIBLIOGRÁFICAS		96
ANEXOS		1
	<i>Anexo 1: Lista de chequeo de especificaciones de requisitos</i>	1
	<i>Anexo 2: Encuesta realizada a líderes de proyectos de simulación virtual del país</i>	4
	<i>Anexo 3: Propuesta de plantilla para la realización de la primera entrevista con el cliente</i>	6
	<i>Anexo 4: Propuesta de plantilla para la descripción textual de un caso de uso</i>	7
	<i>Anexo 5: Propuesta de plantilla para la descripción textual de un escenario</i>	8
	<i>Anexo 6: Propuesta de plantilla para la generación de un caso de prueba</i>	9
	<i>Anexo 7: Primera encuesta realizada a los Expertos para obtener su nivel de competencia</i>	10
	<i>Anexo 8: Segunda encuesta realizada a los Expertos para evaluar el valor práctico de la aplicación de la propuesta y el cumplimiento de los objetivos propuestos</i>	12
	<i>Anexo 9: Encuesta realizada al equipo de desarrollo del proyecto simulador quirúrgico para valorar la calidad de la especificación de requisitos realizada haciendo uso de las técnicas y herramientas propuestas</i>	14

Introducción

En la actualidad la expansión de las nuevas tecnologías de la información y las comunicaciones ha hecho necesario que se desarrolle, a grandes escalas, la rama de la informática. La informática gráfica ha venido destacándose, producto a la sensación virtual de la realidad que proporciona al hombre, durante las dos últimas décadas como la rama más sensacional de la informática. Un concepto a destacar en la informática gráfica es la “Realidad Virtual” (RV), donde se logran efectos de interactividad y realismo, alcanzando cambios en la forma de percibir el tiempo, el espacio e incluso la propia identidad.

El término Realidad Virtual se relaciona al tratamiento de ambientes tridimensionales e interactivos orientados a la visualización de objetos 3D generados por computadoras, mostrando situaciones del mundo real en mundos virtuales (1). Engloba herramientas que permiten visualizar un objeto tridimensional, o un grupo de objetos tridimensionales, que en conjunto forman el entorno virtual; de modo que el usuario puede manipular mediante algún tipo de dispositivo una o varias características de los objetos y esto se visualice de forma interactiva y en tiempo real, creando la ilusión de que se forma parte del entorno. El empleo de estos Sistemas de Realidad Virtual (SRV) posibilita que los cirujanos puedan realizar operaciones simuladas para ensayar las técnicas más complicadas antes de una operación real, los economistas exploran un modelo de acción de un sistema económico para poder entender mejor las complejas relaciones existentes entre sus distintos componentes, los astronautas tienen la posibilidad de volar sobre la superficie simulada de un planeta desconocido y experimentar la sensación que tendrían si estuvieran allí, los arquitectos pueden hacer que sus clientes, enfundados en cascos y guantes, visiten los pisos-piloto en un mundo virtual dándoles la oportunidad de que abran las puertas o las ventanas y enciendan o apaguen las luces del apartamento; en el ámbito científico, investigadores estudian moléculas complejas, desplazando grupos de átomos mediante un instrumento (2).

Para lograr todas estas aplicaciones con el mayor realismo posible se desarrollan los Simuladores Virtuales (SV). Un SV puede usar cualquier combinación de sonido, vista, movimiento y olor para hacer sentir las experiencias de una situación real (3). Es la representación, a partir de modelos matemáticos, de un proceso real en una computadora. Estos modelos muestran una realidad en la cual la complejidad de los eventos o situaciones pueden controlarse. Una de las características principales de la

simulación son sus herramientas audiovisuales e interactivas, como por ejemplo el diseño gráfico de las imágenes tridimensionales de personas y de espacios o lugares, que dan la sensación de que se está en un escenario real (4).

Ingeniería de requisitos. Enfoque para Simuladores Virtuales.

El proceso de desarrollo de los Simuladores Virtuales (SV) en general, presenta carencias metodológicas propias de una industria tan reciente como la del software. Una de las principales dificultades radica en que el equipo de desarrollo no tiene una idea clara de lo que se necesita construir, de lo que realmente desea el cliente, afectándose así la calidad del producto final.

Evidentemente, existen una serie de dificultades que todos los proyectos de desarrollo de software presentan en la etapa de definición de lo que el sistema debe hacer. Por ejemplo el hecho de que los propios usuarios no tienen una idea exacta de lo que se necesita, de qué partes de su trabajo pueden transformarse en software. La solución primaria a este problema en la industria de software, ha sido definir intermediarios (analistas) con el objetivo de obtener los requisitos de cada usuario y de esta forma lograr una única visión por parte, tanto del equipo de desarrollo, como de los usuarios y componer una especificación de requisitos completa, correcta y consistente (5).

La industria, en aras del mejoramiento de estos procesos, ha ido consolidando una nueva rama de la Ingeniería de Software (IS) que se ocupa de la investigación-desarrollo en el campo de los requisitos de software, rama denominada Ingeniería de Requisitos (IR). Reconocida desde 1990 y hasta la fecha, variadas son las técnicas y herramientas que se han desarrollado y que han permitido aplicar esta disciplina no sólo al ámbito del desarrollo de software, sino también a otras esferas (6).

La IR proporciona el mecanismo apropiado para entender lo que el cliente quiere, analizar las necesidades, evaluar la factibilidad, negociar una solución razonable, especificar la solución sin ambigüedades, validar la especificación, y administrar los requisitos conforme estos se convierten en un sistema operacional (7).

La captura de los requisitos sigue siendo difícil y la IR establece procesos útiles y sistemáticos para llevarla a cabo, de manera que guía el desarrollo hacia el sistema correcto (5). Su importancia se sustenta al permitir una definición clara, consistente y compacta de las correctas especificaciones de requisitos que definen el comportamiento del sistema con el fin de minimizar al máximo los problemas presentados en el desarrollo del software y que tanto afectan al producto final (6).

Estas especificaciones de requisitos son tratadas en (8) como el producto de trabajo resultante sobre los requisitos, considerado en la fase inicial del desarrollo del sistema y refinado de manera incremental en las disciplinas subsiguientes. Teniendo en cuenta los criterios tratados en (9), en el estándar publicado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) (10) y en (11), además del análisis realizado en esta investigación, se podrá evaluar la calidad de dichas especificaciones de requisitos, como producto de trabajo de la ingeniería de requisitos, a través de la representación de los siguientes atributos de calidad: correctitud, no ambigüedad, completitud, consistencia, verificabilidad, trazabilidad y priorización de los requisitos por su importancia y/o estabilidad.

Los simuladores virtuales en general, poseen particularidades que los diferencian del resto de las aplicaciones informáticas y que obstaculizan el desarrollo de este proceso que tiene como objetivo fundamental definir claramente las funcionalidades que conformarán el sistema propuesto. A continuación se especifican algunas de estas características.

1. Relaciones estrechas de trabajo entre disímiles campos de las ciencias: los ingenieros de software deben obtener y modelar la información, el conocimiento especializado de profesionales de la medicina, la aviación, la gerencia empresarial, la conducción automovilística, etc.
2. Comportamiento en tiempo real de la simulación manteniendo un nivel aceptable de visualización y realismo: es necesario lograr que la interacción, navegación y visualización se establezca en un tiempo real. Dicho de otra manera, el sistema debe responder las acciones ejecutadas en milésimas de segundos.
3. Modelación de objetos con apariencia y propiedades físicas adicionales y en relación con sus funciones y comportamiento: los simuladores incluyen la modelación de entornos reales con disímiles características visuales mediante la representación tridimensional de sus objetos con sus respectivas características físicas que hacen posible la simulación de su comportamiento real, alcanzando efectos como el movimiento de un carro, de una bala o la deformación sufrida por un órgano al ser cortado.
4. Consideración de diferentes estilos y modalidades de técnicas de interacción acorde a las diferentes tareas y dispositivos de entrada/salida: los SV establecen una comunicación de entrada/salida con diversos equipos externos en dependencia del tipo de simulador, facilitando la comunicación hombre-máquina,

de manera que el sistema pueda ejecutar las órdenes emitidas por el usuario al interactuar con los dispositivos.

Nuestro país, específicamente el Polo de Realidad Virtual (PRV) de la Universidad de las Ciencias Informáticas (UCI), ha venido desarrollando experiencia en el tema de la RV hasta proponerse la construcción de algunos SV como simulador de tiro, simulador de conducción, simulador quirúrgico, entre otros. Para lograr una clara definición de lo que realmente se pretende realizar en cada uno es necesario partir de una eficiente propuesta de requisitos.

En el (Anexo 1) se muestra una lista de chequeo (12) confeccionada con el objetivo de identificar los principales problemas que acometen los procesos de IR en la producción de simuladores virtuales, representados por cuatro proyectos de esta categoría que se desarrollan en el PRV antes mencionado. Concluido un espacio de entrevista con los responsables de dichos proyectos se constató que sólo dos de las propuestas presentan una lista formal de requerimientos que guíe su proceso de desarrollo. En los restantes casos únicamente existen pequeños textos escritos en párrafos que intentan explicar, a groso modo, la idea a implementar pero nunca llegan a considerarse como requisitos. Los entrevistados sustentan el origen de sus desarrollos por simple iniciativa de un pequeño equipo con algunas ideas autodidactas iniciales que fueron enriqueciéndose durante el avance en la implementación como actividad primitiva, lo que provocó constantes cambios durante su proceso sin tener registro o control alguno de dichos cambios. Otro de los desfavorables resultados arrojados fue la falta de claridad e incompletitud en las funcionalidades comprendidas en el producto, lo que influye en el desmedido tiempo a emplear en sus desarrollos.

Luego de aplicada la lista de chequeo a los dos proyectos posibles se evidencia que no todos los requerimientos son correctos y tampoco se expresan con claridad y concisión. En ninguno de los dos proyectos se logra una total completitud ni se tienen en cuenta las posibilidades de cambios en los requisitos. Se manifiesta, además, imprecisiones en la consistencia y trazabilidad y en una de las muestras se presentan requisitos imposibles de verificar. En ningún caso se incluyen anexos como figuras, tablas y/o diagramas que faciliten la comprensión de lo planteado y tampoco se tiene en cuenta una priorización de los requisitos según su importancia y/ estabilidad.

Teniendo en cuenta que la producción de SV en Cuba se concentra en el PRV de la UCI y en el Centro de Investigación y Desarrollo de Simuladores Profesionales (SIMPRO), que a la vez trabajan actualmente en colaboración, se considera una

población de 23 profesionales de la informática en el país vinculados directamente al desarrollo de simuladores virtuales y que además se han desempeñado como líderes y/o analistas de proyectos, de los cuales se tuvieron en cuenta 10 como muestra representativa a estudiar en esta investigación. Como resultado de una encuesta (Anexo 2) que les fue aplicada, se pudo comprobar que el 70% de los encuestados asignan un alto nivel de prioridad a los requisitos dentro del proceso de desarrollo de software y sin embargo sólo el 10% aportan alta calificación a la calidad de los mismos y a sus especificaciones. Estos resultados dejan de ser paradójicos si se tiene en cuenta que el 50 % de los encuestados no desarrollan actividades concernientes a los procesos de análisis, especificación y validación de los requisitos. El 20% de los interrogados notifica no utilizar técnicas para especificar los requisitos y de los restantes, el 50% emplean solamente la conocida técnica de casos de uso. Similares son los resultados acerca del manejo de las técnicas para llevar a cabo la validación de requisitos donde el 50% no hacen uso de ellas y del resto, solo el 10% construyen prototipos. Las ventajas que proporciona el uso de herramientas software para llevar a cabo los procesos de la IR tampoco son aprovechadas en el desarrollo de simuladores virtuales y así lo constatan el 80% de los encuestados. Todo lo explicado permite afirmar que las técnicas y herramientas de la IR utilizadas en el desarrollo de simuladores virtuales no son las idóneas para este tipo de proyectos.

Teniendo en cuenta la situación expuesta anteriormente se define el siguiente **problema de investigación**:

La insuficiencia en la adaptación de las técnicas y herramientas de la ingeniería de requisitos para simuladores virtuales influye negativamente en la calidad del producto de trabajo de la ingeniería de requisitos para proyectos de simulación virtual.

Como **hipótesis de la investigación** se plantea que si se proponen las técnicas y herramientas de la ingeniería de requisitos adecuadas para simuladores virtuales se logrará una mayor calidad del producto de trabajo de la ingeniería de requisitos para proyectos de simulación virtual.

Para dar solución al problema planteado se ha formulado como **objetivo general de la investigación** proponer las técnicas y herramientas de la ingeniería de requisitos adecuadas para proyectos de simulación virtual.

Por lo que se tiene como **objeto de estudio** la ingeniería de requisitos, especificándose como **campo de acción** las técnicas y herramientas de la ingeniería de requisitos para proyectos de simulación virtual.

Para cumplimentar el objetivo general se han formulado algunos **objetivos específicos** que facilitarán el desarrollo del mismo:

- Elaborar marco teórico de la investigación.
- Evaluar el estado del arte acerca de la ingeniería de requisitos en simuladores virtuales.
- Proponer las técnicas y herramientas de la ingeniería de requisitos adecuadas para simuladores virtuales.
- Validar teóricamente la mejora en el cumplimiento de los objetivos de los procesos de la IR mediante las técnicas y herramientas propuestas.
- Aplicar de manera práctica las técnicas y herramientas propuestas en el proyecto simulador quirúrgico que se desarrolla en el PRV de la UCI.
- Evaluar los resultados obtenidos de aplicar las técnicas y herramientas propuestas en el proyecto simulador quirúrgico que se desarrolla en el PRV de la UCI.

A continuación se muestran los **métodos de investigación** a utilizar:

- Métodos teóricos
 - **Histórico-lógico** para la evaluación del estado del arte tomado como referencia para la investigación y como punto de comparación de los resultados alcanzados.
 - **Hipotético deductivo** para el análisis y la definición de la hipótesis de la investigación que será verificada o probada en función del estudio de elementos más particulares, de menor nivel de generalidad.
 - **Sistémico** para determinar los elementos de la ingeniería de requisitos (procesos, técnicas y herramientas) y las posibles relaciones existentes entre ellos, hasta lograr un funcionamiento integrado de todas las técnicas y herramientas propuestas.
- Métodos empíricos
 - **Entrevista y Encuesta:** para describir la situación actual de los procesos de ingeniería de requisitos en los proyectos de simulación virtual del país.
- Métodos matemáticos
 - **Método de expertos:** para determinar la validez y objetividad de las técnicas y herramientas propuestas en la investigación.

La tesis estará estructurada en tres capítulos. En un primer capítulo se abordará el marco teórico de la investigación analizándose primeramente, el estado del arte de los simuladores virtuales y de la ingeniería de requisitos y luego se investigará la situación de la IR en proyectos de simulación virtual del país.

En los inicios del capítulo 2 se especificarán las principales características de los simuladores virtuales que lo diferencian del resto de los sistemas software. A partir de estas particularidades se definirán un conjunto de técnicas para cada uno de los procesos de la ingeniería de requisitos a utilizar en el desarrollo de simuladores virtuales, así como las herramientas que brindan soporte a dichas técnicas.

En el tercer capítulo se incluirá primeramente una validación mediante el Método de Expertos que permite afirmar la posible mejora en el cumplimiento de los objetivos o propósitos de los procesos de la IR y viabiliza la ejecución de una aplicación práctica de las técnicas y herramientas propuestas en uno de los proyectos de simulación virtual que se desarrolla en el polo de realidad virtual de la UCI. Seguidamente se detallará la situación inicial de dicho proyecto antes de aplicada la propuesta que permitirá valorar el impacto de los resultados arrojados luego del empleo de las técnicas y herramientas recomendadas.

Capítulo 1: Fundamentación teórica de la ingeniería de requisitos en simuladores virtuales.

En esta fundamentación se pretende realizar un estudio del estado del arte acerca de la ingeniería de requisitos y su vinculación en el desarrollo de simuladores virtuales. Se analizarán diferentes enfoques de la IR reportados en la literatura y se formalizarán los principales problemas que afronta la industria del software actualmente relacionados con los requisitos. Se verificarán las posibles técnicas a utilizar en cada uno de los procesos definidos en la ingeniería de requisitos, así como las herramientas disponibles a utilizar. Se especificarán las diferentes formas de evaluar la calidad de las especificaciones de requisitos como producto de trabajo de esta disciplina y por último se analizarán los principales problemas relacionados con los requisitos detectados en el desarrollo de simuladores virtuales.

1.1 Estado del arte de los simuladores virtuales

Los avances tecnológicos de los últimos años han posibilitado el desarrollo de la RV de forma acelerada. La simulación se ha convertido en una técnica eficaz que permite reproducir los procesos reales cuando por problemas de tiempo, recursos o seguridad no es posible realizar la actividad en su medio natural, posibilitando el trabajo en campos o a distancias que antes impedían cualquier tipo de comunicación (13). Grandes son los beneficios que brindan los simuladores virtuales en las diferentes ramas como por ejemplo, en la industria del entretenimiento (videojuegos), automotriz (simulador de carro), en la medicina (realización de prótesis, cirugías mínimamente invasivas, ingeniería genética), en la meteorología (estudio de tormentas eléctricas), paseos virtuales o en lo referido a la industria militar desarrollándose habilidades mediante una ambientación digital con un gran ahorro de recursos y un arduo entrenamiento (14).

A continuación se presentan algunos ejemplos de SV desarrollados mundialmente.

- VATSIM e IVAO son dos ejemplos de espacios aéreos existentes en la red (internet) que simulan el tráfico aéreo real. En estos sistemas el usuario puede entrenarse tanto como controlador aéreo o como piloto sin necesidad de subir a un avión real y recibir señales contradictorias enviadas por radio. En estos simuladores se crea una copia perfecta de todas las decisiones tomadas durante el vuelo, obteniéndose la ventaja de que en caso de duda un piloto puede volver

sobre sus pasos y encontrar aquella decisión que le puso en peligro, además de poder aprender de los vuelos de otros usuarios (15).

- LABSAG es una plataforma que contiene 10 simuladores de negocio (*business games*), tanto en español como en inglés, producidos para el desarrollo de capacidades y habilidades de gerencia en el área empresarial, de manera que el participante adquirirá grandes experiencias en la toma de decisiones sin espera alguna y sin el desperdicio de recursos reales (16).
- En (17) se puede encontrar el desarrollo de un simulador y su aplicación a la realización de prácticas de topografía donde los alumnos pueden utilizar un teodolito virtual y realizar mediciones sobre entornos, previamente a la realización de la práctica real.
- Symbionix es una compañía productora de simuladores médicos que recientemente desarrolló un nuevo simulador de endoscopía basado en la producción de un modelo tridimensional geométrico. Este nuevo producto posee una textura grabada en video durante un verdadero procedimiento endoscópico, a través del cual los aprendices pueden ejercitar tales procedimientos hasta alcanzar la capacidad necesaria para enfrentarse a una cirugía real. Este entrenamiento reduce el número de errores críticos que amenazan vida potencialmente (18).
- Racer es un sorprendente simulador de carreras de autos que permite una conducción realista, ya sea por entrenamientos en el circuito o disputando carreras con otros jugadores en red local o a través de internet. Este entrenamiento realista es producto de un estudio de física del movimiento y diseño 3D implementado en *OpenGL (Open Graphics Library)* (19).

1.1.1. Desarrollo de Simuladores Virtuales en Cuba

El desarrollo científico alcanzado por Cuba y la necesidad de avanzar aceleradamente en el uso de las nuevas tecnologías de la información y las comunicaciones, ha provocado una evolución cuantitativa y cualitativa en la rama de la Informática Gráfica. Ejemplo de esto es la concepción y el desarrollo de espacios virtuales en Internet para la ciencia, encaminados a promover el trabajo colaborativo entre comunidades científicas y académicas, tomando como base la infraestructura técnica y organizativa existente en el país y su proyección para los próximos años (20).

A mediados de la década del noventa se hizo necesario buscar alternativas dirigidas a elevar la preparación combativa de las tropas con un mínimo de recursos y sustituir, con medios propios, los antiguos equipos entrenadores electromecánicos soviéticos, deteriorados por las carencias provocadas por el periodo especial. Es por ello que los especialistas del Centro de Investigación y Desarrollo de Simuladores Profesionales, perteneciente a las Fuerzas Armadas Revolucionarias (FAR), en coordinación con la industria militar cubana y con otras entidades de la economía nacional, se trazan como reto la producción de simuladores virtuales que solucionaran la problemática existente. Una muestra de ello lo constituyen varias decenas de simuladores virtuales existentes en el país, diseminados por unidades de las FAR, escuelas de automovilismo del Ministerio de Transporte, y de patrullas del Ministerio del Interior. Unos permiten el entrenamiento para conducción de tanques, carros blindados y de distintos vehículos; otros, el tiro coheteril, de infantería, artillería y de tanques; algunos, los más complejos, capacitan a pilotos de aeronaves de transporte y de combate. Todos son diseñados y fabricados en Cuba.

Actualmente los soldados de las FAR pueden realizar sus prácticas en el simulador de tiro de infantería posibilitando el ahorro de municiones y la creación de habilidades sin correr peligro alguno. Este simulador constituye un modelo a escala real que reproduce íntegramente la cabina de los tanques como vehículos de combate. En la pantalla del conductor se presentan diferentes relieves del terreno a transitar, condiciones climatológicas, movimientos y sonidos característicos de este medio de combate, y obstáculos a vencer: desde puentes, túneles y asentamientos, hasta campos minados y zanjas antitanques (21).

En el año 2003 la empresa SIMPRO en unión con el Polo de Realidad Virtual de la Universidad de Ciencias Informáticas hicieron posible la realización de un simulador de conducción. El aspirante, mediante su entrenamiento en una cabina real que transita por una ciudad virtual, puede obtener las habilidades necesarias para conducir. Esta nueva oportunidad elimina los riesgos de accidentes automovilísticos durante una fase de aprendizaje, además del ahorro de combustible necesario para el entrenamiento.

Actualmente este polo productivo dentro de la UCI se enmarca en el desarrollo de variadas soluciones que abarcan, desde la perfección de simuladores de conducción, hasta la posibilidad de que los médicos cirujanos puedan realizar sus entrenamientos de cirugía mínimamente invasiva en una propuesta de simulador quirúrgico que se ejecuta en unión del ministerio de salud pública.

1.2 Requisitos de software

Uno de los párrafos más citados en la bibliografía de la Ingeniería del Software señala: "La parte más difícil de construir un sistema es precisamente saber qué construir. Ninguna otra parte del trabajo conceptual es tan difícil como establecer los requerimientos técnicos detallados, incluyendo todas las interfaces con personas, máquinas y otros sistemas. Ninguna otra parte del trabajo afecta tanto el sistema si no se realiza correctamente. Ninguna es tan difícil de corregir más adelante. Entonces, la tarea más importante que el ingeniero de software hace para el cliente es la extracción iterativa y el refinamiento de los requerimientos del producto" (7).

Un proyecto no puede ser exitoso sin una especificación correcta y exhaustiva de los requerimientos. Precisamente algunos de los problemas que persisten en la industria del software actualmente están dados por un inadecuado entendimiento de las necesidades de los usuarios, la incapacidad de absorber cambios en los requerimientos y las insatisfacciones de los clientes por inaceptable o bajo desempeño del software. Algunos investigadores de la industria de software han utilizado diferentes definiciones para representar verbalmente el concepto de requisito.

- La *IEEE Standard Glossary of Software Engineering Terminology* define un requisito como (22):
 - I. Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
 - II. Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
 - III. Una representación documentada de una condición o capacidad como en I o II.

En esta definición no se tratan los requisitos en función de la satisfacción de clientes y usuarios.
- Craig Larman define los requisitos como una descripción de las necesidades o deseos de un producto y delimita la meta primaria de la fase de requerimientos en identificar y documentar lo que en realidad se necesita, en una forma que claramente se lo comunique al cliente y a los miembros del equipo de desarrollo (23).

En esta definición sí se tiene en cuenta a los requisitos como forma de entendimiento entre los desarrolladores y los clientes.

- SWEBOK *Guide de la IEEE Computer Society* en su versión del 2004 da también su propia definición de requisito. “En lo más básico, un requisito de software es una propiedad que tiene que ser expuesta para resolver un problema determinado del mundo real. Por lo tanto, es una propiedad que tiene que ser exhibida por un software desarrollado o adaptado para resolver algún problema en particular” (24).

Después de analizar las diferentes definiciones de requerimientos citadas anteriormente, en la presente investigación se define un requisito como las características o cualidades, condiciones y capacidades de un sistema, descritas claramente tanto para el equipo de desarrollo como para clientes y/o usuarios en función de satisfacer sus necesidades.

1.3 Importancia de los requisitos en la Industria del Software

No todos los gerentes de software están convencidos de que su equipo necesita hacer un mejor trabajo en el desarrollo y gestión de los requisitos. Sin embargo numerosos estudios de la industria indican que los requisitos constituyen una causa del fracaso de los proyectos de software. La primera edición del informe de CHAOS realizado por el *Standish Group* en el año 1994 indica que tres de los principales factores que conducen al fracaso de proyectos concuerdan con actividades dentro de la IR (25) (26).

- La falta de entrada de información por parte de los usuarios: 13% de todos los proyectos.
- Requisitos y especificaciones incompletas: 12% de todos los proyectos.
- Requisitos y especificaciones cambiantes: 12% de todos los proyectos.

Aunque en la mayoría de los proyectos puede parecer experiencia los excesos de cronograma y presupuesto o la completa cancelación, en el año 2006 el *Standish Group* revela, en su sexta edición del mismo reporte, que el 35% de los proyectos pueden ser catalogados como exitosos; lo que significa que terminaron en tiempo, sobre el presupuesto y en correspondencia con los requisitos de los usuarios. Aunque es notable la mejora de estos resultados con respecto a los arrojados en el informe de 1994, donde se señalaban como exitosos sólo el 16,2% de los proyectos, aún no son satisfactorios. Este mismo estudio indica además, que el 19% de los proyectos

iniciados han sido cancelados antes de que se completen y que el 46% se han cambiado o modificado, lo que representa que se excedieron los costos o el tiempo y/o que no satisfacen los requerimientos del usuario (27).

En informes del SEI (*Software Engineering Institute*) se plantea que alrededor de un 50% de los problemas que sufren nuestros proyectos de software están originados en una deficiente etapa de definición y gestión de requisitos. Sobre el 45% del esfuerzo total de un proyecto se debe a costos de re-trabajo debido también a requisitos defectuosos. Un problema en la fase de requisitos, si no es encontrado a tiempo, puede costar entre 5 y 200 veces más esfuerzo solucionarlo que si se hubiese detectado a tiempo (28).

Otros estudios han llegado a resultados más sorprendentes. Por ejemplo, la ESPITI (*European Software Process Improvement Training Initiative*) realizó un estudio para determinar la importancia relativa de los diversos tipos de problemas de software en la industria. Los resultados de esta encuesta a gran escala, basado en 3800 respuestas, se muestran en la (Figura 1.1) (29).

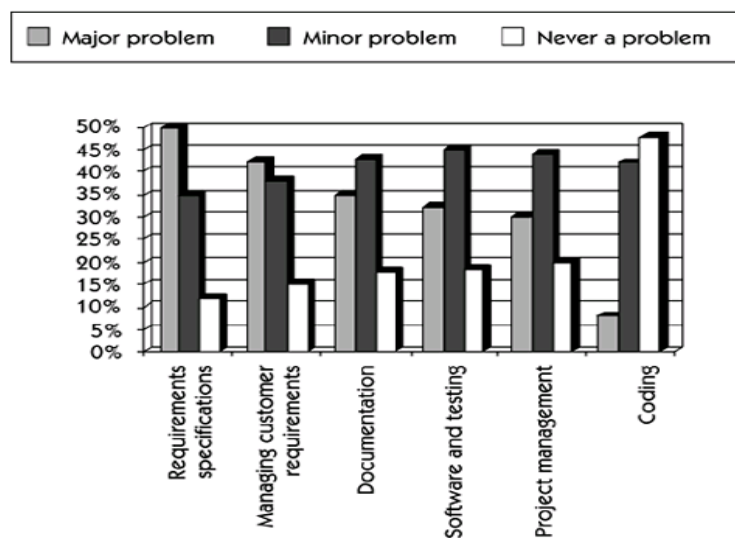


Figura 1.1: Los mayores problemas de desarrollo de software por categoría (29).

Los dos principales problemas que aparecen en aproximadamente la mitad de las respuestas son:

- Especificación de requisitos
- Gestión de requisitos de los clientes

Según Boehm el re-trabajo, hacer algo que ya se pensaba hecho, constituye la mayor consecuencia de los problemas en los requisitos. Ello puede consumir del 30% al 50%

del costo total de desarrollo y los errores en los requisitos representan un 70% a un 85% del costo del re-trabajo. Como se ilustra en la (Figura 1.2), cuesta mucho más corregir un defecto encontrado en etapas posteriores del desarrollo del proyecto que solucionar el problema poco después de su creación. Por tanto la prevención de errores en los requisitos tiene un efecto considerable en la reducción del re-trabajo (30).

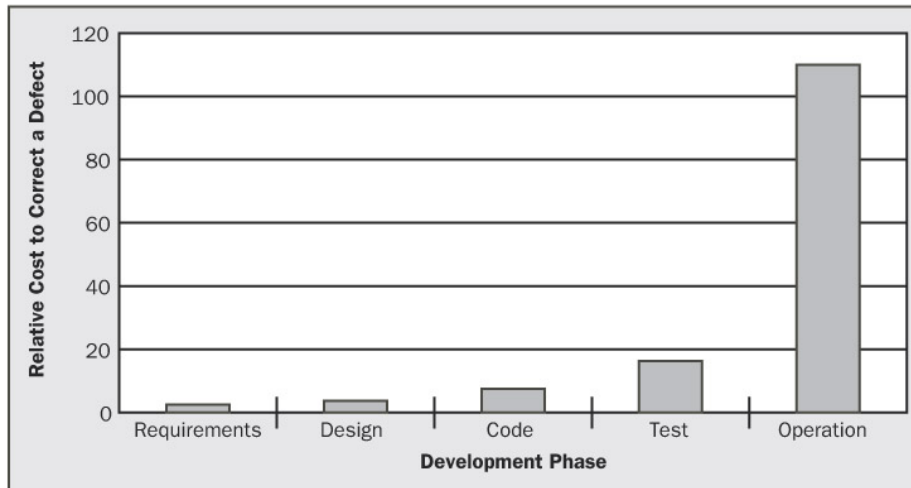


Figura 1.2: Costo relativo a la corrección de defectos en las etapas de desarrollo (30).

Estos detalles constan más especificados en (25), donde se confirma que los errores en la etapa de requisitos son muy perjudiciales principalmente por el empeño que se requiere por parte del equipo de desarrollo en una amplia revisión y un arduo trabajo para corregirlos. Múltiples estudios han demostrado que el costo de corregir un defecto de software aumenta de manera espectacular en la medida en que avance el desarrollo del proyecto, como se muestra en la (Tabla 1.1). Puede costar de 68 a 110 veces más corregir un defecto de requisitos en la etapa operacional de lo que costaría si ese mismo defecto fuera encontrado durante la etapa de desarrollo de requisitos. Un error, omisión, o una mala interpretación en los requisitos exige a los desarrolladores rehacer todo el trabajo ya realizado sobre la base de los requisitos incorrectos. Por lo tanto, cualquier técnica que puede reducir los errores en los requisitos y prevenir algunos de estos derrochados esfuerzos es un gran multiplicador de las inversiones efectivas.

Etapa en que se detecta el error	Costo relativo para corregir el error
Desarrollo de requisitos	1X
Diseño	2-3X
Construcción	5-10X
Pruebas de sistema o aceptación	8-20X
Operación	68-110X

Tabla 1.1: Costo relativo de corregir un defecto de requisitos (25) .

1.4 La Ingeniería de requisitos

Muchos son los ejemplos y estadísticas que demuestran el alto costo de errores en las actividades relacionadas con los requisitos en los sistemas de software. Es por ello que en los últimos años, a la par del desarrollo de la industria de software, ha aumentado también el interés por adoptar las mejores prácticas en los requisitos de software. En este sentido ha surgido el término Ingeniería de Requisitos con el objetivo de englobar los procesos de gestión de requisitos en el ciclo de vida del software.

Una definición algo antigua, pero todavía vigente es la citada por Boehm en 1979.

“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“ (6).

Según (31) una posible visión de la IR es considerarla como un proceso de construcción de una especificación de requerimientos en el que se avanza desde especificaciones iniciales, que no poseen las propiedades oportunas, hasta especificaciones finales completas, formales y acordadas entre todos los participantes. (Ver Figura 1.3)

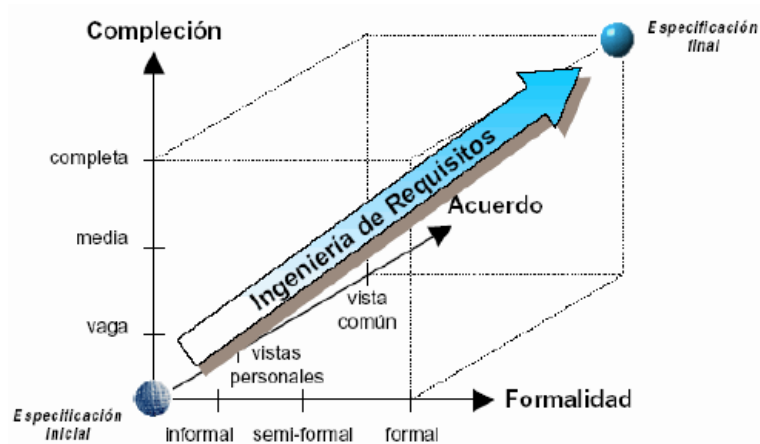


Figura 1.3: Posible visión de la IR (31).

En (9) Pressman retoma que la IR facilita el mecanismo apropiado para comprender lo que quiere el cliente, analizando necesidades, confirmando su viabilidad, negociando una solución razonable, especificando la solución sin ambigüedad, validando la especificación y gestionando los requisitos para que se transformen en un sistema operacional.

El Instituto de Ingeniería de Software (SEI) en su glosario de términos define Ingeniería de Requisitos de la siguiente manera: “La ingeniería de Requisitos involucra todas las actividades del ciclo de vida dedicadas a la identificación de los requisitos de usuario, análisis de requisitos para derivar requisitos adicionales, documentación de los requisitos como una especificación y validación de los requisitos documentados contra las necesidades del usuario, así como los procesos que soportan estas actividades” (32).

Cualquiera que sea la definición usada para la IR, debe señalarse su importancia en el proceso de desarrollo de software de un proyecto.

1.5 La ingeniería de requisitos según estándares de calidad

Los estándares de calidad desarrollan normas y modelos en función de organizar y formalizar los procesos en la industria del software incluyendo los relativos a la Ingeniería de Requisitos. El Modelo Integrado de Madurez de Capacidades (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) constituyen un conjunto representativo de los más

conocidos y adoptados a nivel mundial.

1.5.1 Ingeniería de requisitos en CMMI

El CMMI, desarrollado por el Instituto de Ingeniería del Software (SEI) constituye un modelo de procesos mejorados de manera incremental, de ahí que su objetivo consiste en medir la madurez organizacional o la capacidad de ejecutar procesos de una entidad productora de software. CMMI describe 6 áreas de proceso ingenieriles. Las 2 primeras abordan las actividades de la IR: Desarrollo de Requisitos, Gestión de Requisitos, Solución Técnica, Integración del Producto, Verificación y Validación (33).

Área de proceso: Desarrollo de requisitos

El área del proceso Desarrollo de Requisitos se ubica en el nivel 3 de madurez y su propósito consiste en producir y analizar los requisitos del cliente, del producto, y de las componentes del producto. El área incluye tres objetivos específicos y diez prácticas específicas (33):

Objetivos específicos	Prácticas específicas
<p>OE1 Obtener requisitos del cliente</p> <p>Se relevan las necesidades, expectativas y restricciones y se traducen en requisitos del cliente.</p>	<p>PE 1.1 Obtener necesidades.</p> <p>PE 1.2 Elaborar requisitos del cliente.</p>
<p>OE2 Obtener requisitos del producto</p> <p>Los requisitos del cliente son refinados para obtener los requisitos del producto y sus componentes.</p>	<p>PE 2.1 Establecer requisitos del producto y de los componentes del producto.</p> <p>PE 2.2 Ubicar requisitos de componentes del producto.</p> <p>PE 2.3 Identificar requisitos de interfaz.</p>
<p>OE3 Analizar y validar requisitos</p> <p>Los requisitos son analizados y</p>	

<p>validados, y se desarrolla una definición de la funcionalidad requerida.</p>	<p>PE 3.1 Establecer conceptos operacionales y escenarios asociados.</p> <p>PE 3.2 Establecer definición de funcionalidades requeridas.</p> <p>PE 3.3 Analizar Requisitos.</p> <p>PE 3.4 Analizar los requisitos para ver su alcance.</p> <p>PE 3.5 Validar Requisitos.</p>
---	---

Tabla 1.2: Objetivos y prácticas específicas para el desarrollo de requisitos.

Área de proceso: Gestión de requisitos

El área del proceso Gestión de Requisitos, ubicada en el nivel 2 de madurez tiene como objetivo controlar los requisitos que deberán satisfacer el producto a desarrollar. El área incluye solamente un objetivo específico y cinco prácticas específicas (33):

Objetivos específicos	Prácticas específicas
<p>OE1 Gestionar Requisitos</p> <p>Los requisitos son gestionados y se identifican las inconsistencias entre los productos del trabajo y los requisitos.</p>	<p>PE 1.1 Obtener un entendimiento de los requisitos.</p> <p>PE 1.2 Obtener compromiso con los requisitos.</p> <p>PE 1.3 Gestionar cambios en los requisitos.</p> <p>PE 1.4 Mantener trazabilidad bidireccional de los requisitos.</p> <p>PE 1.5 Identificar inconsistencias entre el trabajo del proyecto y los requisitos.</p>

Tabla 1.3: Objetivos y prácticas específicas para la gestión de requisitos.

1.5.2 Ingeniería de requisitos en las prácticas recomendadas por la IEEE

El documento normativo Prácticas Recomendadas para la Especificación de Requisitos de Software (PRERS) de la IEEE establece criterios que deben ser considerados para producir buenas Especificaciones de Requisitos de Software (ERS). Cada uno de estos criterios son detallados en la norma (10):

- a) La naturaleza de la ERS.
- b) El ambiente de la ERS.
- c) Las características de una buena ERS.
- d) La preparación conjunta de las ERS.
- e) La evolución de la ERS.
- f) Prototipos.
- g) El diseño embebido de la ERS.
- h) Los requisitos del proyecto embebidos en la ERS.

En la norma también se propone un estándar para guiar la formalización de una ERS donde se detallan las partes esenciales:

Tabla de contenidos

1. Introducción
 - 1.1. Propósito
 - 1.2. Alcance
 - 1.3. Definición, acrónimos y abreviaturas
 - 1.4. Referencias
 - 1.5. Apreciación general
2. Descripción General
 - 2.1. Perspectiva del producto
 - 2.2. Funciones del producto
 - 2.3. Características del usuario
 - 2.4. Restricciones
 - 2.5. Suposiciones y dependencias
3. Requisitos Específicos

Apéndices

Índice

Tabla 1.4: Prototipo de secciones de una ERS (10).

Las prácticas recomendadas en las normativas de la IEEE constituyen una significativa referencia para la ERS.

1.5.3 Ingeniería de requisitos en la ISO

La Organización Internacional para la Estandarización pretende estandarizar internacionalmente la industria del software a través de las guías o modelos como la ISO 9001 en la cual se describen los elementos de un sistema de calidad orientado al software, se especifican los requisitos para un sistema de gestión de la calidad aplicable a toda organización en aras de proporcionar productos que satisfagan los requisitos del cliente. En su acápite 7.2, referente a los procesos relacionados con el cliente, la ISO 90003 establece los elementos afines a la determinación y revisión de los requisitos relacionados con el producto (34).

En la determinación de los requisitos relacionados con el producto se enfatiza en que 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.

En la revisión de los requisitos relacionados con el producto, según la norma, la organización debe revisar los requisitos relacionados con el producto. Esta revisión debe efectuarse antes de que la organización se comprometa a proporcionar un producto al cliente y debe cerciorarse que: estén definidos los requisitos del producto, estén resueltas las diferencias existentes entre los requisitos del contrato o pedido y los expresados previamente y que la organización tiene la capacidad para cumplir con los requisitos definidos.

1.6 Procedimientos usados en IR

En (7) Pressman define el proceso de IR en siete áreas de esfuerzo: inicio, obtención, elaboración, negociación, especificación, validación y gestión.

En (30) Wiegers divide el dominio de la IR en:

- Desarrollo de requisitos
- Gestión de requisitos

El desarrollo de requisitos es dividido a su vez en cuatro subprocesos (Ver Figura 1.4).

- Obtención o Elicitación
- Análisis
- Especificación
- Validación

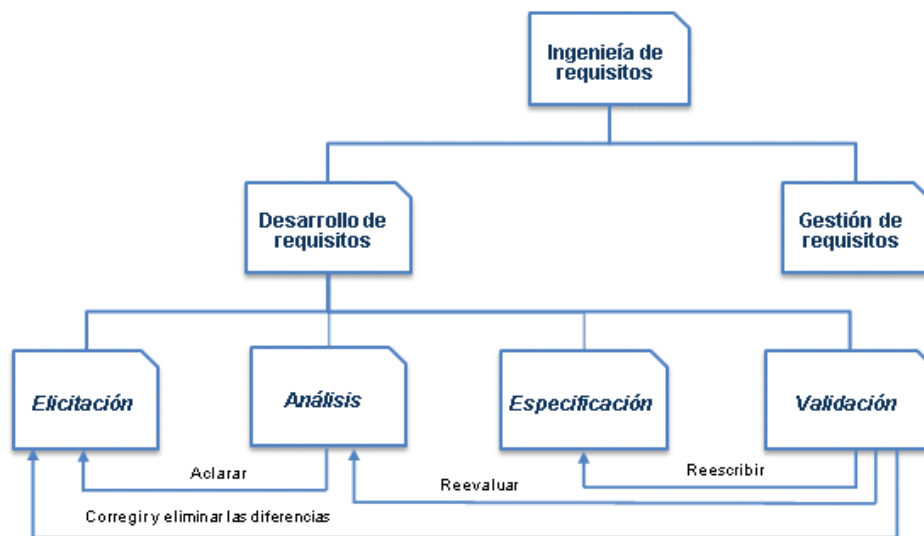


Figura 1.4 Subcomponentes de la IR (30).

En (35) Michael Arias también coincide y afirma que, independientemente de que existen diferentes enfoques de procedimientos usados en la IR, se tiene un denominador común que consiste en el desarrollo de requisitos y que puede resumirse en las siguientes etapas fundamentales: Elicitación, Análisis, Especificación, Validación (Ver Figura 1.5).

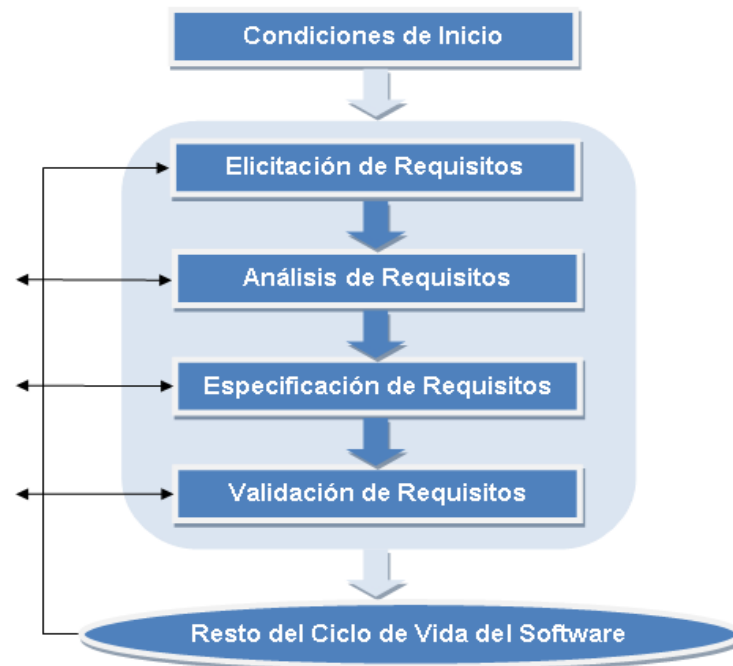


Figura 1.5: Procedimiento del desarrollo de requisitos (35).

1.7 Técnicas utilizadas en Ingeniería de requisitos

Para llevar a cabo los procesos de la ingeniería de requisitos durante la construcción de un sistema no existe una técnica estandarizada y estructurada que ofrezca un marco de desarrollo que garantice la calidad del resultado. Existe en cambio, un conjunto de técnicas que proponen las diferentes metodologías para el desarrollo de aplicaciones. La selección de estas técnicas y el éxito de los resultados que se obtenga dependen en gran medida, tanto del equipo de análisis y desarrollo, como de los propios clientes o usuarios que en ella participen.

1.7.1 Técnicas para la obtención de requisitos

El primer proceso de la ingeniería de requisitos comienza cuando un equipo de desarrollo de software responde a la petición de clientes que pretenden solucionar un problema a partir de una aplicación informática. El éxito en la obtención de requisitos depende del entendimiento a partir de la comunicación entre clientes y desarrolladores. Debido a la complejidad que esto puede implicar, la ingeniería de requisitos ha trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa.

Entrevistas

Las entrevistas resultan una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. A través de esta técnica el equipo de trabajo se acerca al problema de una forma natural. Existen muchos tipos de entrevistas y son muchos los autores que han trabajado en definir su estructura y dar guías para su correcta realización. Básicamente, la estructura de la entrevista abarca tres pasos: identificación de los entrevistados, preparación de la entrevista, realización de la entrevista y análisis de los resultados.

La entrevista no es una técnica sencilla de aplicar, pues requiere que el entrevistador sea experimentado y tenga capacidad para elegir bien a los entrevistados para obtener de ellos toda la información posible en un período de tiempo siempre limitado (12).

JAD (Joint Application Development / Desarrollo conjunto de aplicaciones)

Esta técnica es una práctica de grupo donde se desarrolla un conjunto de reuniones durante un período de 2 a 4 días en las que participan analistas, usuarios, administradores del sistema y clientes. En estos talleres se les ayuda a los clientes y usuarios a formular problemas y proponer posibles soluciones, involucrándolos y haciéndolos partícipes del desarrollo del proyecto (12).

Tormenta de ideas (Brainstorming)

Tormenta de ideas es también una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. Las sesiones de brainstorming deben estar integradas por un número de cuatro a diez personas, uno de los cuales debe asumir el rol de moderador de la sesión, pero sin carácter de controlador.

Como técnica de captura de requisitos es sencilla de aplicar. Frente al JAD tiene la ventaja de que es muy fácil de aprender y requiere poca organización. Además suele ofrecer una visión general de las necesidades del sistema, pero normalmente no se obtienen resultados con el mismo nivel de detalle que en otras técnicas. Sus fases son: preparación, generación, consolidación y documentación (12).

Mapas conceptuales

Los mapas conceptuales son grafos en los que los vértices representan conceptos y las aristas representan posibles relaciones entre dichos conceptos. Estos grafos de relaciones se desarrollan con el usuario y sirven para aclarar los conceptos

relacionados con el sistema a desarrollar. Son muy usados dentro de la ingeniería de requisitos, pues son fáciles de entender por el usuario, más aún si el equipo de desarrollo hace el esfuerzo de elaborarlo en un lenguaje común. Sin embargo, deben ser usados con cautela porque en algunos casos pueden llegar a ser ambiguos en casos complejos, si no se acompaña de una descripción textual (12).

Storyboards

Tal vez ninguna técnica de obtención de requisitos ha sido objeto de tantas interpretaciones como la técnica de storyboard. Sin embargo, la mayoría de estas interpretaciones coinciden en que el propósito de los storyboard es obtener el criterio de los usuarios sobre las funcionalidades propuestas para la aplicación en etapas tempranas del ciclo de vida, mucho antes de que sean comprometidos a código.

Storyboard puede acelerar el desarrollo conceptual de diferentes facetas de una aplicación. Puede ser utilizado para entender reglas del negocio que serán implementadas, para definir algoritmos y otras construcciones matemáticas que deben ser ejecutadas en el sistema o para demostrar reportes u otras salidas impresas (26).

Los storyboards también pueden ser usados en la etapa de especificación de requisitos, de conjunto con la técnica escenario, para explorar, entender y razonar sobre los requisitos funcionales del sistema, específicamente cómo los usuarios interactuarán con el sistema. Proporcionan una descripción lógica y conceptual de las funcionalidades del sistema para un escenario determinado, brindándole un mayor realismo a través el uso de medios pictóricos para especificar los requisitos. No constituyen un primer borrador de la interfaz de usuario, sino que están destinadas a representar sólo la interacción del usuario con el sistema (36).

Aprendiz

Esta técnica se basa en la idea del maestro y en la observación del trabajo real. El ingeniero de software representa al aprendiz y el usuario/cliente cumple el rol de maestro. El aprendiz se sienta con el maestro a aprender por medio de la observación, haciendo preguntas y también realizando algún trabajo bajo la supervisión del maestro.

Generalmente para el cliente/usuario es difícil explicar completamente su trabajo. En ocasiones tiende a omitir elementos que consideran triviales. La técnica del aprendizaje es muy útil para mitigar este problema. Es también una técnica apropiada para un proyecto donde el problema no es estructurado, o sea, en casos en los que se

necesita obtener conocimiento tácito de los clientes/usuarios. Tiene como limitante que su implementación requiere de mucho tiempo (37).

1.7.2 Técnicas para el análisis de requisitos

El análisis de requisitos es el proceso encargado de llevar a un nivel superior de detalle los requisitos del sistema. Implica el perfeccionamiento de los requerimientos para garantizar que todas las partes interesadas los comprendan y examinen lo que evitaría errores, omisiones y otras deficiencias. El análisis incluye además, la descomposición de los requisitos de alto nivel en detalles, la evaluación de viabilidad, y la negociación de las prioridades. El objetivo en esta etapa es formalizar requisitos de suficiente calidad y detalles que permitan construir estimaciones realistas del proyecto y el personal técnico pueda proceder con el diseño y construcción de la solución.

Aunque en la literatura no abundan técnicas dirigidas específicamente al trabajo en esta etapa, en (30) sí se recomiendan algunas prácticas de análisis útiles para la representación y entendimiento de los requisitos desde diferentes puntos de vista, de manera que se favorezca el entendimiento común entre las partes involucradas.

Dibujar un diagrama de contexto

Un diagrama de contexto es un simple modelo de análisis que muestra cómo el nuevo sistema se adapta a su medio ambiente. Define los límites y las interfaces entre el sistema que se está desarrollado y las entidades externas al sistema (usuarios, dispositivos de hardware y otros sistemas de información). Se crea interfaces de usuario que ayuden a las partes interesadas a alcanzar un mejor entendimiento del problema a resolver (30).

Analizar factibilidad de requisito

Esta técnica se basa en evaluar la factibilidad de la implementación de cada requisito a un costo aceptable y el rendimiento en el entorno operativo. Comprender los riesgos asociados a la implementación de cada requisito, incluidos los conflictos con otros requisitos, dependencias de factores externos, y obstáculos técnicos (30).

Diccionario de datos

Se recomienda definir todos los elementos de datos y estructuras asociadas con el sistema, posibilitando que todos los que trabajan en el proyecto utilicen definiciones coherentes. En esta fase de requisitos, el diccionario de datos debe definir los

elementos de datos del problema de dominio para facilitar la comunicación entre los clientes y el equipo de desarrollo (30).

Priorizar los requisitos

Esta práctica propone aplicar un método analítico para determinar la prioridad de la implementación de los requisitos. Basado en la prioridad, se determina cuáles soluciones contiene cada característica o conjunto de requisitos (30). En la literatura se reportan algunos métodos o técnicas a utilizar para establecer la priorización de los requisitos.

- **Árbol de búsqueda binaria**

BST, por sus siglas en inglés (*Binary Search Tree*), es un algoritmo utilizado en la búsqueda de información y puede resultar fácil su adaptación para la priorización de requisitos. El enfoque para este objetivo se basa en colocar todos los requisitos en una pila y establecer comparaciones entre todos los requisitos con un llamado nodo raíz. De esta manera se obtiene una lista con todos los requisitos ordenados según su orden de prioridad. El primer elemento de la lista corresponde al requisito más importante y el último valor de la lista indica el requisito de menor importancia.

- **Técnica de asignación numérica**

Esta técnica proporciona una escala para cada requisito. Se propone dividir los requisitos en tres grupos: obligatorios, deseables, y no esenciales. A cada requisito se le asigna un número en una escala del 1 al 5 para indicar su importancia según la siguiente interpretación:

1. No es de interés (el cliente no lo necesita)
2. No es importante (el cliente acepta su ausencia)
3. Bastante importante (el cliente lo agradece)
4. Muy importante (el cliente no quiere el producto sin su implementación)
5. Obligatorio (el cliente no puede satisfacer sus necesidades sin su implementación).

La clasificación final es el promedio de todas las clasificaciones de los participantes para cada requisito (38).

- **Decisión basada en Costo-Valor usando el Proceso de Jerarquía Analítica**

AHP, por sus siglas en inglés (*Analytic Hierarchy Process*), involucra una jerarquía de criterios de decisión claves y luego hace comparaciones entre cada posible par de cada grupo. Su aplicación específica en la priorización de requisitos se basa en los elementos costo y valor relativos. Para todos los pares de requisitos candidatos se estima un valor y un costo de implementación que permite la comparación de ambos requisitos. Basados en esta información, los jefes de proyecto pueden tomar decisiones como qué requisitos pueden ser postergados o excluidos para mantener el tiempo de entrega del producto a un mínimo (39).

- **Método de los 100 puntos**

Este método se basa en un sistema de votación que se utiliza en ejercicios de tormenta de ideas. Cada uno de los interesados dispone de 100 puntos que pueden utilizar a favor de los requisitos más importantes. Los puntos pueden ser distribuidos de la manera que el interesado desee (38).

1.7.3 Técnicas para la especificación de requisitos

La escritura de los requisitos se refiere a la tarea de reunir una descripción del producto desde el punto de vista de negocio. Típicamente se le llaman una especificación a esta descripción, aunque realmente no es una actividad independiente (37).

Las técnicas más comúnmente usadas para especificar los requisitos de las aplicaciones de software están basadas en lenguaje natural debido a que es fácilmente entendida por los involucrados en el proceso. No obstante existen casos en los que la ambigüedad de los lenguajes naturales no es tolerable, particularmente cuando los requisitos tratan asuntos tan delicados como la vida de un paciente o cuando el comportamiento erróneo de un sistema puede conducir a graves consecuencias financieras o legales (40). En la literatura se reportan algunas técnicas que permiten especificar esos requisitos de una manera más práctica.

Escenarios

Técnica que describe las características del sistema a desarrollar mediante una secuencia de pasos. La representación del escenario puede variar dependiendo del autor, puede ser casi textual o ir encaminada hacia una representación gráfica en forma de diagramas de flujo. El análisis de los escenarios, hechos de una forma u otra, puede ofrecer información importante sobre las necesidades funcionales de sistema

(12). Existen varios estilos para construir escenarios, ya sea mediante la narrativa textual, o haciendo uso combinado con las técnicas de storyboards y prototipos escritos (41).

Pseudocódigo

El pseudocódigo es un “cuasi” lenguaje de programación, que trata de combinar la informalidad del lenguaje natural con la sintaxis estricta y el control de estructuras de un lenguaje de programación (26).

Tablas y árboles de decisión

Es común encontrar requisitos que tengan relación con una combinación de entradas; diferentes combinaciones de estas entradas se relacionan con diferentes comportamientos o salidas. La solución en este caso consiste en describir explícitamente cada una de ellas en una tabla de decisión. Alternativamente puede ser dibujado un árbol de decisión que exprese la misma información (26).

Modelos Entidad-Relación

Si un conjunto de requisitos está asociado con la descripción de la estructura y la relación entre datos dentro del sistema, es conveniente representar esta información en un diagrama Entidad-Relación (DER), utilizando la notación establecida para las técnicas de análisis estructurado. Los DER son específicamente útiles en aplicaciones donde los datos y las relaciones que gobiernan los datos son complejos (7).

A pesar de ser los DER una competente técnica de modelación, tiene la desventaja de ser difícil de leer y entender por personas sin preparación técnica.

Diagramas de actividad

Los diagramas de actividades elaborados utilizando el Lenguaje Unificado de Modelado (UML), tienen la ventaja de una familiaridad razonable. Incluso, personas sin entrenamiento en informática saben qué es un diagrama de actividad. La notación UML proporciona una representación visual que resulta más sencilla de entender que otras representaciones como el pseudocódigo (26).

Máquinas de estado finito (MEF)

En algunos casos es conveniente considerar el sistema o un subconjunto discreto del sistema como una máquina hipotética que puede estar en sólo uno de los estados especificados. Producto de una entrada, como la entrada de datos del usuario o de un dispositivo externo, la máquina cambia su condición y luego genera una salida o lleva

a cabo una acción. Ambos, la salida y el siguiente estado, pueden ser determinadas solamente entendiendo el estado actual y el evento que causó la transición. De esa manera el comportamiento de un sistema es determinista; matemáticamente se puede determinar cada posible estado y, por consiguiente las salidas del sistema, basado en cualquier conjunto de entradas suministradas (26).

Casos de Uso

Aunque inicialmente se desarrollaron como técnica para la especificación de requisitos, algunos autores proponen casos de uso como técnica para la captura de requisitos. Los casos de uso describen una secuencia de acciones que realiza un sistema con un resultado de valor perceptible para un usuario/actor. Permiten mostrar el contorno (actores) y el alcance (requisitos funcionales expresados como casos de uso) de un sistema. La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente. En comparación con métodos tradicionales de requisitos, son relativamente fáciles de escribir y más fácil de leer (29), sin embargo carecen de la precisión necesaria si no se acompañan con una información textual detallada o con otra técnica como pueden ser los diagramas de actividades (12).

1.7.4 Técnicas para la validación de requisitos

Los requisitos una vez definidos necesitan ser validados. Esta etapa tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea. Pocas son las propuestas existentes que ofrecen técnicas para la realización de la validación y muchas de ellas consisten en revisar los modelos obtenidos en la definición de requisitos con el usuario para detectar errores o inconsistencias (12). Aún así, se registran algunas técnicas que pueden ser aplicadas.

Prototipos

Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario (12). En (42) se retoma la definición emitida por el estándar ISO 13407, que define un prototipo como una representación de todo o parte de un producto o sistema que, aunque limitado de algún modo, puede utilizarse con fines de evaluación.

El uso de esta técnica presenta el inconveniente de que el usuario debe entender que lo que está viendo es un prototipo y no el sistema final (12). Sin embargo posibilita una mayor efectividad en la comunicación de los diseñadores y reduce la necesidad y el costo de rehacer un sistema ya implementado cuando los problemas se identifican en etapas avanzadas del desarrollo. La construcción de prototipos es conveniente cuando se pretende involucrar a usuarios en el proceso de desarrollo. En estos casos las especificaciones técnicas y los modelos abstractos no suelen ser una buena vía de comunicación (42).

Listas de chequeo (Checklist)

Esta técnica es muy fácil de utilizar. En general es una lista de preguntas que el analista debe usar para evaluar cada requerimiento verificando y marcando los puntos de la lista mientras se lee el documento de requisitos. Cuando se descubren problemas potenciales deben ser anotados, ya sea en los márgenes del documento o en una lista de análisis. Las listas brindan un recordatorio de lo que se debe buscar y reducen la posibilidad de obviar alguna verificación importante.

Auditorías

La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una lista de chequeo predefinida o definida a comienzos del proceso, de manera que sólo una muestra es revisada (12).

Generación de casos de prueba

El propósito en esta técnica es comprobar que el producto cumple todos los requisitos que pueden ser verificados antes de continuar con el desarrollo de las disciplinas subsecuentes del proceso de desarrollo del software. Basada en la premisa de que cada requerimiento debe ser posible de probar, la técnica propone el diseño de casos de prueba donde se evalúen todos los requerimientos. Un caso de prueba para requisitos funcionales generalmente consiste en definir las entradas y las salidas del sistema, además de acciones del usuario que posibilitan completar lo que expresa el requisito (43).

1.7.5 Técnicas para la gestión de requisitos

Durante el proceso de desarrollo de un software los requisitos pueden ser modificados, por lo que se establece el concepto de Gestión de Requisitos, que es el tratamiento y control de las actualizaciones y cambios a los mismos (40).

Una vez definidos los requisitos iniciales del proyecto es inevitable enfrentar los cambios que los clientes, los administradores, o el equipo de desarrollo puedan solicitar. Este proceso involucra un conjunto de actividades que ayudan al equipo de proyecto a identificar, controlar y rastrear los requisitos y los cambios a estos en cualquier momento mientras se desarrolla el proyecto (9). A continuación se referencian algunas de las prácticas reportadas en la literatura para lograr tales objetivos (30).

Comité de control de cambio

Se propone definir un Comité de Control de cambio, conformado por un pequeño grupo de involucrados (*stakeholders*) del proyecto, que tiene como función recibir los cambios propuestos, evaluarlos, decidir cuáles aceptar y cuáles rechazar y establecer prioridades de aplicación.

Historial de cambios de requisitos

Se establece un registro de las fechas de modificación de las especificaciones de los requisitos, además de especificar los cambios que se hicieron y por qué se propusieron. Una herramienta de control de versiones o herramienta de gestión de requisitos puede automatizar estas tareas.

Rastreo del estado de cada requisito

Se recomienda establecer una base de datos con un registro por cada requisito funcional. Almacenar atributos clave de cada requisito, incluido su estado (propósito, implementado o verificado), de modo que en cualquier momento se pueda controlar el número de requisitos en cada categoría.

Matrices de trazabilidad de requisitos

Matrices de trazabilidad o de seguimiento identifican las relaciones de los requisitos con uno o más aspectos del sistema o su entorno. Se recomienda llenar esta matriz durante el desarrollo y no al final del proyecto. Pueden citarse matrices de trazabilidad de:

- Características: relación de los requisitos con características específicas de la aplicación.
- Orígenes: relación de los requisitos con su fuente.
- Dependencia: relación entre requisitos.
- Subsistemas: relación de los requisitos con el subsistema al que pertenece.

- Interfaz: relación de los requisitos con las interfaces de la aplicación.

1.8 Herramientas utilizadas en ingeniería de requisitos

1.8.1 Herramientas para el desarrollo de requisitos

CASE, o *Computer-Aided Software Engineering*, es un término aplicado por décadas a cualquier sistema o colección de herramientas que ayude a automatizar el diseño de software y los procesos de desarrollo, posibilitando a los desarrolladores de software trabajar, con un alto nivel de abstracción, en la definición de sistemas que posteriormente serán construidos. En los últimos 15 años las herramientas CASE han estado asociadas con familias muy específicas de herramientas que ayudan a definir y validar las especificaciones de un sistema o automatizar el proceso de construcción del software. Para lograr tales propósitos la mayoría de estas herramientas soportan el Lenguaje Unificado de Modelado (UML), el cual posibilita especificaciones del sistema que pueden resultar relativamente fácil su transformación hacia la elaboración y mantenimiento del código (44). A continuación se citan algunas herramientas aceptadas mundialmente.

Rational Rose

Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases.

Enterprise Architect

Enterprise Architect es una herramienta gráfica diseñada con el propósito de apoyar la construcción de softwares robustos y sostenibles. Soporta el análisis y diseño desde el UML 2.1 y cubre las necesidades del desarrollo de software desde la recopilación de requisitos a través de la fase de análisis, modelos de diseño, prueba y mantenimiento. (45).

ArgoUML

Es una herramienta de código abierto (*Open Source*) de modelado UML que incluye soporte para todos los diagramas del estándar UML 1.4. Se puede ejecutar sobre cualquier plataforma Java (46).

Visual Paradigm para UML

Es una herramienta para modelado que soporta UML 2.1, DER y mapas objeto-relacionales. Realiza análisis de requisitos con un amplio rango de opciones (Diagramas de Requisitos SysML, Modelado de casos de uso, Análisis Textual, CRC etc.) (47).

1.8.2 Herramientas para la gestión de requisitos

El proceso de gestión de requisitos a lo largo del proyecto resulta un tanto costoso debido a la carencia de documentación formal y estandarizada de las necesidades del cliente, lo que afecta el proceso de mantenimiento adaptativo o evolutivo del software impidiendo la identificación de los impactos del cambio de algún requisito en el desarrollo del sistema y aumentando la ocurrencia de errores.

Con el aumento de las necesidades en administrar requisitos, algunas compañías han desarrollado herramientas para contribuir al mercado. Algunas funciones comunes realizadas por estas herramientas consisten en la identificación de requisitos, revisión y edición, rastreo de requisitos a su origen y generación de informes (48).

Según (48) el uso de herramientas de gestión de requisitos proporciona a la organización:

- Ahorro en costos de especificación y de desarrollo minimizando el impacto de errores.
- Mejora la calidad mediante un adecuado análisis y gestión de los requisitos.
- Mejora la productividad facilitando la reutilización real desde la especificación.
- Reduce las no-conformidades del sistema.
- Permite administrar más fácilmente la especificación.
- Ayuda a cumplir con estándares de calidad.
- Permite centralizar toda la información del problema.
- Permite especificar sistemas de una forma estructurada y gráfica.
- Proporciona una trazabilidad completa de la especificación.

El Consejo Internacional de Ingeniería de Sistemas (INCOSE), por sus siglas en inglés (*International Council on Systems Engineering*) identifica las siguientes características básicas necesarias para una herramienta ser considerada como una herramienta de Gestión de Requisitos (48):

- Identificación de requisitos "individuales".
- Asignación a un destino y clasificación de requisitos.
- Grupo de requerimientos (recopilación), revisión, identificación / punto de arranque.
- Proveer un interfaz de datos básicos.

Tomando como referencia el estudio realizado por INCOSE (49), donde compara 34 herramientas comerciales en función de sus características y funcionalidades a partir de una solicitud de información a sus proveedores con un total de 81 interrogantes se muestra a continuación cuatro de las herramientas incluidas en el conjunto de aquellas que cumplen con la mayoría de las funciones:

RequisitePro

Herramienta centrada en documentos, que almacena los requisitos asociándolos a documentos (aunque también permite guardarlos directamente en la base de datos), mientras que las otras herramientas están orientadas a requisitos. Auxilia especialmente en el control de cambio de requisitos, con trazabilidad para especificaciones de software y pruebas. Se destaca su integración total con Microsoft Word y permite el uso de Oracle sobre Unix o Windows y también soporta SQL Server sobre Windows (48), (50).

IRqA

Según sus siglas en inglés se denomina "*Integral Requisite Analyzer*". Es una herramienta de ingeniería de requisitos especialmente diseñada para soportar el proceso completo de ingeniería de requisitos. En IRqA el ciclo de especificación completo incluye la captura de requisitos, análisis, especificación de sistema, validación y la organización de requisitos es soportada por modelos estándares (48).

CaliberRM

Herramienta para sistemas grandes y complejos y proporciona una base de datos de requisitos con trazabilidad. Los requisitos se tienen en cuenta como parte del proceso de gestión de la calidad del software. CaliberRM está basado en internet y maneja referencia de documentos, responsabilidad de usuario, trazabilidad, prioridad y estado entre otras características (48).

DOORS

"Dynamic Object Oriented Requirements System" es el significado de sus siglas en inglés. A diferencia del resto de las herramientas, considera los requisitos como objetos y los documentos como módulos. Tiene una orientación basada en objetos, frente a RequisitePro y Caliber-RM, que manejan solamente requisitos y sus atributos. Es una herramienta para organizaciones grandes que necesitan controlar grandes conjuntos de usuarios y requisitos de sistemas con una completa trazabilidad. Proporciona buena visualización de tales documentos como jerárquicas y su lenguaje de extensión permite una gran variedad de soporte de herramientas a ser construidas (48).

1.9 Calidad de las especificaciones de requisitos

Como se mencionaba a inicios de este trabajo, las especificaciones de requisitos son consideradas como el producto de trabajo de la ingeniería de requisitos, basado en la colección, organización y representación de todos los requisitos del proyecto. Su importancia está representada en la posibilidad de establecer las bases de un acuerdo entre clientes/usuarios y equipo de desarrollo sobre los que el producto de software debe hacer, en reducir el esfuerzo para evitar rediseños, recodificación y repuebas y además, provee las bases para la estimación del costo y entrega del producto.

Es indudable la influencia de la calidad de los requisitos y sus especificaciones en la calidad de la solución final. Muchos son los ingenieros de software que se han visto forzados a lidiar con especificaciones incompletas, inconsistentes o ambiguas y han experimentado la confusión que invariablemente provocan, reflejándose los resultados en la fecha de entrega y la calidad del software (9). Algunos autores han definido formalmente atributos para medir la calidad de tales especificaciones de requisitos.

- En (9) Pressman maneja la calidad de las especificaciones de requisitos a partir de los parámetros de claridad, completitud y consistencia.
- En las prácticas recomendadas por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) (10) para la especificación de requisitos de software se considera que tal especificación es de calidad cuando es "correcta, no ambigua, completa, consistente, ordenada por importancia y/o estabilidad, verificable, modificable y trazable".

- La dirección de calidad de software de la UCI recomienda algunas prácticas para la especificación de requisitos de software a utilizar en los proyectos productivos que se encuentran en desarrollo (11). La calidad de las especificaciones de requisitos es bien aceptada si es “correcta, no ambigua, completa, consistente, verificable, modificable, trazable y los requisitos son clasificados por su importancia y/o estabilidad”, coincidiendo de esta manera con los establecidos en las prácticas de la IEEE.

En el presente trabajo se concibe una especificación de requisitos de software con calidad si es “correcta, no ambigua, completa, consistente, verificable, trazable y los requisitos son clasificados por importancia/estabilidad”. Seguidamente se especifican cada uno de los atributos mencionados.

- Especificación correcta: todos los requisitos contenidos representan algo que es requerido para la construcción del sistema y no hay errores que afecten al diseño”. La corrección depende del usuario final del sistema, que es el indicado para decidir si la especificación es correcta o no.
- Especificación no-ambigua: un requisito puede ser interpretado de formas diversas por diferentes personas. El estándar IEEE 830 establece que un requisito es no-ambiguo “si y sólo si, puede estar sujeto a una única interpretación.”
- Especificación completa: una especificación es completa “si y sólo si, describe todos los requisitos relevantes para el usuario, incluyendo requisitos asociados con funcionalidad, rendimiento, restricciones de diseño, hardware o interfaces externas.”
- Especificación consistente: una especificación es consistente “si y sólo si, no hay ningún subconjunto de requisitos descrito dentro de ella que esté en conflicto con cualquier otro.” El mayor problema en este ámbito es la repetición de información en los requisitos (requisitos repetitivos y redundantes).
- Especificación verificable: una especificación es verificable “si lo son cada uno de los requisitos constituyentes”. A su vez, se considera que un requisito individual es verificable “si existe un proceso acotado (en plazo y presupuesto) que permita determinar que el sistema construido satisface lo descrito en el propio requisito”. La descripción detallada y prueba de los requisitos una vez implementados ayudan considerablemente a su verificación.

- Especificación trazable: una especificación se considera trazable si el origen de cada requisito individual está claro y existe algún mecanismo que permita rastrear la incorporación de dicho requisito en el resto de las actividades del ciclo de vida.
- Requisitos ordenados por importancia y/o estabilidad: permite establecer prioridades a la hora de abordar el desarrollo, contribuyendo al cumplimiento de los presupuestos y tiempo de entrega del producto.

1.10 La IR en simuladores virtuales

En una encuesta realizada por la autora a 10 líderes y/o analistas de proyectos de simulación virtual, sólo el 10% de los encuestados han asignado un nivel bajo de prioridad a los requisitos dentro del proceso de desarrollo de software según sus experiencias en los proyectos que han participado (Ver Figura 1.6).

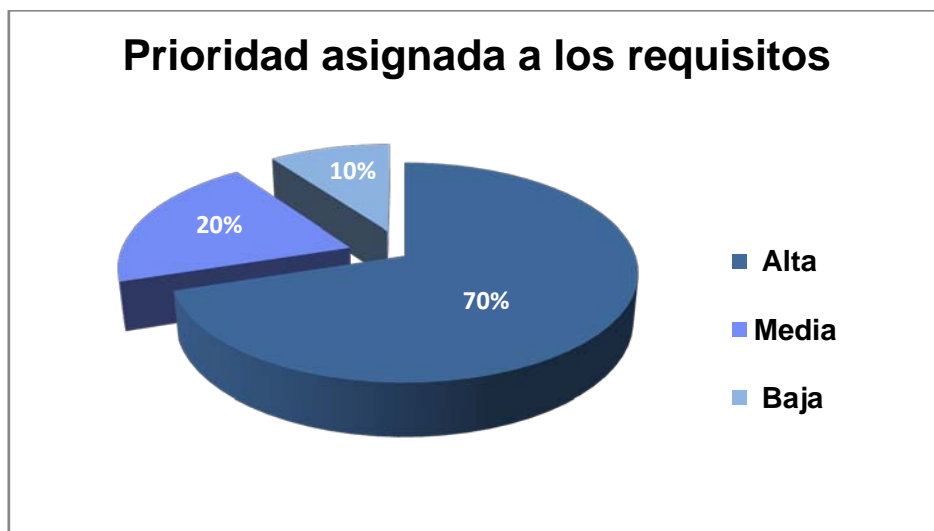


Figura 1.6. Prioridad asignada a los requisitos dentro del proceso de desarrollo de SV.

Sin embargo sólo el 10% de los encuestados asignan alta calificación a la calidad de los requisitos y sus especificaciones dentro del proceso de desarrollo de los SV (Ver Figura 1.7).

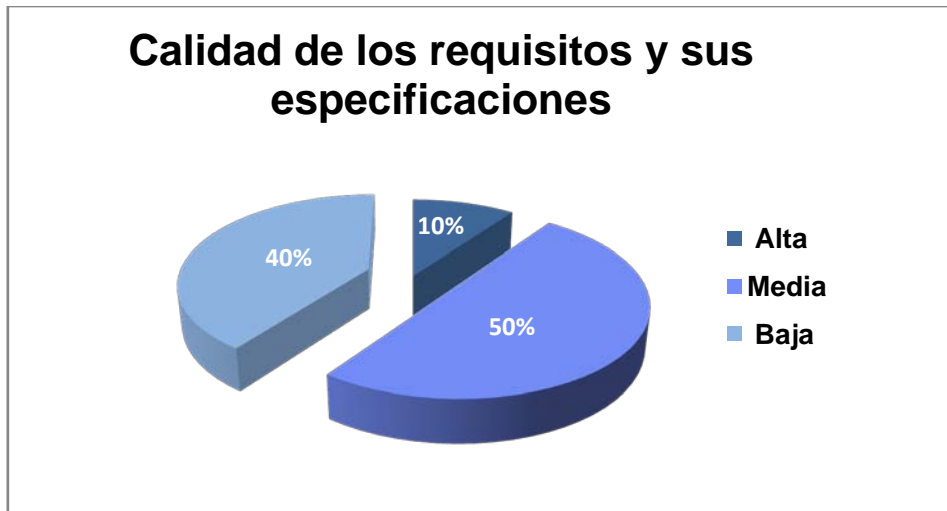


Figura 1.7: Nivel de calidad asignado a los requisitos y sus especificaciones en proyectos de SV.

Estos resultados dejan de ser un tanto paradójicos si se tiene en cuenta que el 50 % de los encuestados manifiestan no desarrollar actividades concernientes a los procesos de análisis, especificación y validación de los requisitos (Ver figura 1.8) y que el 80% de ellos no hace uso de las herramientas existentes para llevar a cabo los procesos de la IR (Ver figura 1.9).

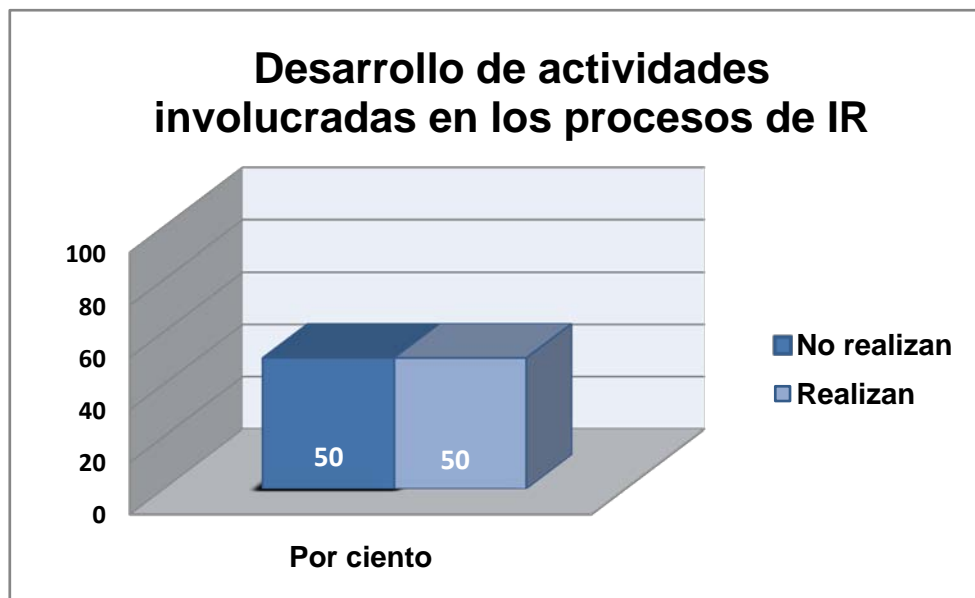


Figura 1.8: Desarrollo de actividades concernientes a los procesos de Análisis, Especificación y Validación de requisitos de la IR en proyectos de SV.

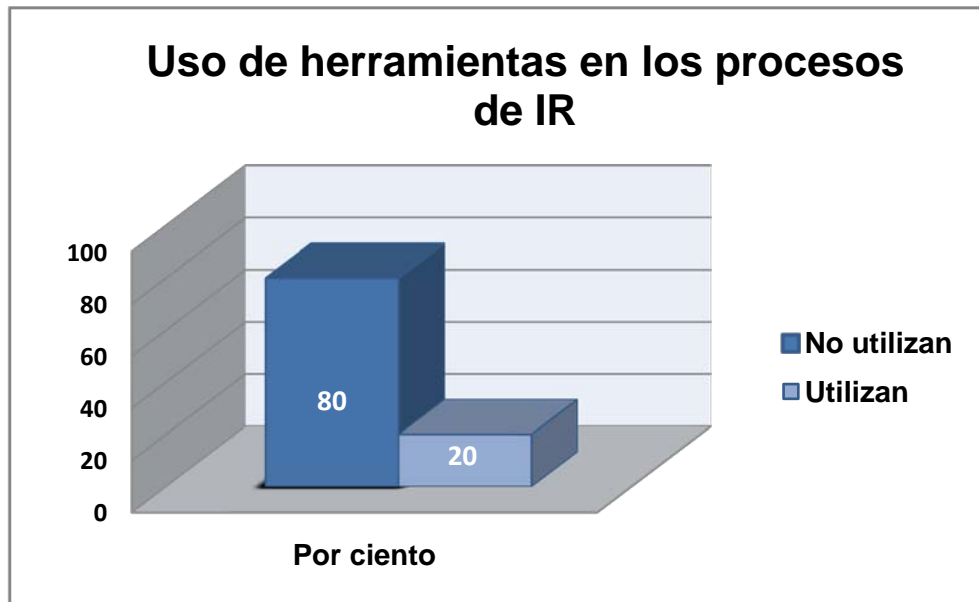


Figura 1.9: Uso de herramientas en los procesos de IR en proyectos de SV.

Consideraciones del capítulo

- Existe abundante literatura científica que definen procedimientos, técnicas, lenguajes y herramientas para la IR; sin embargo persisten dificultades con la calidad final de los proyectos. La insatisfacción de los clientes y los incumplimientos con los cronogramas y presupuestos en muchos casos son provocados por deficiencias en el desarrollo y gestión de requisitos.
- En las múltiples fuentes bibliográficas consultadas no se encuentra referencia de aplicación o adaptación de las técnicas y herramientas de los procesos de la IR en proyectos de simulación virtual.
- La mayoría de los proyectos de simulación virtual del país le asignan una alta prioridad a los requisitos; sin embargo la calidad del trabajo en esta área en muy pocos casos es alta. La no utilización de técnicas y herramientas para desarrollar los procesos constituyen factores claves en esta paradoja.

Las consideraciones anteriores permiten inferir la necesidad de definir las técnicas y herramientas para llevar a cabo los procesos de la IR adecuadas para proyectos de simulación virtual que posibiliten implantar esta disciplina en centros de producción de tales sistemas, específicamente en el polo de realidad virtual de la uci.

Capítulo 2: Técnicas y herramientas de la IR para SV.

En este capítulo se fundamentan las características que presentan los SV que suponen un impacto en la calidad de los procesos de IR. Se formaliza la propuesta de los procesos o etapas para la IR en SV a partir del estudio bibliográfico realizado y se propone un conjunto de técnicas adecuadas a utilizar en cada etapa. De igual forma se recomiendan las herramientas software que soporten, en la mayor medida posible, las técnicas propuestas.

2.1 Características de los Simuladores Virtuales

Desde los inicios del presente trabajo se ha hecho referencia a las definiciones de sistemas de realidad virtual y de simuladores virtuales en específico. De manera general los SV se basan en un entorno virtual basado en técnicas de realidad virtual que reproducen el ambiente de trabajo en las diferentes máquinas simuladas y en un conjunto de modelos lógicos y matemáticos que logran que el comportamiento del entorno virtual sea similar al del entorno y la máquina real. En aras de aportar claridad al entendimiento de estos conceptos y al funcionamiento de un SV, se ilustra en la (Figura 2.1) la representación de un modelo genérico de un ambiente virtual (51).

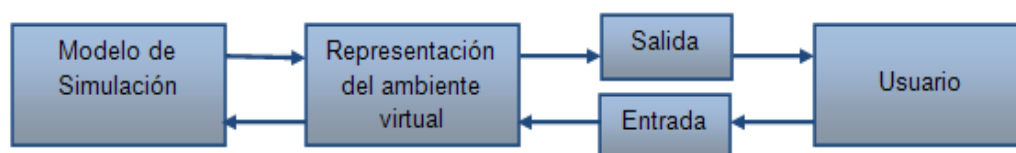


Figura 2.1: Modelo genérico de un ambiente virtual (51).

- Modelo de simulación: es la representación matemática del sistema que se está presentando. Un modelo virtual necesita responder dinámicamente en respuesta de la entrada del usuario. Se pueden crear modelos matemáticos sofisticados de realidad virtual, pero lo importante es la manera en que estos modelos están asociados con un sistema visual y auditivo.
- Representación del ambiente virtual: representación tridimensional del entorno real a simular a partir del cual el usuario establece la comunicación con el sistema.

- Entrada: los dispositivos de entrada se utilizan para interactuar con el ambiente virtual como puede ser el teclado o el mouse, un joystick o guantes, pedales, etc.
- Salida: la salida se refiere a la tecnología mediante la cual el usuario percibe estímulos. Esta tecnología abarca un amplio rango e incluye aparatos de despliegue visual, sistemas sonoros, sistemas sensoriales, entre otros.
- Usuario: el usuario es la razón de existir de un sistema de realidad virtual, pues es él quien recibe los estímulos por parte del sistema y a su vez se encarga de retroalimentarlo y definir su comportamiento.

En la investigación se ha concretado, a partir de criterios de varios profesionales con marcada experiencia obtenida de la participación en el desarrollo de SV, además del estudio y análisis de diferentes fuentes bibliográficas (51), (52), (53), una serie de características comunes a los proyectos de simulación virtual que tienen impacto en los procesos inherentes a la IR. A continuación son señaladas y argumentadas:

1. Relaciones estrechas de trabajo entre disímiles campos de las ciencias: los proyectos de desarrollo de simuladores virtuales se caracterizan además por las diferencias en cuanto a lenguaje, formación y visión entre clientes y equipo de desarrollo. Los ingenieros de software deben obtener y modelar la información, considerable cantidad de conocimiento especializado que los profesionales de múltiples ramas como la medicina, la aviación, la gerencia empresarial, la conducción automovilística, etc. han obtenido a lo largo de su desempeño profesional e investigativo. Cada profesional recibe años de preparación en su materia, haciendo suyos conceptos respaldados por un elevado nivel de conocimientos y en ocasiones suponen como trivial su comprensión por parte del interlocutor. Este hecho desfavorece el entendimiento mutuo entre clientes y equipo de desarrollo.
2. Comportamiento en tiempo real de la simulación manteniendo un nivel aceptable de visualización y realismo: una de las principales y más complejas características en este tipo de sistemas es lograr que la interacción, navegación y visualización se establezca en un tiempo real. Ello representa que el sistema en milésimas de segundos, no superior a 100 msec, debe responder a las acciones ejecutadas y la visualización de las imágenes debe ser a una frecuencia no menor de 20 HZ. Para mostrar estos resultados es valorada las prestaciones del procesador donde se establezca el software, las cuales se pueden mejorar con tarjetas de video, pero

igualmente la calidad de la representación del modelo tridimensional influye en la disminución de la velocidad de la animación.

3. Modelación de objetos con apariencia y propiedades físicas adicionales y en relación con sus funciones y comportamiento: los simuladores incluyen la modelación de entornos reales con disimiles características visuales. Para tal modelación se deben representar las formas tridimensionales de los objetos, especificando sus colores reales según su correspondencia en el mundo real e incorporar efectos que aporten una mayor definición al entorno simulado. A partir de la modelación obtenida se simula el comportamiento real de los elementos que componen el entorno. Tal simulación puede ser alcanzada desde las básicas transformaciones geométricas de traslación, rotación o escalación hasta la simulación de comportamientos tan complejos y diversos como el movimiento de un carro, de una bala o la deformación sufrida por un órgano al ser cortado. Para lograr dichos efectos se requiere modelar correctamente y de manera eficiente las características físicas de los objetos y establecer modelos matemáticos que rijan su comportamiento.
4. Consideración de diferentes estilos y modalidades de técnicas de interacción acorde a las diferentes tareas y dispositivos de entrada/salida: los simuladores virtuales se caracterizan por establecer una comunicación de entrada /salida con diversos equipos externos según las necesidades y formas de interacción dado el tipo de simulador. A través de estos dispositivos de entrada y salida de datos se facilita la comunicación hombre-máquina. Con los dispositivos de entrada (*input*) el usuario puede transmitir sus órdenes al simulador, indicándole sus preferencias - desea desplazarse, cambiar el punto de vista o interactuar con algún objeto del mundo virtual-. Los dispositivos de salida (*output*) permiten que el usuario se sienta inmerso en el mundo virtual creado. Para hacer efectiva las órdenes ejecutadas por los usuarios, en el sistema se debe hacer corresponder la manipulación de los instrumentos reales con sus análogos en el mundo virtual.

2.2 Propuesta de procesos o etapas para la IR en SV

Los resultados de estudios (lista de chequeo de requerimiento y encuesta) realizados a proyectos de SV y presentados en acápite anteriores de este trabajo evidenciaban que las técnicas y/o herramientas usadas en la IR para este tipo de proyectos influían

negativamente en la calidad de las especificaciones de requisitos como producto de trabajo de esta disciplina.

Teniendo en cuenta que en la literatura no se reportan referencias de adaptaciones de la IR a las necesidades de los SV y considerando las características propias del proceso de desarrollo de los SV expuestas anteriormente, se propone en esta investigación las técnicas y herramientas adecuadas para la IR con el objetivo de ser aplicadas en este tipo de proyectos en aras de mejorar la calidad del producto resultante de los requisitos y por tanto de los restantes procesos ingenieriles subsecuentes. En la (Figura 2.2) se representa conceptualmente la estructura básica de la propuesta.



Figura 2.2: Estructura básica de la propuesta.

- Técnicas: conjunto de técnicas de la ingeniería de requisitos adecuadas para los proyectos de simulación virtual con el objetivo de mejorar la calidad del producto de trabajo de la IR.
- Herramientas: selección de herramientas que permiten automatizar o semi-automatizar los procesos de IR mediante el soporte a las técnicas recomendadas.

En el capítulo 1 se referenciaban las metas de la IR. Se trataba como un mecanismo apropiado para comprender a los involucrados, analizar sus necesidades, especificar la solución sin ambigüedades, validar la especificación, y administrar los requisitos conforme estos se transforman en un sistema operacional (9). Diversos autores (30), (35) coincidían en la propuesta de cinco procesos o etapas comprendidas en la IR:

- Obtención o Elicitación de requisitos (OR).
- Análisis de requisitos (AR).
- Especificación de requisitos (ER).
- Validación de requisitos (VR).
- Gestión de requisitos (GR)

Los autores consultados también coincidían en agrupar las cuatro primeras etapas en la categoría Desarrollo de Requisitos (DR) debido a que estos procesos están enfocados a establecer lo que debe hacer el sistema y el proceso de gestión de requisitos va dirigido a evolucionar los requisitos desarrollados conforme evoluciona el proyecto. Sobre la base del trabajo de estos autores y las normativas de la IEEE, CMMI, e ISO (10), (33), (34) se tendrán en cuenta dichas etapas o procesos de la IR para formalizar la propuesta de esta investigación, sustentada básicamente por la definición de las técnicas y herramientas adecuadas para desarrollar dichos procesos en proyectos de simulación virtual, como se explicó anteriormente y se ilustra en la (Figura 2.3).

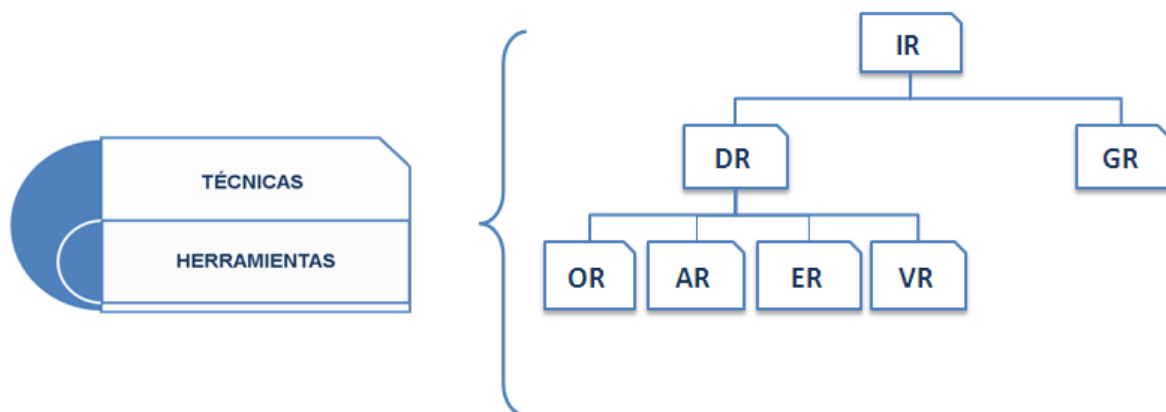


Figura 2.3: Procesos de la IR involucrados en la propuesta.

2.3 Técnicas en cada etapa o proceso de la ingeniería de requisitos

En la literatura se disponen numerosas técnicas aplicables a distintas etapas de la IR. Generalmente sucede que ninguna de ellas funciona de manera independiente ni en todos los tipos de proyectos. En esta investigación se proponen un conjunto de técnicas, tomando en cuenta las características de los simuladores virtuales y las actividades involucradas en los procesos de de IR definidos en (30) y citados

anteriormente. El empleo de estas técnicas recomendadas sugiere una mejora en la calidad del producto de trabajo de la IR. Como se muestra en la (Figura 2.4) las técnicas se clasifican o agrupan según los procesos.

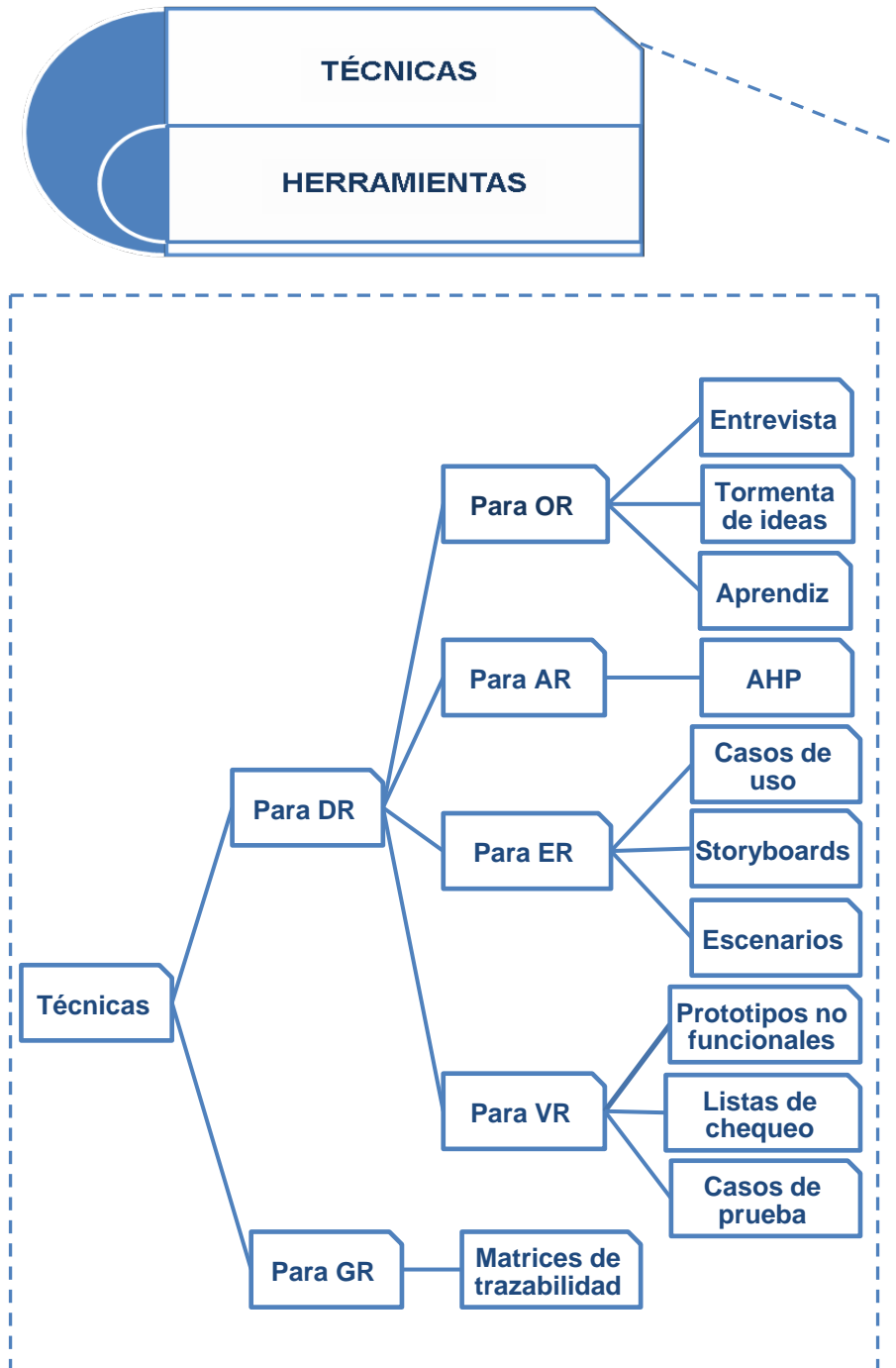


Figura 2.4: Técnicas propuestas en cada proceso de la ingeniería de requisitos.

- Técnicas para la OR: se incluyen en esta categoría la entrevista, tormenta de ideas y la técnica del aprendiz.
- Técnicas para el AR: se propone la priorización de requisitos como técnica a utilizar en esta etapa, la cual será puesta en práctica mediante el método de jerarquía analítica, conocido como AHP (*Analytical Hierarchy Process*) por sus siglas en inglés.
- Técnicas para la ER: se propone en esta etapa el uso combinado de escenarios con storyboards, además de la conocida técnica de casos de uso.
- Técnicas para la VR: se incluyen en esta categoría la técnica construcción de prototipos combinado con el uso de listas de chequeo, además del empleo de la técnica generación de casos de pruebas.
- Técnicas para la GR: se recomienda en esta fase el empleo de las matrices de trazabilidad como técnica para gestionar los cambios que puedan surgir en los requisitos, la cual se definirá en función de las técnicas utilizadas en los procesos de ER y VR, obteniéndose finalmente un conjunto de matrices de trazabilidad (matriz de requisitos vs casos de uso, matriz de requisitos vs escenarios, matriz de requisitos vs prototipos y matriz de requisitos vs casos de prueba).

2.4 Técnicas para la obtención de requisitos

En (7) Pressman cataloga el proceso de obtención de requisitos como uno de los procesos más importantes y complejos de la disciplina en cuestión. Su objetivo se enmarca en obtener del cliente la definición de sus necesidades y para lograrlo es preciso ayudar al cliente a definir sus necesidades, establecer cuáles son los objetivos para el sistema o producto, qué es lo que se debe lograr, de qué forma el producto satisface las necesidades del negocio y cómo se utilizará el sistema final.

El proceso se hace más engorroso en el caso de los proyectos de simulación virtual debido a la característica 1 definida en el epígrafe 2.1 del presente capítulo. En aras de disminuir tal complejidad se recomienda el uso de algunas técnicas como la entrevista, la tormenta de ideas y la técnica del aprendiz para recopilar información. A continuación se describen con más detalle.

2.4.1 Entrevista

Las entrevistas constituyen la técnica de obtención más utilizada, son prácticamente inevitables en cualquier desarrollo al presentarse como una de las formas de comunicación más naturales entre personas (28).

En esta propuesta de entrevista se identifican tres fases iniciales: preparación, realización y análisis.

Preparación de la entrevista

Las entrevistas no deben improvisarse, por lo que conviene realizar las siguientes tareas previas:

- Estudiar el dominio del problema
- Seleccionar a la(s) persona(s) a entrevistar
- Determinar el objetivo y contenido de la entrevista
- Planificar la entrevista

Realización de la entrevista

En la realización de la entrevista se distinguen tres etapas:

- **Apertura:** se realiza la presentación formal del entrevistador y se enuncian los objetivos de la entrevista, especificándose alguna notación particular necesaria a utilizar; procurando siempre de lograr la mejor impresión al entrevistado en los minutos iniciales.
- **Desarrollo:** el tiempo de duración no debe exceder las dos horas, estableciendo una distribución del tiempo de un 20% para el entrevistador y un 80% para el entrevistado. Se deben evitar los monólogos y mantener el control por parte del entrevistador, contemplando la posibilidad de que una tercera persona tome notas durante la entrevista o grabar la entrevista en cinta de vídeo o audio, siempre que el entrevistado esté de acuerdo. Se recomienda utilizar la plantilla inicial propuesta en el (Anexo 3).
- **Terminación:** se resume la información recogida para evitar confusiones, se le agradece al entrevistado la colaboración dejando siempre abierta la posibilidad de un nuevo contacto en caso de que surjan dudas al estudiar la información o al contrastarla con otros entrevistados.

Análisis de la entrevista

Una vez realizada la entrevista es necesario leer y reorganizar la información formalizando un documento memoria de la entrevista. Una vez elaborado se recomienda enviar al entrevistado para confirmar los contenidos. También es importante evaluar la propia entrevista para determinar los aspectos mejorables.

En el (Anexo 3) se propone una plantilla inicial de la primera entrevista. Es recomendable la utilización de una guía para evitar preguntas importantes en el momento de aplicación de la técnica. La plantilla propuesta constituye una guía que estará sujeta a modificaciones en la medida en que se adapte a la situación específica de un proyecto. Se han incluido tres tipos de preguntas. Primeramente se realizan preguntas que conlleven a un entendimiento básico del problema, enfocadas sobre los clientes, objetivos generales y beneficios esperados. Un segundo conjunto de preguntas van dirigidas a obtener una mejor comprensión del problema por parte del entrevistador y conocer las opiniones del entrevistado acerca de la solución. Por último se realizan una serie de preguntas enfocadas a valorar la eficacia de la reunión (7).

2.4.2 Tormenta de ideas

Tormenta de ideas o *brainstorming* es una técnica de reuniones en grupo que tiene como objetivo la generación de ideas en un ambiente libre de críticas o juicios. Las sesiones de brainstorming suelen estar formadas por un número de cuatro a diez participantes, uno de los cuales es el jefe de la sesión. Posibilita formular el problema de diferentes formas o puntos de vistas.

En el brainstorming se distinguen las siguientes fases (28):

- Preparación: se selecciona el jefe de la sesión y los participantes que normalmente son los clientes, usuarios, analista, desarrolladores y si es necesario algún experto en temas relevantes para el proyecto y se define además, el local donde sesionará la reunión.
- Generación: se enuncia el problema de manera general estimulando la generación de ideas libremente por parte de los participantes. El proceso continua hasta que el jefe decide detener la sesión ya sea por tener un número suficiente de ideas para pasar a la siguiente fase o porque no se estén generando suficientes ideas, en cuyo caso se recomienda posponer la reunión.
- Consolidación: en esta fase se debe organizar y evaluar las ideas generadas durante la fase anterior. Básicamente se revisan las ideas generadas para

esclarecerlas, se descartan aquellas ideas que los participantes consideren excesivamente avanzadas priorizándose las ideas restantes e identificando las absolutamente esenciales.

- Documentación: terminada la sesión el jefe produce la documentación oportuna conteniendo las ideas priorizadas y comentarios generados durante la consolidación.

2.4.3 Aprendiz

Después de realizada la primera entrevista y sesión de brainstorming se recomienda aplicar la técnica del aprendiz para garantizar un mayor entendimiento del problema a resolver.

Como se planteaba anteriormente, en la característica 1 del epígrafe 2.1, los proyectos de desarrollo de SV procuran el entendimiento de problemas, en diferentes y múltiples campos de aplicación, que requieren una solución informática. Ello conlleva a una estrecha relación entre los profesionales de la informática, que obtienen y modelan la información, y el cliente del área en cuestión. Generalmente a dichos clientes se les hace difícil explicar completamente su trabajo y más aún si se tiene en cuenta el elevado nivel de detalle y precisión que requiere el desarrollo de SV. En muchas ocasiones se omiten elementos considerados triviales o se emplean mayormente términos desconocidos, lo que dificulta un total entendimiento al emplear técnicas como la entrevista. Haciendo referencia a la técnica del aprendiz, descrita en el capítulo 1 y basada en la observación por parte del ingeniero de software al trabajo realizado por el usuario/cliente, su uso propicia una adecuada solución a la necesidad de lograr un entendimiento de los problemas a simular.

Es habitual además en este tipo de proyectos la necesidad de obtener las habilidades que han desarrollado los usuarios al pasar del tiempo, por lo que no es factible el análisis de documentación existente ya que no se trata de conocimientos explícitos, sino de habilidades prácticas obtenidas con la experiencia. La técnica del aprendiz es muy útil para atenuar estos problemas.

2.5 Técnicas para el análisis de requisitos

El proceso de análisis de requisitos se encarga de elevar el nivel de detalle de los requisitos del sistema. Se definen las prioridades, cuáles aspectos son esenciales y en qué momento se requieren. Se negocia el alcance del proyecto, los requisitos

conflictivos, los inconvenientes relacionados con la prioridad de requisitos y se identifican y analizan los riesgos asociados a los requisitos. Además aporta las bases para la planificación del proyecto debido a que los requisitos analizados son entrada para las estimaciones preliminares del esfuerzo requerido para el desarrollo. De una manera iterativa los requisitos se eliminan, combinan o modifican hasta conseguir alcanzar los objetivos planteados. El análisis proporciona un ciclo de retroalimentación al analista que refina el entendimiento de los requisitos logrado en el proceso de obtención (25), (9).

Poder establecer una priorización de los requisitos en esta etapa constituye sin dudas, un avance significativo para cualquier proyecto software, más aun si se alcanzan resultados confiables de una manera simple y se valora la satisfacción posible de los usuarios y/o stakeholders del proyecto. Obtener un orden adecuado de implementación de los requisitos posibilita una respuesta a los usuarios en sus habituales interrogantes sobre el avance del producto, permite un balance entre el tiempo de entrega y la cantidad de funcionalidades y ayuda a decidir qué requisitos se incluirán en cada entregable (54).

En el capítulo 1 se referenciaban algunos métodos reportados en la literatura para satisfacer este propósito como la técnica de asignación numérica, AHP, árbol de búsqueda binaria, entre otros. El método AHP es flexible y simple de aplicar, origina una salida cuantitativa y además presenta un conjunto de pasos definidos para su implementación. Seguidamente se propone el empleo este método para implementar la técnica de priorización de requisitos en el desarrollo de los proyectos de simulación virtual.

2.5.1 Priorización de los requisitos mediante el método AHP

AHP es un poderoso y flexible proceso que ayuda a tomar la mejor decisión cuando diferentes aspectos necesitan ser considerados. En los años recientes ha incrementado su aplicación en el sector del software, aunque ha sido ampliamente usado en diferentes industrias con anterioridad (55). Una de sus aplicaciones concretas en la rama del software ha sido en la priorización de requisitos basado en su costo y su valor relativo. De esta manera los jefes de proyectos pueden tomar decisiones en relación a qué requisitos pueden ser postergados o excluidos para minimizar el tiempo de entrega del producto. En este sentido, según sus autores, los términos costo y valor son interpretados de la siguiente manera (39):

Costo: el costo de implementar el requisito candidato. En la práctica, el costo se interpreta únicamente en términos económicos.

Valor: el valor es interpretado como la contribución de un requisito a la satisfacción del cliente con el sistema resultante.

Para la puesta en práctica de la técnica se plantea la ejecución de cinco pasos fundamentales. A continuación se describe la propuesta de aplicación de cada paso enmarcada en la priorización de requisitos de un simulador virtual.

1. Revisar los requisitos candidatos. El equipo (ingenieros de requisitos) revisa cuidadosamente los requisitos candidatos para garantizar que estén correctos, completos y que hayan sido expresados de manera no ambigua.
2. Aplicar el método de comparación por pares de requisitos para determinar el valor relativo de cada uno. Los clientes y usuarios (stakeholders por el lado del cliente) implementan el método de comparación de pares de AHP para definir el valor relativo de los requisitos candidatos. Para n requisitos candidatos en análisis el método debe producir una matriz de $n \times n$ valores, de manera que una arbitraria entrada en la fila i y columna j de la matriz, etiquetada como a_{ij} , indican cuán alto o bajo es el valor del requisito i con respecto al requisito j y cada requisito tiene valor 1 cuando es comparado con él mismo (Ver Tabla 2.1).

	RF1	RF2	...	RFn
RF1	1			
RF2		1		
...			1	
RFn				1

Tabla 2.1: Representación matricial de la comparación por pares.

El valor de un requisito es medido mediante una escala de números enteros, desde 1 hasta 9 según la siguiente interpretación (Ver Tabla 2.2).

Intensidad del valor	Interpretación
1	Los requisitos i y j tienen el mismo valor.
3	El valor del requisito i es ligeramente más alto que el valor del requisito j .
5	El valor del requisito i es fuertemente

	más alto que el valor del requisito j .
7	El valor del requisito i es muy fuertemente más alto que el valor del requisito j .
9	El valor del requisito i es absolutamente más alto que el valor del requisito j .
2,4,6,8	Constituyen escalas intermedias entre dos criterios o valoraciones adyacentes.
Recíproco	Si el requisito i tiene un valor inferior al requisito j .

Tabla 2.2: Interpretación de los valores en la matriz.

3. Aplicar el método de comparación por pares para determinar el costo relativo de cada requisito.

Los ingenieros de software (*stakeholders* por el lado del equipo de desarrollo) aplican el sistema de comparación por pares AHP para determinar el costo relativo de los requisitos candidatos. Este proceso es homólogo al ejecutado en el paso anterior para determinar el valor relativo. Se genera una matriz como la representada en la tabla 2.1, pero la escala de valores a seguir es la siguiente (Ver Tabla 2.3).

Intensidad del valor	Interpretación
1	Los requisitos i y j tienen igual costo.
3	El costo del requisito i es ligeramente más alto que el costo del requisito j .
5	El costo del requisito i es fuertemente más alto que el costo del requisito j .
7	El costo del requisito i es muy fuertemente más alto que el costo del requisito j .
9	El costo del requisito i es absolutamente más alto que el costo del requisito j .
2,4,6,8	Constituyen escalas intermedias entre dos criterios o valoraciones adyacentes.
Recíproco	Si el requisito i tiene un costo inferior al

	requisito <i>j</i> .
--	----------------------

Tabla 2.3: Interpretación de los costos de la matriz.

Los participantes rellenan las celdas de las matrices de priorización según la escala anterior demostrando el nivel de interés expresado por cada requisito candidato.

4. Determinar la prioridad de los requisitos.

En este paso los ingenieros en requisitos usan el AHP para calcular el valor y costo relativo de cada requisito y luego lo grafican en un diagrama de costo-valor.

Se parte de las matrices obtenidas en el paso anterior. A cada una se le agrega una columna en la parte final que representa un estimado del vector propio de la matriz, el cual indica la puntuación relativa (valor y costo) a cada requisito finalmente según la matriz (Ver Tabla 2.4).

La estimación del vector propio (columna puntuación) de la matriz se logra promediando los datos sobre columnas normalizadas. Dicha normalización se determina calculando la suma de las columnas de la matriz y luego se divide cada valor en la matriz por la suma de la columna correspondiente. Finalmente, para determinar los puntos relativos, se promedia cada fila de la matriz normalizada y se almacenan en la columna agregada. Esta puntuación representa el por ciento que cada requisito aporta sobre el valor/costo total de los requisitos.

	RF1	RF2	...	RFn	Puntuación
RF1	1				
RF2		1			
...			1		
RFn				1	

Tabla 2.4: Representación matricial con la puntuación incluida.

Seguidamente se procede a representar cada requisito en un diagrama costo-valor como se ilustra en la (Figura 2.5).

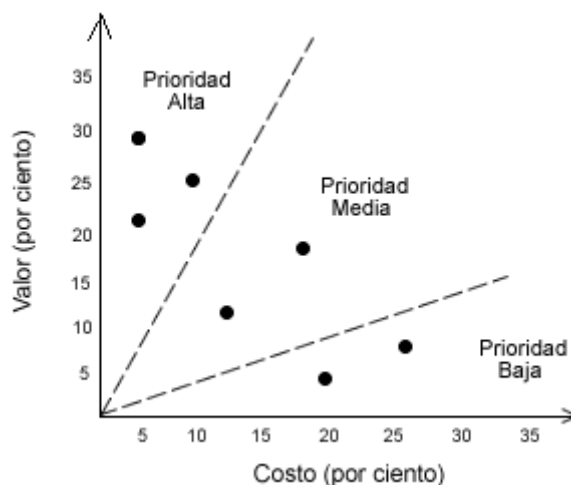


Figura 2.5: Diagrama Costo-Valor de Requisitos.

5. Análisis de los requisitos utilizando el diagrama costo/valor.

Los stakeholders usan este diagrama para el análisis y discusión de los requisitos candidatos. El área de alta prioridad indica los requisitos que ofrecen alta satisfacción para el cliente y tienen un bajo de costo de implementación, por el contrario el área de baja prioridad ubica los requisitos que ofrecen baja satisfacción al usuario y presentan un alto costo de implementación. Basado en el análisis del diagrama se pueden determinar o seleccionar aquellos requisitos que serán implementados, y cuales pudieran ser desechados. Esta información también se puede tener en cuenta en la planificación de estrategias de entregas del sistema, determinando los requisitos que serán implementados en una primera versión del sistema y los que pueden ser pospuestos para versiones subsiguientes.

Consideraciones del método AHP

La utilización del método AHP resulta un proceso directo y no complejo para la priorización de requisitos. Aún así pueden presentarse dificultades en la determinación de los valores o costos de los requisitos. Estos elementos pueden variar dramáticamente según el punto de los stakeholders. Una posible solución a este inconveniente podría ser aplicar el método de manera independiente para cada uno de los usuarios y obtener una matriz de priorización por cada uno. Posteriormente estos resultados son promediados por el equipo de ingenieros en requisitos. Otra variante es involucrar a todos los participantes en el proceso hasta

llegar a un consenso en cuanto a criterios y valores asignados a cada requisito en cada paso necesario. Luego de corregidos estos detalles la priorización alcanza un mayor grado de confiabilidad.

2.6 Técnicas para la especificación de requisitos

El proceso de especificación de requisitos persigue la obtención de una detallada descripción de los requisitos que sirva como base para las actividades de ingeniería subsecuentes, además de validar los artefactos obtenidos técnicamente y desde el punto de vista del cliente. Hasta el momento en la mayoría de los proyectos de simulación virtual desarrollados en el país, esta etapa se ha desarrollado haciendo uso únicamente, de la conocida técnica de casos de uso. En algunos de ellos ni siquiera se ha tenido en cuenta esta técnica y se han limitado a una simple especificación por escrito de los requisitos. Esto provoca que en fases posteriores del desarrollo del software se detecten deficiencias derivadas de una ineficiente especificación de requisitos. Con el objetivo de mitigar estos problemas se proponen un conjunto de técnicas que permiten una descripción de los requisitos que facilita una sólida comprensión a los miembros del equipo de desarrollo de las disciplinas ingenieriles subsecuentes.

Para los requisitos que representen interacción de usuarios se recomienda describirlos mediante la técnica de casos de uso con sus respectivos diagramas de actividad, descripción textual y prototipo para cada escenario de interacción.

Para los requisitos que concuerden con las características 2 y 3 de los proyectos de SV descritas en el epígrafe 2.1 se recomienda el uso de la técnica de escenarios combinado con storyboards. La técnica de escenarios permite hacer una representación de los requisitos mediante videos y/o storyboards, alternativa esta que se ajusta a las necesidades descritas; pues su uso permite mostrar de una forma más precisa, algunos requisitos que lleven implícito gran interactividad y que exigen una representación gráfica de las funcionalidades del sistema para lograr un mejor entendimiento de los mismos.

La utilización de las técnicas no es excluyente para cada tipo de requisito. Pueden combinarse dependiendo de las necesidades de cada caso específico. A continuación se describen las técnicas propuestas antes mencionadas.

2.6.1 Casos de uso

Los casos de uso se consideran la técnica fundamental para especificar requisitos que representan interacción con actores. Un caso de uso describe una serie de eventos en los que un actor interactúa con un sistema para lograr un resultado de valor. Proporcionan un mecanismo para ordenar los requisitos, especificando su momento de ejecución. Constituyen además una herramienta fundamental en los procesos de análisis y diseño e implementación, pues ayudan a comprender las necesidades del sistema y la forma en que podría realizarlas y reducen el riesgo de transformar una expresión de los requisitos en una implementación diferente.

En el caso específico de los sistemas de simulación virtual el empleo de esta técnica se propone para representar aquellos requisitos que conllevan una interacción de usuarios. Su especificación estaría dada por una secuencia de acciones por parte del actor y una correspondiente secuencia de respuestas del sistema. La descripción de este tipo de funcionalidades no necesariamente requiere de elementos visuales para su entendimiento, sino que las descripciones textuales y los diagramas de actividades que acompañan esta técnica son suficientes para comprenderlos.

El conjunto de casos de uso de un sistema fuerza su desarrollo a partir de su diseño y unido a los actores y sus relaciones conforman el modelo de casos de uso del sistema. En (29) se proponen una serie de pasos a seguir para la obtención del modelo de casos de uso de un sistema.

1. Identificar y describir los actores
2. Identificar los casos de uso y escribir una breve descripción
3. Identificar las relaciones actor-caso de uso
4. Describir cada caso de uso
5. Refinar los casos de uso

Para la descripción de los casos de uso se suele utilizar una descripción textual, que puede estar acompañada por un diagrama de actividades. Las descripciones contienen un flujo normal de eventos y pueden tener además uno o más flujos alternativos en aras de lograr una mayor precisión y calidad en el sistema a obtener. Esta descripción textual puede estar dividida según los escenarios que pueden presentarse en el proceso, los cuales aportan generalmente un prototipo no funcional de la futura aplicación que también forma parte de la descripción del caso de uso. Todas estas consideraciones acerca de las descripciones textuales se pueden apreciar en la propuesta mostrada en el (Anexo 4).

2.6.2 Storyboards

Los storyboard de manera general, son de fácil manejo, interactivos y proporcionan una rápida revisión de las interfaces de usuario del sistema además de ser fácil de crear y fácil de modificar. En la práctica no existen reglas o limitaciones para su construcción. Las herramientas y técnicas a utilizar pueden ser tan variadas como la imaginación de los miembros del equipo y de los usuarios lo permita. La mayoría de los storyboards se pueden clasificar en tres tipos, dependiendo del modo de interacción con el usuario: pasivos, activos, o interactivos (26).

- Storyboards pasivos: pueden consistir en bocetos, fotografías, capturas de pantalla o presentaciones en PowerPoint. El analista desempeña el papel del sistema trasladando al usuario a través del guión.
- Storyboards activos: intentan lograr que el usuario conozca una aplicación que no ha sido construida. Se activan por una secuencia automática de diapositivas de presentación, una herramienta de animación o una secuencia de comandos grabados o simulados. Este tipo de storyboards proporciona una descripción automatizada de la forma en que el sistema se comporta en el uso de un típico escenario.
- Storyboards Interactivo: permiten la experiencia del usuario en el sistema como una forma realista de la práctica, por lo que requieren la participación del usuario. Estos storyboards puede ser simulaciones o maquetas o puede ser tan avanzado hasta el punto de requerir líneas de código y pueden construirse con una variedad de paquetes de software especiales para prototipos interactivos y herramientas tales como Macromedia Director.

La flexibilidad e interactividad de esta técnica resultan de gran utilidad en el desarrollo de simuladores virtuales, especialmente en la especificación de aquellos requisitos que, para su total entendimiento, requieran representación de entornos y de objetos con apariencia y propiedades físicas adicionales, como se explicó en el epígrafe 2.1 de este capítulo. Características visuales como estas son difíciles de mostrar mediante el empleo de otras técnicas de especificación que se limitan solamente a una descripción textual de sus funcionalidades como por ejemplo, la conocida y empleada técnica de casos de uso.

El empleo de storyboards en los SV se recomienda en alguna de sus tres categorías, según el grado de complejidad de los requerimientos o funcionalidades que se representen en el escenario que se esté referenciando. Para implementar el desarrollo

de esta técnica se recomienda utilizar representaciones de imágenes y grabaciones de videos que pudieran ser capturadas durante el desarrollo de la técnica del aprendizaje propuesta para la obtención de requisitos. Además de estos recursos, en el epígrafe 2.9.1 se proponen herramientas que también pueden utilizarse en la construcción de esta técnica, sobre todo para desarrollar aquellos storyboards que requieren visualizar interactividad entre los componentes de una escena o como variante alternativa a la captura de imágenes y videos.

2.6.3 Escenarios

La técnica de escenarios consiste en describir las características del sistema a desarrollar mediante una secuencia de pasos y su representación puede ser casi textual o ir encaminada hacia una representación gráfica en forma de diagramas de flujo. Existen varios estilos para construir escenarios, ya sea la narrativa textual, los storyboards, videos o los prototipos escritos (41).

El modelo de escenario que se utiliza comúnmente posee una estructura compuesta por las siguientes entidades: Título, Objetivo, Precondiciones, Actores, Episodios y Excepciones, Poscondiciones y el atributo Restricción. Un escenario, identificado por un título, debe satisfacer un objetivo que se alcanza mediante la ejecución de los episodios. Los episodios representan el curso de acción principal, pero incluyen también variaciones o alternativas posibles y mientras se ejecutan pueden surgir excepciones, que señalan una limitación para lograr el objetivo. Las precondiciones y poscondiciones se corresponden con una breve descripción del estado del sistema antes y después de ejecutado el escenario respectivamente. El atributo restricción es usado para caracterizar requisitos no funcionales aplicados al contexto, recursos y episodios (56).

El empleo de esta técnica en los proyectos de simulación virtual se propone como un uso conjunto con la técnica de storyboards, descrita anteriormente. Su utilización permitirá especificar aquellos requisitos que presenten una mayor interactividad y que no estén guiados específicamente por acciones del usuario, sino por acciones principales del sistema que forman un episodio que podrá visualizarse mediante el storyboard comprendido en el escenario.

Para la conformación de los escenarios se hará uso de una plantilla definida en este trabajo (Anexo 5), tomando en cuenta los elementos más importantes presentados en (41), donde se reserva un espacio que permite mostrar los videos y/o storyboard que describan los requisitos involucrados dentro de cada escenario.

2.7 Técnicas para la validación de requisitos

Es de vital importancia asegurar que las declaraciones de los requisitos son las correctas, demostrar la calidad de sus especificaciones y la satisfacción por parte del cliente. Precisamente son estos los objetivos que se persiguen en el proceso de VR, y así lo reconoce Pressman en (9) cuando se refiere a la validación de requisitos como un mecanismo que examina las especificaciones para asegurar que todos los requisitos del sistema han sido establecidos sin ambigüedad, sin inconsistencias, sin omisiones, que los errores detectados hayan sido corregidos, y que el resultado del trabajo se ajusta a los estándares establecidos para el proceso, el proyecto y el producto.

En ocasiones requerimientos que parecen correctos en la especificación podrían presentar problemas cuando los desarrolladores intentan implementarlos. Ambigüedades o imprecisiones como estas son reveladas a menudo por los casos de prueba. Cuestiones de esta índole deben ser corregidas en una etapa o proceso de validación, de manera que los requisitos puedan aportar una base fiable para las disciplinas subsecuentes del proceso de desarrollo de software. Esta etapa de validación no es un solo proceso que se realiza después de la recopilación y documentación de todos los requisitos, algunas actividades de validación se involucran iterativamente en los procesos de obtención, análisis y especificación de requisitos. A continuación se mencionan algunas actividades propuestas en el proceso de validación (30).

- Revisión de requerimientos: constituye una potente técnica para la identificación de requerimientos ambiguos o no verificables, así como para identificar los requisitos que no están definidos de manera suficientemente clara para comenzar el diseño.
- Prueba de los requisitos: se basa en derivar casos de pruebas funcionales a partir de los requisitos para documentar el comportamiento esperado del producto en determinadas condiciones.
- Definir criterios de aceptación: permite conocer los criterios de los usuarios mediante pruebas de aceptación en función de determinar si el producto satisface sus necesidades y se considera apto para su uso.

En función de lograr los objetivos perseguidos en el proceso de VR, así como de llevar a cabo las actividades involucradas, se propone en esta investigación el uso de tres

técnicas de validación referenciadas en la literatura: las técnicas de prototipado y listas de chequeo de manera combinada y la técnica de generación de casos de prueba.

2.7.1 Construcción de prototipos

Como se había especificado en acápites anteriores del presente trabajo, los prototipos constituyen un modelo del software a fabricar que se construyen con el objetivo de ser validados por el cliente y el desarrollador.

Varios son los paradigmas en la creación de prototipos que se han valorado por diferentes autores en la materia, aunque sus diferencias no son relevantes. En (6) se han identificado dos tipos de prototipos: los prototipos rápidos, que se basan en facilitar la evaluación de los requerimientos en la etapa previa al diseño general del sistema y los prototipos evolutivos, que sufren modificaciones y mejoras a medida que se avanza en el ciclo de vida del proyecto con el fin de llegar hasta el producto final. En (9), Pressman también certifica este tipo de prototipo evolutivo a partir del enfoque de creación de prototipo abierto y corrobora además en estos casos su uso como primera parte de una actividad de análisis a la que seguirá el diseño y la construcción. Otro paradigma manejado en (9) es el de creación cerrado, o a menudo denominado prototipo desechable, que sirve únicamente como una vasta demostración de los requisitos, luego se desecha y se procede al desarrollo del sistema de una manera diferente.

Según (9) se pueden definir varios factores candidatos a la creación de prototipos: área de aplicación, complejidad, características del cliente y del proyecto. En general, cualquier aplicación que cree pantallas visuales dinámicas, interactúe intensamente con el ser humano o demande algoritmos o procesamiento de combinaciones que deban crearse de manera progresiva, es un buen candidato para la creación de un prototipo.

Para materializar la propuesta de confección de prototipos en la etapa de validación de requisitos durante el proceso de desarrollo de simuladores virtuales se propone su construcción a partir de los resultados obtenidos durante el proceso de especificación de requisitos propuesto en acápites anteriores de este documento. Para representar las funcionalidades que fueron especificadas mediante la técnica de casos de uso se procedería a diseñar las pantallas o los prototipos en total correspondencia con lo detallado en la descripción textual del caso de uso teniendo en cuenta tanto el curso normal como el curso alterno de eventos, concretándose así una representación

gráfica de lo descrito en un lenguaje natural. En el caso de la representación de las funcionalidades que le precede una especificación mediante imágenes y videos, confeccionados como resultado de la aplicación de la técnica de escenarios combinado con storyboards, el diseño de los prototipos se recomienda utilizando esta representación precisamente; de manera que se evidencie la interactividad o el dinamismo de la funcionalidad a mostrar.

2.7.2 Listas de chequeo

Como se referenciaba en el epígrafe 1.7.3 las listas de chequeo constituyen una estrategia de fácil manejo para validar requerimientos. Básicamente consiste en un conjunto de interrogantes a utilizar en función de la evaluación de cada requerimiento, verificando los elementos de la lista mientras se chequean los requisitos. Como resultado de este proceso se pueden revelar problemas que deben ser registrados para su posterior corrección.

Para llevar a cabo el uso de esta técnica en la validación de los requisitos de los simuladores virtuales se propone su aplicación haciendo uso de la técnica de construcción de prototipos expuesta anteriormente. Primeramente se elaborarían las listas de chequeo correspondientes a cada pantalla o prototipo construido, con elementos enmarcados en la funcionalidad y en la visualización de los requisitos y haciendo uso de ella, conjuntamente con la lista de los requerimientos formalizados en la etapa de análisis, se verificaría su representación en el prototipo no funcional. La ejecución de todo este proceso estaría presenciada por el cliente y/o usuario del futuro sistema.

Un segundo objetivo para emplear esta técnica es el de la validación de la calidad de las especificaciones de requisitos como producto de trabajo de la disciplina IR. Para ello se recomienda utilizar la lista de chequeo propuesta en el Anexo 1 y utilizada para el diagnóstico inicial del presente trabajo, la cual permite chequear la calidad de las especificaciones a partir de los atributos de calidad definidos a inicios de esta investigación.

Con la aplicación combinada de estas técnicas no solo se reduce la posibilidad de obviar alguna funcionalidad o verificación importante en el futuro sistema, sino que se revelan deficiencias relacionadas con detalles de su especificación o representación en función de las expectativas del cliente; evitando de esta manera confusiones en el

entendimiento de clientes y el equipo de desarrollo y extendiendo su validación hacia elementos no funcionales del sistema.

2.7.3 Generación de casos de prueba

Generar casos de prueba, aunque no reportada en la literatura con el término específico de técnica, sí hay varias fuentes (57), (58) que abalan su uso, sobre todo en el desarrollo de SRV, como una eficiente estrategia a utilizar en la validación de requisitos a partir de casos de uso y/o escenarios.

En correspondencia con la bibliografía consultada se propone la construcción de, al menos, un caso de prueba por cada requerimiento que haya sido especificado; de manera que cada requisito referencie un caso de prueba. Aunque la base de los casos de prueba sea el curso básico de los eventos, ese recomienda desarrollar al menos un curso excepcional de eventos para cada curso básico.

A continuación se describe un método para crear casos de prueba funcional a partir de casos de uso y/o escenarios basado en la ejecución de los siguientes pasos:

1. Generar un sistema completo de funcionalidades involucradas en cada uno de los casos de uso o escenarios definidos anteriormente: se identifican todas las combinaciones posibles de caminos de ejecución del caso de uso o escenario. Las funcionalidades suelen ser diferentes si el comportamiento del sistema se puede activar de manera diferente, por lo tanto para la definición de las funcionalidades se debe tener en cuenta tanto el flujo principal como el flujo alterno de eventos.
2. Para cada funcionalidad, identificar al menos un caso de prueba especificando las condiciones para su ejecución: se estudia la descripción del caso de uso o escenario de partida y se extraen las condiciones o variables requeridas para la ejecución de cada funcionalidad, formalizando así el caso de prueba correspondiente a cada funcionalidad incluida.
3. Para cada caso de prueba, definir los valores que serán asignados a cada variable para la ejecución de la prueba: luego de identificado los casos de prueba se deben validar para asegurar la exactitud e identificar los casos de prueba redundantes o no definidos y se sustituyen los valores reales de los datos a ejecutar.

Luego de ejecutado los casos de prueba se procede a identificar los problemas detectados o no conformidades que serían procesadas posteriormente con el objetivo de ser enmendados antes de pasar a nuevas etapas del desarrollo del software. Teniendo en cuenta el propósito del uso de esta técnica, basado en validar la

especificación de requerimientos realizada, las no conformidades detectadas estarían relacionadas básicamente con errores de incompatibilidad respecto a lo pactado con el cliente. En el (Anexo 6) se propone una plantilla a utilizar como guía en la definición de los casos de prueba.

2.8 Técnicas para la gestión de requisitos

La gestión de requisitos tiene como objetivo dar seguimiento a los requisitos y controlar sus cambios durante todo el ciclo de vida del software. Este proceso comienza cuando el equipo determina que un conjunto de requisitos tienen la calidad suficiente como para comenzar el diseño y codificación. En este momento se establece una línea base que contiene los requisitos aprobados para la liberación específica del producto. Sobre esta base se establecen las estimaciones y planificación del desarrollo subsecuente, por tanto los cambios a la línea base de requisitos deben seguir un procedimiento establecido de control de cambios (25).

Es importante el seguimiento de los requisitos a través de todo el proceso de desarrollo para asegurar la calidad del software y para lograr tal propósito se recomienda, como técnica a utilizar en este proceso, la trazabilidad de los requisitos haciendo uso de las matrices de trazabilidad.

2.8.1 Trazabilidad de requisitos

Seguidamente se describe un conjunto de pasos que guiarán la trazabilidad de los requisitos en los proyectos de simulación virtual.

1. Identificar las matrices de trazabilidad a utilizar. Se recomienda definir:
 - a. Matriz de trazabilidad que muestra la relación entre requisitos y casos de uso para rastrear aquellos requisitos que fueron especificados mediante la técnica de casos de uso.
 - b. Matriz de trazabilidad que muestra la relación entre requisitos y escenarios para rastrear aquellos requisitos que fueron especificados mediante la técnica de escenarios.
 - c. Matriz de trazabilidad que muestra la relación entre requisitos y prototipos para rastrear los requisitos en la etapa de validación mediante los prototipos.
 - d. Matriz de trazabilidad que muestra la relación entre requisitos y casos de prueba para rastrear los requisitos en la etapa de validación mediante los casos de prueba.

Las Tablas 2.5, 2.6, 2.7 y 2.8 constituyen una representación de la estructura que guiará la obtención de la matriz de trazabilidad de requisitos vs casos de uso, matriz de trazabilidad de requisitos vs escenarios, matriz de trazabilidad de requisitos vs prototipos y matriz de trazabilidad de requisitos vs casos de prueba respectivamente.

De manera general, las matrices referenciadas se basan en el rastreo de los elementos relacionados con las técnicas propuestas en los anteriores procesos de la ingeniería de requisitos, enmarcando la propuesta solamente a esta disciplina. Aún así es importante resaltar que la gestión de requisitos es un proceso efectivo durante todo el ciclo de vida del proyecto y por tanto la trazabilidad de los requisitos trasciende las fronteras de la ingeniería de requisitos. En este sentido se debe tener en cuenta la definición de otras matrices de trazabilidad que estén en consecuencia con los artefactos generados o técnicas aplicadas en las demás etapas o procesos del desarrollo del simulador como los son las etapas de diseño, implementación, validación o prueba y etapa de simulación o despliegue según se define en (52).

2. Almacenar los datos para conformar las matrices identificadas. Se propone hacer uso de la herramienta de gestión de requisitos RequisitePro, que se propone en el epígrafe 2.9.2 del presente capítulo para la ejecución de esta actividad.
3. Identificar las partes del producto para las cuales se desea mantener la información de trazabilidad. Se recomienda iniciar por las funciones básicas más críticas o por aquellas áreas que se consideren necesiten mayor mantenimiento y evolución durante el ciclo de vida del producto.
4. Incluir en el proceso de desarrollo la actualización por parte de los desarrolladores de los vínculos después de la implementación de un requisito o un cambio aprobado. La trazabilidad de los datos debe ser actualizada en el momento de culminación de una tarea mediante la creación o el cambio de un vínculo en la cadena de requisitos.
5. Identificar al personal responsable de coordinar las actividades de la trazabilidad y administración de los datos.
6. Planificar auditoría de trazabilidad periódicamente para asegurar su actualización. Es importante la sistematicidad de la actualización de la trazabilidad y no limitarla sólo al momento de cumplimiento de un hito importante o fin del proyecto.

Requisito Funcional	Caso de Uso
----------------------------	--------------------

	CU1	CU2	...	CUn
RF1				
RF2				
...				
RFn				

Tabla 2.5: Matriz de trazabilidad que muestra la relación entre requisitos y casos de uso.

Requisito Funcional	Escenario			
	E1	E2	...	En
RF1				
RF2				
...				
RFn				

Tabla 2.6: Matriz de trazabilidad que muestra la relación entre requisitos y escenarios.

Requisito Funcional	Prototipo			
	P1	P2	...	Pn
RF1				
RF2				
...				
RFn				

Tabla 2.7: Matriz de trazabilidad que muestra la relación entre requisitos y prototipos.

Requisito Funcional	Casos de prueba			
	CP1	CP2	...	CPn
RF1				
RF2				
...				
RFn				

Tabla 2.8: Matriz de trazabilidad que muestra la relación entre requisitos y casos de prueba.

2.9 Herramientas propuestas para la IR en SV

Múltiples son las razones que evidencian la necesidad e importancia del uso de herramientas software en los procesos de IR que permitan automatizar sus actividades: para mantener los documentos actualizados y sincronizados, para facilitar y agilizar el proceso de almacenamiento de información suplementaria a cada requerimiento, para almacenar los cambios efectuados en los requerimientos durante el desarrollo del proyecto y además, para automatizar el proceso de comunicación de dichos cambios a todos los miembros del equipo implicados. Las herramientas, aunque no constituyen un sustituto para la eficacia de los procesos de desarrollo y las necesidades técnicas, sí impulsan la gestión y seguimiento de la información que se almacena en ellas (25).

Como resultado de esta investigación se recomienda el uso de algunas herramientas durante el desarrollo de los procesos involucrados (Ver Figura 2.6).

Herramientas para el desarrollo de requisitos

- Herramientas CASE: Enterprise Architect (EA).

Además de las herramientas CASE se han incluido un conjunto de herramientas para apoyar el desarrollo de requisitos, específicamente la construcción de storyboards y prototipos, técnicas recomendadas a utilizar en las etapas de especificación y validación de requisitos respectivamente:

- Herramientas para apoyar la construcción de Storyboards.
 - Power Point del paquete Office 2007.
 - Blender.
- Herramientas para apoyar la construcción de prototipos.
 - Motores gráficos
 - Herramientas QT Creator/CEGUI.

Herramientas para gestión de requisitos

- RequisitePro

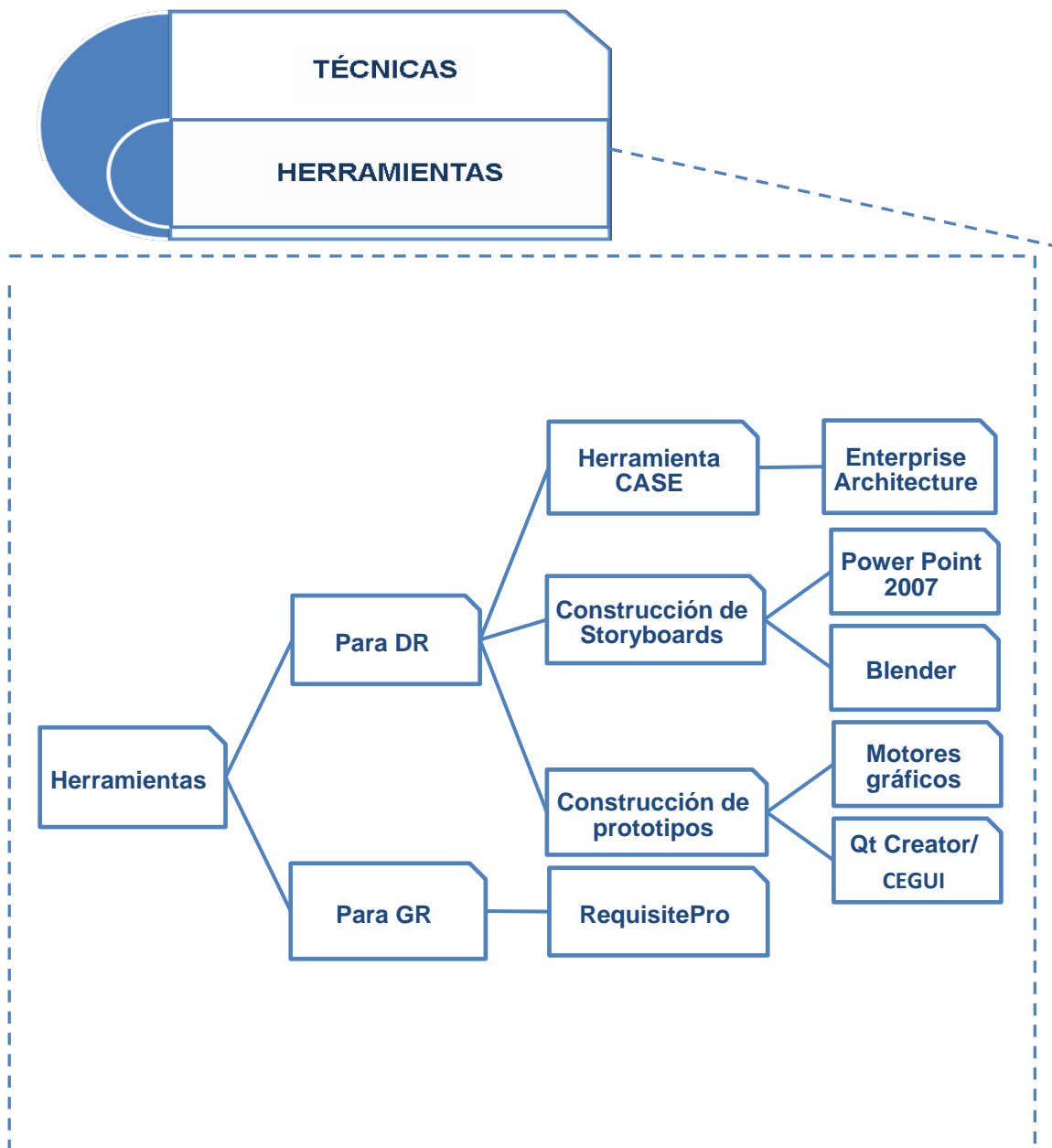


Figura 2.6: Herramientas recomendadas para apoyar la utilización de las técnicas propuestas.

Las diversas opciones de herramientas recomendadas contribuyen a una propuesta abarcadora que posibilita cierto grado de libertad para su selección en el momento de ser ejecutada. Su utilización no está sujeta a restricciones, sino que dependerá de las características de los requisitos a representar y del problema en cuestión.

Reportes actuales

El estudio realizado en (59) señala un enfoque diferente en la creación de sistemas de realidad virtual basado en solucionar los mayores problemas presentados en el desarrollo de ambientes virtuales:

- Necesidad de satisfacer los tres requisitos importantes de un ambiente virtual: presencia, usabilidad y rendimiento.
- Complejidad de los objetos de realidad virtual con sus tres aspectos distintos: forma o apariencia, función y comportamiento.
- La diferencia entre desarrollo y ejecución.

Básicamente la solución de estos problemas está dada por la construcción de herramientas que posibiliten el diseño y la validación inmediata de varios de los elementos de los sistemas de realidad virtual y por ende de los simuladores. PVoT, por sus siglas en inglés (*Portable Virtual reality systems development Tool*) es la herramienta reportada en (59) para satisfacer tal objetivo. Sus principales funcionalidades se basan en el modo de construir y representar la información relativa a la forma, función y comportamiento de los objetos virtuales, incluidos las configuración de dispositivos entrada / salida y la facilidad de interacción. La mayoría de actividades de desarrollo del sistema en esta herramienta se realizan mediante la técnica de arrastrar y soltar (*drag-and-drop*) componentes visuales. En <http://pvot2.postech.ac.kr/> se encuentra disponible su versión 20051029, la cual permanece en su etapa de desarrollo y hasta el momento presenta un conjunto reducido de funcionalidades, un número limitado de objetos y escenas de prueba, lo que limita su uso en a la producción de simuladores profesionales que constituyen productos con alto nivel de realismo y en ocasiones simulaciones de alta complejidad. PVoT se ha utilizado como medio de enseñanza para la construcción de SRV con el objetivo de facilitar la comprensión de conceptos esenciales en el desarrollo de estos sistemas (60).

2.9.1 Herramientas para el desarrollo de requisitos

La herramienta CASE de ciclo completo Enterprise Architect constituye la propuesta más utilizada para guiar el desarrollo de simuladores virtuales en el polo de realidad virtual de la UCI (objeto identificado con la propuesta desarrollada en esta investigación). Dicha herramienta además, soporta las actividades de los procesos de desarrollo de requisitos desarrolladas mediante algunas de las técnicas propuestas en este trabajo.

Para el empleo de la técnica de casos de uso recomendada en el proceso ER, Enterprise Architect soporta los diagramas de casos de uso y los diagramas de actividades. Los prototipos no funcionales también pueden ser elaborados en esta herramienta, aunque no con el suficiente nivel de precisión que se necesita para los SV. Es por ello que a continuación se describe la propuesta de un conjunto de herramientas que ayudan a construir prototipos con exigencias más ambiciosas que los obtenidos mediante EA satisfaciendo de esta manera los simuladores que lo requieran.

En el proceso de ER, además de la técnica de casos de uso, se presentó la técnica de storyboards. Su construcción es una de las debilidades de Enterprise Architect y de las herramientas CASE en general. Es por ello la necesidad de recomendar herramientas que apoyen este proceso, las cuales serán descritas a continuación.

- Power Point del paquete Office 2007 para la construcción de los storyboards más simples o los catalogados como pasivos en el epígrafe 2.6.3 de este capítulo.
- Blender para representar las escenas con un nivel adecuado de realismo y visualizar la interactividad entre los componentes de una escena en caso necesario.

Blender es un programa multiplataforma, dedicado especialmente al modelado, animación y creación de gráficos tridimensionales. Actualmente es compatible con todas las versiones de Windows, Mac OS X, Linux, entre otros. La herramienta posee capacidad para una gran variedad de primitivas geométricas, incluyendo curvas, mallas poligonales, vacíos, etc. Presenta también características interactivas para juegos como detección de colisiones, recreaciones dinámicas y lógica. Posibilita además ediciones de audio y video y simulaciones dinámicas para partículas y videos. Aunque su interfaz gráfica no se basa en el sistema clásico de ventanas posee

ventajas importantes como la configuración personalizada de la distribución de los menús y vistas de cámara.

Otra herramienta a valorar para este fin es el 3D Studio Max. Esta herramienta también es uno de los programas de animación 3D más utilizados. Dispone de una sólida capacidad de edición, una arquitectura de plugin y una larga tradición en plataformas Microsoft Windows, lo que constituye en ocasiones una limitación para su uso.

Herramientas para apoyar la construcción de prototipos.

- Motor gráfico 3D (*Graphic Engine 3D*).

El desarrollo de aplicaciones de simulación virtual a menudo está ligado al empleo de bibliotecas de programación o motores gráficos con el objetivo de agilizar su construcción mediante la reutilización de elementos ya creados. Un motor gráfico 3D es un término que hace referencia a una serie de rutinas de programación que permiten el diseño, la creación y la representación de aplicaciones gráficas. Estos motores poseen, o se integran, con componentes para el desarrollo de Interfaces de usuario (GUI), por sus siglas en inglés (*Guide User Interface*) que pueden ser utilizadas para el desarrollo de prototipos base de la aplicación en un corto período. Estas interfaces formando un prototipo general pueden ser reutilizadas posteriormente en el desarrollo del simulador. Las funcionalidades presentadas por estos motores posibilita el desarrollo de prototipos con un nivel mayor de realismo o fidelidad a la aplicación final.

Son varios los motores gráficos reportados en la actualidad. Su utilización está muy relacionada a las potencialidades del mismo y a las características del sistema a construir. No obstante, motores tan populares y potentes como Ogre 3D y Cristal Space constituyen opciones recomendables en la construcción de tales sistemas. Ambos son motores multiplataforma bajo licencia LGPL (*Lesser General Public License*), lo que significa que su uso es gratuito y apenas existen exigencias para su empleo. En el capítulo 3 de este trabajo se ilustra el desarrollo de un prototipo general para un simulador quirúrgico mediante el empleo de un motor gráfico denominado *Scen Toolkit* desarrollado en el polo de realidad virtual de la UCI.

Independientemente de las potencialidades de un motor gráfico, Qt Creator y CEGUI (*Crazy Eddie's GUI*) constituyen variantes alternativas para el desarrollo de prototipos

de interfaces de usuario. Ambas herramientas constituyen bibliotecas que permiten desarrollar interfaces gráficas de usuario de manera ágil.

- Herramienta Qt Creator.

Qt creator es un completo ambiente de desarrollo integrado (IDE) multiplataforma para crear aplicaciones con las bibliotecas Qt. Qt incluye soporte para OpenGL, el estándar multiplataforma para la producción de gráficos 2D y 3D. Posee además, un componente que permite escribir código OpenGL y visualizar una escena 3D.

Una especificación más detallada de las características de Qt Creator se puede encontrar en el sitio de la herramienta publicado en la siguiente dirección:

<http://www.qtsoftware.com>.

- CEGUI

CEGUI es una biblioteca libre y multiplataforma que brinda prestaciones de ventanas y elementos visuales que pueden ser integrados con motores gráficos. La biblioteca está orientada a objetos, implementada en C++ y dirigida a la construcción de una rápida interfaz visual en aplicaciones virtuales.

2.9.2 Herramientas para la gestión de requisitos

El Grupo de Trabajo de la Base de Datos de Herramientas (TDWG) del Consejo Internacional de Ingeniería de Sistemas (INCOSE) realizó una encuesta a los proveedores de herramientas para la gestión de requisitos (61). La herramienta RequisitePro resalta en el conjunto de aquellas que cumplen con una mayor cantidad de funcionalidades según las interrogantes planteadas. Además de ser esta la más ejercitada en el polo de realidad virtual y en la UCI en general.

La herramienta RequisitePro incorpora una poderosa infraestructura de base de datos para facilitar la organización de los requisitos y la trazabilidad, soportando de esta manera el empleo de la técnica de matrices de seguimiento propuesta en el epígrafe 2.8.1 de este capítulo. Sus resultados se basan en mejorar la comunicación y la administración de requisitos incrementando la probabilidad de completar proyectos en tiempo, dentro del presupuesto y por encima de las expectativas. Una relación de las características de la aplicación se puede encontrar en la encuesta de INCOSE publicada en la siguiente dirección:

http://www.paper-review.com/tools/rms/response.php?vendor=OSRMTv1_3#1

Consideraciones del capítulo

- Se definieron cuatro características de los proyectos de Simulación Virtual que suponen un impacto en la IR: diferencias en cuanto a lenguaje, formación y visión entre clientes y equipo de desarrollo, características visuales especiales para simular el comportamiento realista de los objetos en un entorno, interacción con componentes externos de acuerdo a las diferentes acciones e instrumentos de entrada / salida y ejecución en tiempo real de la simulación manteniendo un nivel aceptable de visualización y realismo del ambiente simulado.
- Se propusieron técnicas y herramientas que posibilitan adaptar la Ingeniería de Requisitos a las características de los proyectos de Simulación Virtual.
- Se presentaron herramientas que permiten llevar a cabo un eficiente desarrollo y gestión de los requisitos.

Capítulo 3: Validación y aplicación práctica de la propuesta.

El objetivo de este capítulo está basado en determinar la validez y objetividad de la propuesta desarrollada. Teniendo en cuenta el déficit de datos históricos que permitan un análisis comparativo se aplicarán técnicas que permiten emplear el conocimiento tácito que poseen los expertos del tema para lograr una evaluación de la propuesta. Mediante este procedimiento se obtendrá una validación acerca de la mejora en el cumplimiento de los objetivos de los procesos de la IR y viabiliza la ejecución de una aplicación práctica de las técnicas y herramientas propuestas en el proyecto simulador quirúrgico que se desarrolla en el PRV de la UCI. Los resultados de dicho experimento forman parte también en este capítulo.

3.1 Validación del modelo propuesto por un método teórico.

Una propuesta como la desarrollada en esta investigación puede ser validada a partir de los resultados de su aplicación práctica en proyectos de SV, estableciéndose una comparación con los datos históricos relacionados con la ejecución de dichos proyectos. Generalmente, y así se afirma en (62), este tipo de información no es recopilada de forma explícita en los proyectos, por lo que faltaría información histórica para establecer las comparaciones. En (62) se revela además el uso predominante, sobre todo en investigaciones asociadas a la Ingeniería de Software y las TIC, de técnicas que permiten extraer la información más fiable de los expertos, de los que más conocen el tema y sus implicaciones. Haciendo uso de estas técnicas, para la validación de la propuesta se empleó específicamente el método de expertos, obteniéndose criterios de especialistas en los temas de Ingeniería de Requisitos y Simuladores Virtuales de la UCI.

1.1.1 Selección de los expertos.

Una alta calificación técnica y profundidad en los conocimientos del área objeto por parte de los expertos, determina el valor de las respuestas que puedan emitir. En aras de garantizar tales objetivos se empleó el procedimiento de autoevaluación de expertos para la selección.

En la presente investigación confluyen dos áreas del conocimiento, la simulación virtual y la IR. La aplicación del método demostró que no es común encontrar expertos con altos coeficientes de competencia K en ambas áreas.

Para obtener el valor representativo (K) de la experticia del encuestado se utilizó la fórmula 3.1.

$$K = 1/2 (kc + ka) \quad (3.1)$$

Donde:

- kc representa el coeficiente de conocimiento o información de los expertos tanto en IR como en SV, calculado sobre la valoración del propio experto en una escala del 0 al 10 y multiplicado por 0.1.
- ka representa el coeficiente de argumentación o fundamentación de los criterios del experto, obtenido como resultado de la suma de los puntos alcanzados a partir de una tabla patrón (Ver Tabla 3.1).

FUENTES DE ARGUMENTACION	Grado de influencia de cada una de las fuentes en sus criterios.		
	A (alto)	M (medio)	B (bajo)
Análisis teóricos realizados por usted	0.3	0.2	0.1
Su experiencia obtenida	0.5	0.4	0.2
Trabajos de autores nacionales	0.05	0.05	0.05
Trabajos de autores extranjeros	0.05	0.05	0.05
Su propio conocimiento del estado del problema en el extranjero	0.05	0.05	0.05
Su intuición	0.05	0.05	0.05

Tabla 3.1: Tabla patrón para determinar el grado de influencia de los expertos.

Utilizando los valores de la tabla patrón para cada una de las casillas marcadas por el experto se calculó el número de puntos obtenidos en total; de manera que si el coeficiente $kc = 1.0$ el grado de influencia de todas las fuentes es alto, si $ka = 0.8$ es un grado medio y si es igual a 0.5 se considera con grado bajo de influencia de las fuentes. Tanto el valor de ka como el de kc fueron determinados a partir de la aplicación de una primera encuesta (Anexo 7) a una población de 10 especialistas.

El valor de K representa la alta, media o baja experticia del encuestado teniendo en cuenta el siguiente código de interpretación:

- Si $0.8 < K < 1.0$ coeficiente de competencia **alto**
- Si $0.4 < K < 0.8$ coeficiente de competencia **medio**
- Si $K < 0.4$ coeficiente de competencia **bajo**

El procesamiento de las encuestas aplicadas arrojó que no en muchos casos el coeficiente de competencia resultaba elevado para ambas áreas del conocimiento simultáneamente. De los 10 especialistas involucrados se escogieron aquellos que su K fue alta tanto en SV como en IR. Siguiendo este criterio resultaron seleccionados 7 expertos. De ellos 4 son máster en ciencias, 1 es licenciado y dos son ingenieros.

1.1.2 Determinación de la validez de la propuesta.

Luego de conformado el panel de expertos se les entregó un documento con los aspectos más relevantes de la propuesta acompañado de una segunda encuesta (Anexo 8). La primera parte del cuestionario tenía el propósito de evaluar el valor práctico de la aplicación de la propuesta en proyectos de desarrollo de simuladores virtuales. En este sentido se establecieron aspectos a evaluar: posibilidad de emplear las técnicas propuestas en el desarrollo de SV, conveniencia de utilizar las herramientas propuestas para desarrollar los procesos de IR durante el desarrollo de SV, así como el grado de importancia de utilizar la propuesta en los proyectos de producción de SV. La (Tabla 3.2) refleja el porcentaje representativo de los resultados obtenidos en esta primera parte.

Preguntas	Sí	No	No sé
¿Es necesaria la definición de técnicas adecuadas para llevar a cabo los procesos de ingeniería de requisitos en	100%	0%	0%

proyectos de simulación virtual?			
¿Considera posible el empleo de las técnicas para la IR propuestas en el desarrollo de simuladores virtuales?	100%	0%	0%
¿Considera conveniente el uso de las herramientas propuestas para desarrollar los procesos de ingeniería de requisitos durante el desarrollo de simuladores virtuales?	85,7%	0%	0%
	Muy importante	Medianamente importante	No importante
¿Considera importante la utilización de la propuesta en los proyectos de producción de simuladores virtuales?	100%	0%	0%

Tabla 3.2: Resultados de la primera parte del cuestionario de validación.

Una segunda parte del cuestionario tuvo como propósito validar la viabilidad de la aplicación de las técnicas y herramientas recomendadas en los procesos de IR para los proyectos de simulación. En este sentido se tuvieron en cuenta criterios de los expertos que abalan la mejora en el cumplimiento de los objetivos de cada uno de los procesos en cuestión de una manera eficiente. Los elementos establecidos para la valoración fueron: comunicación entre involucrados en el proyecto, mejor desarrollo del proceso de obtención o captura de requisitos, formalización y especificación de requisitos con suficiente calidad y detalles de manera que el personal técnico pueda proceder con el diseño y construcción de la solución y eficiencia en la gestión o administración de requisitos.

El hecho de afirmar que con el empleo de las técnicas y herramientas propuestas se alcanza una mejora en el cumplimiento de los objetivos o propósitos de los procesos de la IR, así como en la calidad de sus resultados, viabiliza la aplicación práctica de la propuesta en un proyecto de simulación virtual. Los resultados obtenidos se encuentran expresados en por ciento en la (Tabla 3.3).

Preguntas	Sí	No	No sé
¿Considera que el empleo de las técnicas y herramientas propuestas en los proyectos de producción de simuladores virtuales mejoraría la comunicación entre equipo de desarrollo y clientes y otros involucrados?	100%	0%	0%
¿Considera que el empleo de las técnicas y herramientas propuestas en los proyectos de producción de simuladores virtuales permitiría un mejor desarrollo del proceso de obtención o captura de requisitos?	100%	0%	0%
¿Considera que el empleo de las técnicas y herramientas propuestas en los proyectos de producción de simuladores virtuales ayudaría a obtener una formalización y especificación de requisitos con suficiente calidad y detalles de manera que el personal técnico pueda proceder con el diseño y construcción de la solución?	100%	0%	0%
¿Considera que el empleo de las técnicas y herramientas propuestas en los proyectos de producción de simuladores virtuales permitiría gestionar o administrar los requisitos eficientemente?	85,7%	0%	14,3%

Tabla 3.3: Resultados de la segunda parte del cuestionario de validación.

Teniendo en cuenta la aproximación o similitud entre los resultados arrojados por las encuestas a los expertos se consideró suficiente una primera iteración del método experto para validar la investigación. Es importante resaltar que el proceso arrojó algunas recomendaciones emitidas por los encuestados, relacionadas fundamentalmente con las herramientas sugeridas para el desarrollo de prototipos

que, aunque no marcaron diferencias significativas con la propuesta definida, fueron asumidas en la investigación. Ello conllevó a indicar el uso de algunos motores gráficos en específico, como lo son el Ogre 3D y Cristal Space que inicialmente no se habían concebido, sino que se definía una propuesta abierta al manejo de cualquier motor gráfico considerado por el equipo de desarrollo. Análogamente ocurrió con la herramienta CEGUI que en los inicios no se había tenido en cuenta, tomando en consideración solo el empleo de Qt Creator. Aún conociendo la limitante de ser privativa la herramienta RequisitePro para la gestión de requisitos, los criterios emitidos sobre su uso fueron positivos en la mayoría de los expertos tomando en consideración la insuficiente cantidad de herramientas libres en esta área.

3.2 Aplicación práctica de la propuesta en el proyecto simulador quirúrgico.

Luego de una validación de la propuesta por parte de un grupo de expertos en el área, asegurando la posibilidad de aplicar las técnicas y herramientas recomendadas en un proyecto de simulación virtual, se presentará en este apartado los resultados de su aplicación en el proyecto de simulación virtual "Simulador Quirúrgico" que se desarrolla en el polo de realidad virtual de la UCI. Primeramente se abordará una breve explicación sobre el proyecto en cuestión que permita comprender la aplicación de las técnicas propuestas y los resultados obtenidos.

3.2.1 Estructura del proyecto Simulador Quirúrgico

El proyecto Simulador Quirúrgico tiene como propósito general obtener un prototipo cubano de simulador que aplique la Realidad Virtual al proceso de enseñanza de la cirugía de mínimo acceso, específicamente la cirugía laparoscópica y la endoscopia diagnóstica, basado en la adquisición de imágenes provenientes de pacientes. De esta forma el cirujano podrá no solo entrenarse a partir de una extensa base de ejercicios, sino que además podrá practicar y planificar una intervención quirúrgica antes de realizarla. El proceso de desarrollo del simulador se divide básicamente en cuatro etapas o fases de desarrollo que generan a su vez cuatro productos entregables:

1. Simulador de habilidades básicas, a partir de modelos de imitación de procesos con objetos rígidos.
2. Simulador de habilidades avanzadas, a partir de la imitación de procesos.
3. Primera versión del simulador a partir de la generación de órganos virtuales.

4. Simulador profesional con escenarios contruidos a partir de estudios de Tomografía Axial Computarizada (TAC) y Resonancia Magnética (RM).

Cada uno de estos productos constituye una versión funcional a partir de la cual se podrán entrenar las diferentes habilidades a desarrollar en los médicos cirujanos, desde las más básicas hasta el entrenamiento en un simulador profesional en la cuarta y última etapa. Las habilidades podrán ejercitarse a partir de los diferentes ejercicios que comprende cada entregable. Se tiene además la documentación y ayuda necesaria para el entendimiento y posible uso de cada versión funcional (Ver Figura 3.1).

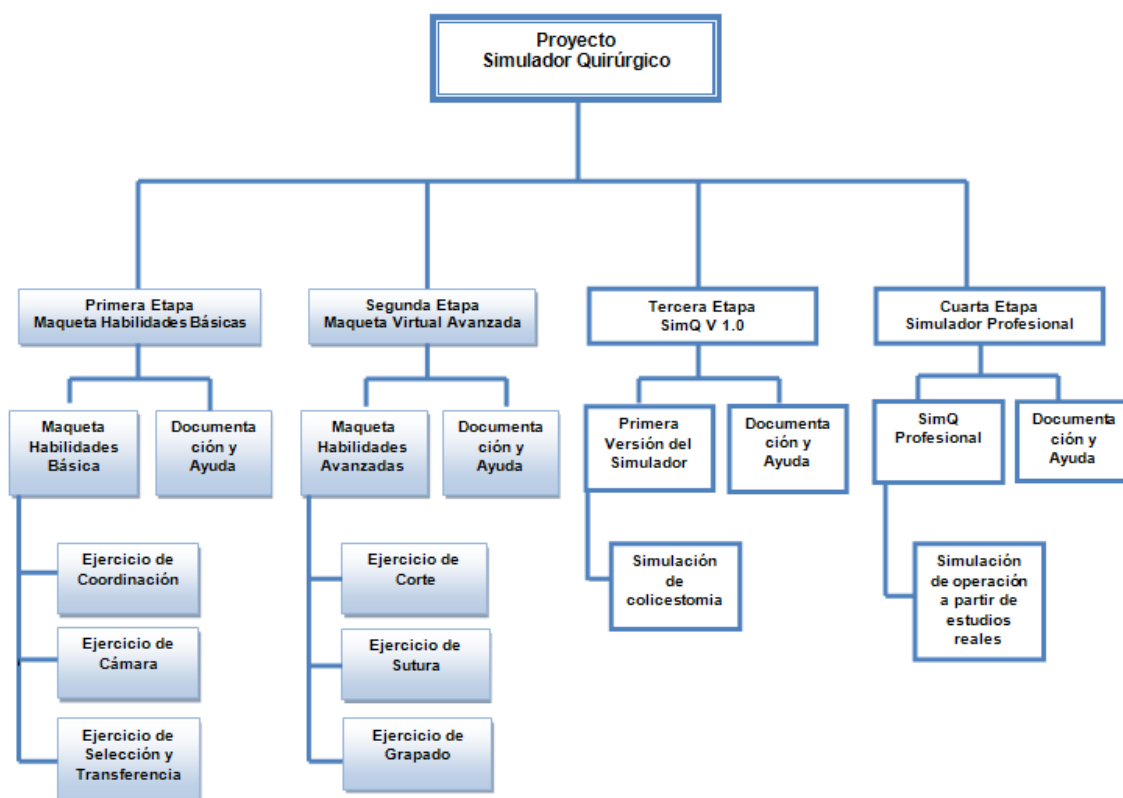


Figura 3.1: Estructura de desglose del proyecto Simulador Quirúrgico.

La propuesta de solución técnica que se presentará, basada en la aplicación de las técnicas y herramientas propuestas anteriormente para este proyecto, así como la valoración del equipo de trabajo del proyecto acerca de los resultados obtenidos, se enmarcan en la segunda etapa o fase de desarrollo del simulador, Maqueta Virtual Avanzada, debido a la correspondencia del período actual con el cronograma de ejecución.

3.2.2 Situación inicial del simulador quirúrgico

Este proyecto simulador quirúrgico, es uno de los proyectos del PRV de la UCI tomado como objeto de estudio en esta investigación al aplicarle la lista de chequeo referenciada a inicios de este documento (Anexo 1). Este paso reveló las deficiencias relacionadas con el tratamiento de los requisitos que existían inicialmente, antes del empleo de las técnicas y herramientas propuestas en este trabajo.

Primeramente no se tenía una especificación de requisitos, como principal producto de trabajo de la IR, que permitiera una mayor claridad en el desarrollo de las restantes fases del ciclo de vida del proyecto.

Sobre los requisitos existentes se pudo comprobar lo siguiente:

- No todos los requerimientos eran correctos, pues no todos representaban elementos requeridos para la construcción del sistema y se mostraron además, errores que afectaban el diseño del sistema.
- Existían requisitos que no eran expresados con claridad. No todos se expresaban en un lenguaje comprensible para el cliente y en algunos casos solían estar sujetos a más de una interpretación.
- No se mostró una completitud. No todos los requisitos estaban definidos, tampoco se identificaban las áreas carentes de información. En ningún caso existían especificaciones acerca del rendimiento, restricciones de diseño o elementos de hardware.
- No eran concisos, pues no todos se mostraban en un único lugar y existía además, información no significativa para la especificación.
- No todos los requisitos eran verificables. En algunos casos su implementación era imposible de realizar.
- No se tenía en cuenta la trazabilidad de los requisitos. No estaba concebido el rastreo de los requisitos hacia su incorporación en el resto de las actividades del ciclo de vida y tampoco se consideraban las posibilidades de cambio.
- No se tenía en cuenta una priorización de los requisitos según su importancia y/ estabilidad. En ningún requisito se especificaba su nivel de prioridad para afrontar el desarrollo.
- En ningún caso se incluye anexos como figuras, tablas y/o diagramas que faciliten la comprensión de lo planteado.

En la (Figura 3.2) se puede apreciar la representación gráfica de la disponibilidad de cada uno de los anteriores atributos de calidad de los requisitos en el proyecto simulador quirúrgico.

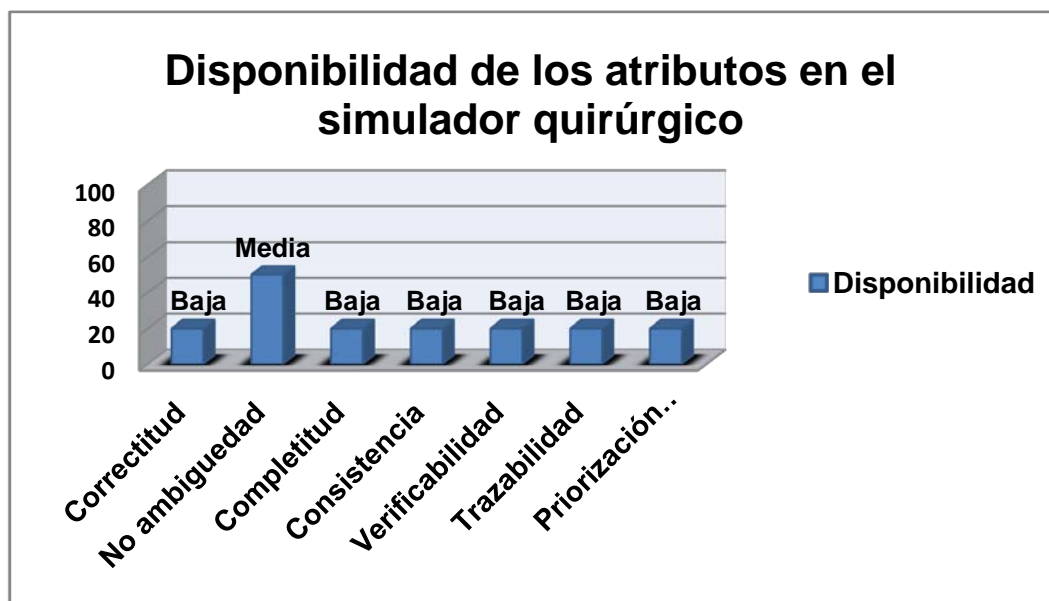


Figura 3.2: Disponibilidad de los atributos de calidad de los requisitos en el proyecto Simulador Quirúrgico.

3.2.3 Técnicas utilizadas en el simulador quirúrgico.

El trabajo realizado en el proyecto Simulador Quirúrgico funcionó como experimento práctico de la propuesta formulada en esta investigación. Inicialmente se concibió un proceso de obtención de requisitos haciendo uso de las técnicas propuestas, las cuales fueron ejecutadas por un equipo multidisciplinario de médicos cirujanos, analista y el líder del proyecto. El uso de la técnica del aprendiz en este proceso constituyó, sin lugar a dudas, el aporte más significativo al entendimiento entre los involucrados y la clara comprensión de los requisitos.

Luego de un proceso de análisis para lograr una formalización en detalle de los requisitos se construyó la especificación de los mismos haciendo uso de las técnicas de casos de uso y escenarios combinado con storyboards.

Casos de uso

Se definieron casos de uso para modelar y describir los requisitos que representaban interacción con actores: Autenticar usuario, Gestionar usuarios, Iniciar ejercicio, Resetear ejercicio, Emitir evaluación y Gestionar evaluación. La (Figura 3.2),

construida con la herramienta Enterprise Architect, recomendada en la propuesta para llevar a cabo el desarrollo de requisitos, ilustra el modelo de casos de uso del sistema que representa la relación de dichas funcionalidades con los correspondientes actores del sistema. Cada caso de uso representado contiene una descripción textual que especifica la secuencia de acciones a ejecutarse tanto por parte del usuario como por el sistema.

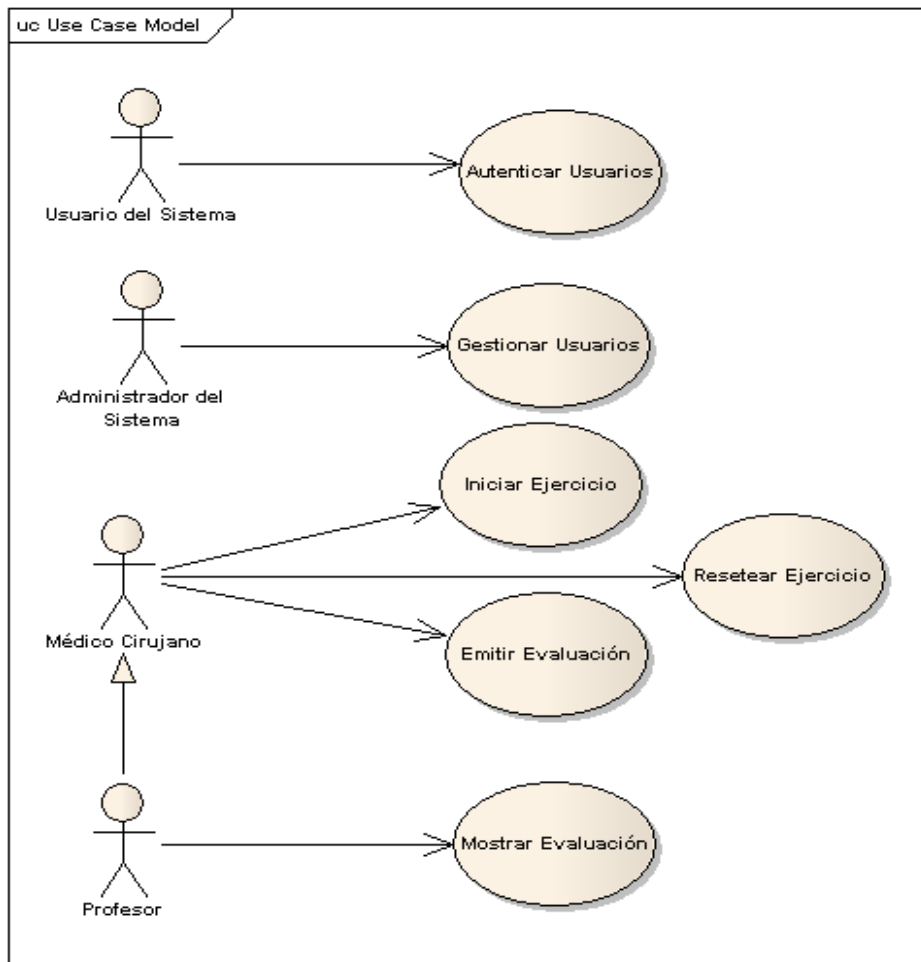


Figura 3.2: Modelo de casos de uso del sistema Simulador Quirúrgico.

Escenarios y Storyboards

Se definieron Escenarios con storyboards incluidos para representar los requisitos que llevan implícito una mayor interactividad y que exigen una representación gráfica para lograr su entendimiento: Interactuar con la cámara, Focalizar objetos, Seleccionar objetos, Interactuar con pinzas, Agarrar y transferir objetos, Entrenar corte, Entrenar sutura, Entrenar grapado. La construcción de los storyboards estuvo apoyada en la

captura de imágenes, las cuales fueron obtenidas durante el proceso de obtención de requisitos, específicamente mediante el empleo de la técnica del aprendiz.

Construcción de prototipos

Como parte de la validación de la especificación de requisitos obtenida se construyeron los prototipos del sistema haciendo uso de la técnica de prototipado y finalmente se obtuvo un prototipo general que incluye todas las pantallas diseñadas. Para su construcción se acudió a la recomendación dada en el epígrafe 2.9.1 sobre el uso de un Motor Gráfico como herramienta de construcción, dado que se contaba con el motor Scen Toolkit, desarrollado en el propio polo de realidad virtual de la UCI, el cual se ajustaba a las necesidades del proyecto y su manipulación resultaba familiar para el equipo de trabajo.

Tanto los storyboards incluidos en los escenarios, como las descripciones textuales de los casos de uso propuestos en la especificación de requisitos, funcionaron como base para la construcción de los prototipos. La composición del prototipo general consiste en varias interfaces interconectadas entre sí, de manera que cuando se pulse algún botón de la interfaz la aplicación responderá a dicha acción mostrando la interfaz correspondiente.

Al iniciarse la aplicación se carga en pantalla una interfaz que le permitirá al usuario autenticarse y de esta forma se le concedan los permisos correspondientes, activándose el perfil consecuente con su rol (Ver Figura 3.3).



Figura 3.3: Interfaz autenticar usuario.

Si el usuario que se autentica tiene asignado el rol administrador, la interfaz que se presentará, brindará la posibilidad de escoger mediante el menú si desea registrar, modificar o eliminar los datos de un usuario (Ver Figura 3.4).

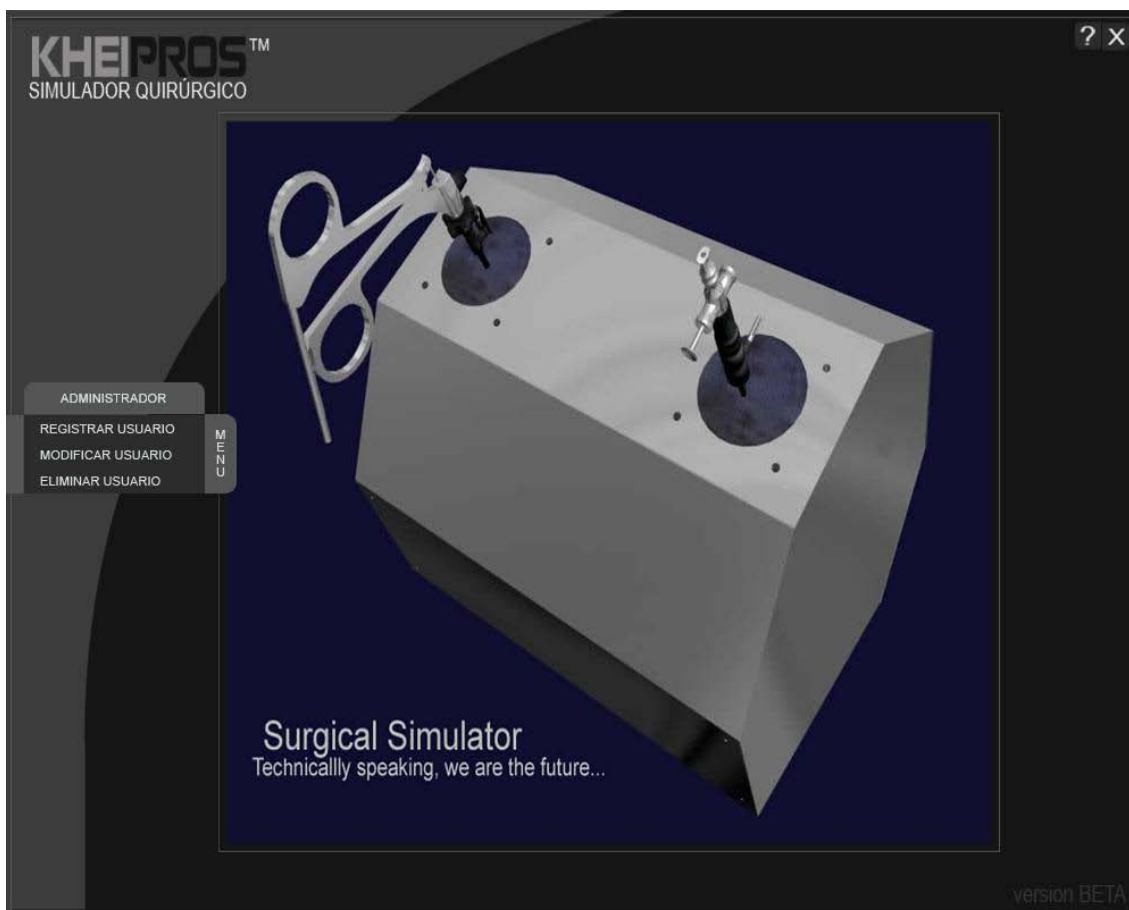


Figura 3.4: Interfaz principal de la sesión administrador.

Al seleccionarse algunas de las opciones del menú antes mencionadas se activa la ventana correspondiente a dicha opción, mostrándose los datos necesarios para ejecutar la acción.

Si el usuario autenticado tiene asignado el rol médico cirujano, se presentará una interfaz en la cual el médico al desplegar el menú podrá seleccionar el ejercicio en el que se desea entrenar (Ver Figura 3.5).

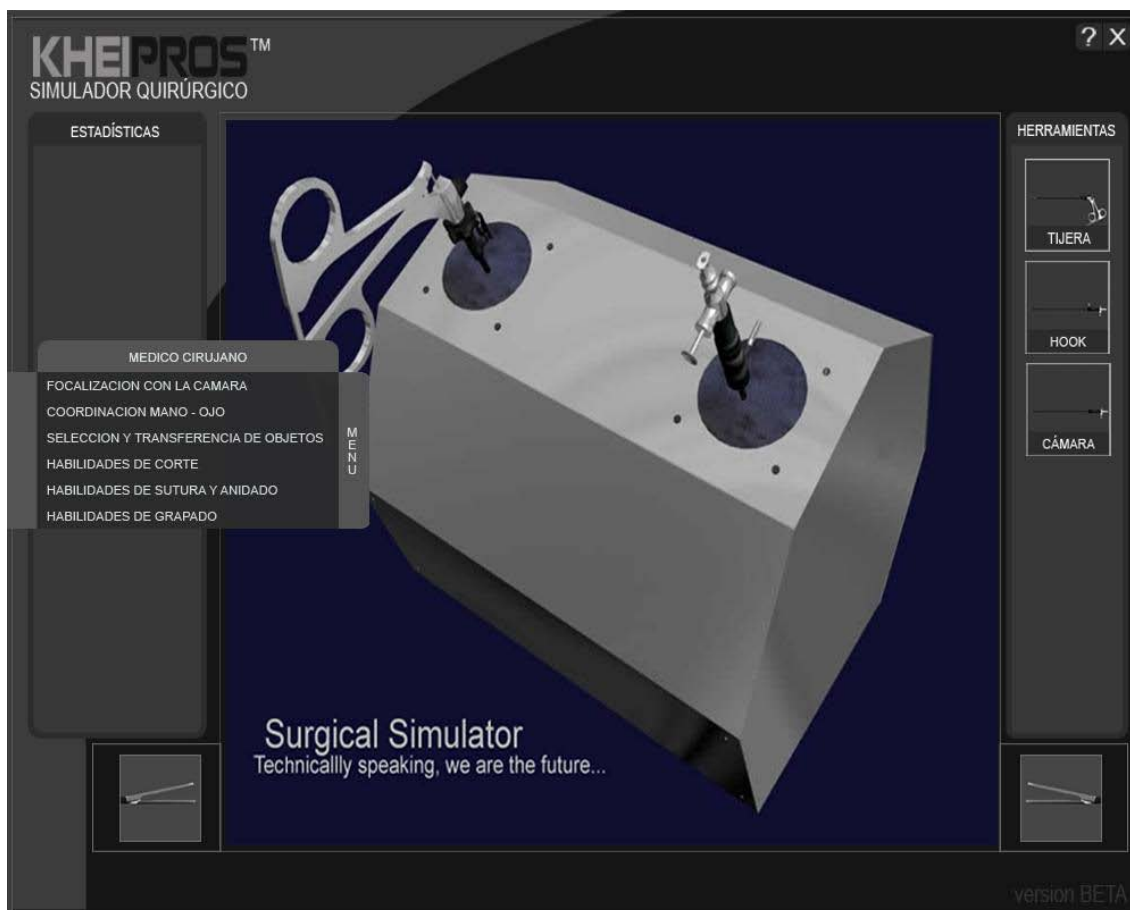


Figura 3.5: Interfaz Principal del médico cirujano.

Al seleccionar el ejercicio deseado se activa la pantalla correspondiente al mismo con sus respectivas acciones e información.

Si el usuario que se autentica tiene asignado el rol profesor se mostrará una interfaz con la opción de entrenarse en algunos de los ejercicios, al igual que un médico cirujano y además podrá revisar las evaluaciones obtenidas por los médicos que ya se han entrenado en dichos ejercicios (Ver Figura 3.6)



Figure 3.6: Interfaz principal del profesor.

De igual manera que en los roles anteriores, al ejecutar alguna de las acciones del menú se activará su ventana representativa. Si el profesor decide revisar la evaluación de algún médico se carga en pantalla una interfaz que permitirá seleccionar el nombre del médico que se desea consultar sus evaluaciones, así como el ejercicio que se revisará (Ver Figura 3.7).

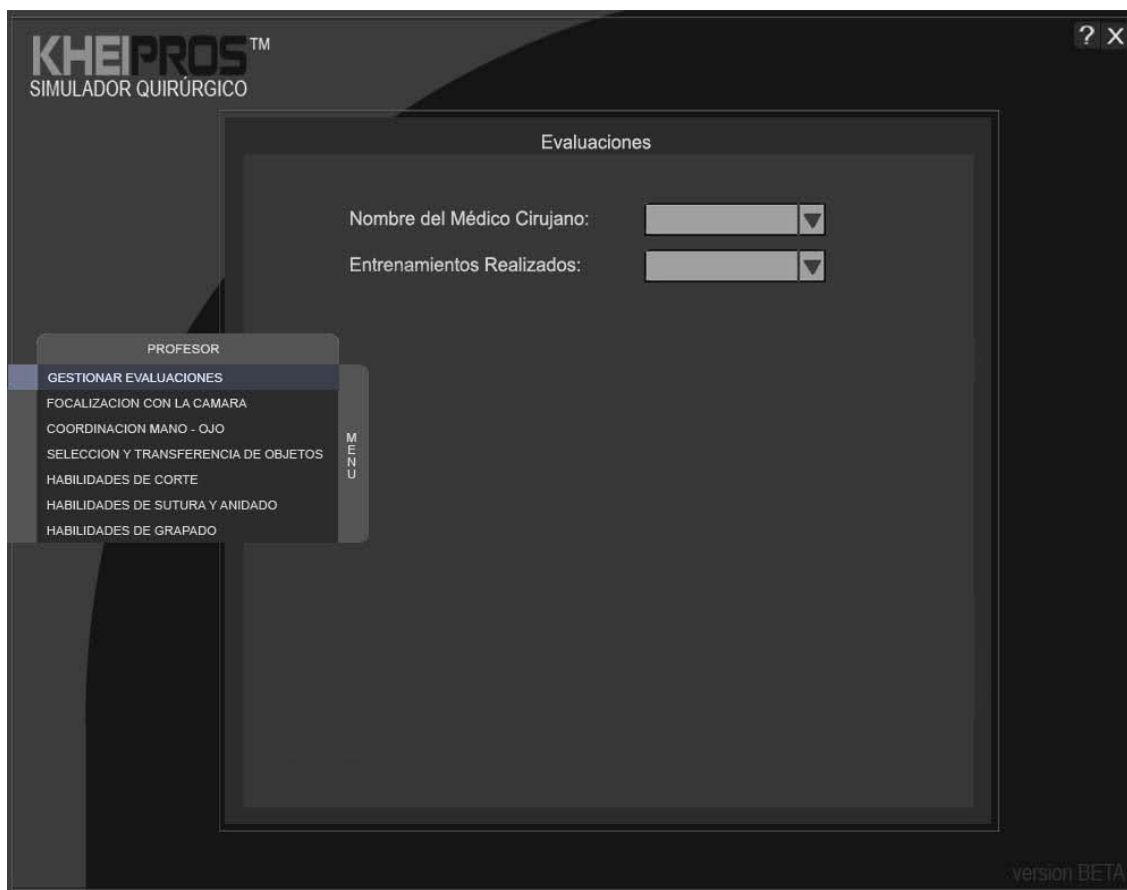


Figura 3.7: Interfaz gestionar evaluaciones.

3.2.4 Resultados arrojados en el proyecto simulador quirúrgico.

Luego de empleada las técnicas y herramientas propuestas en el proyecto simulador quirúrgico se recopilieron algunos criterios por parte del equipo de desarrollo que reflejan el positivo efecto de su aplicación. Estos resultados suelen ser más impactantes si se confrontan con los descritos en el epígrafe 3.2.2, donde se comprobó una baja disponibilidad de la mayoría de los atributos de calidad de los requisitos en dicho proyecto. Seguidamente se presenta un resumen de los resultados obtenidos después de aplicada la propuesta donde se enfatiza la disponibilidad de dichos atributos de calidad en la especificación de requisitos realizada.

Para recopilar los datos obtenidos se realizó una encuesta a 10 integrantes del equipo de desarrollo del proyecto simulador quirúrgico (Anexo 9) donde se pudo constatar que el 100% de los encuestados afirman que la especificación de requisitos obtenida es en alta medida: correcta, clara/no ambigua, verificable y posible de llevar a cabo la trazabilidad. Para el caso de los atributos de completitud, consistencia y priorización

de los requisitos según su importancia y/o estabilidad el 90% de los encuestados ratifican su disponibilidad en alta medida en la especificación obtenida y solo el 10% lo afirman en valor medio (Ver Figura 3.8).

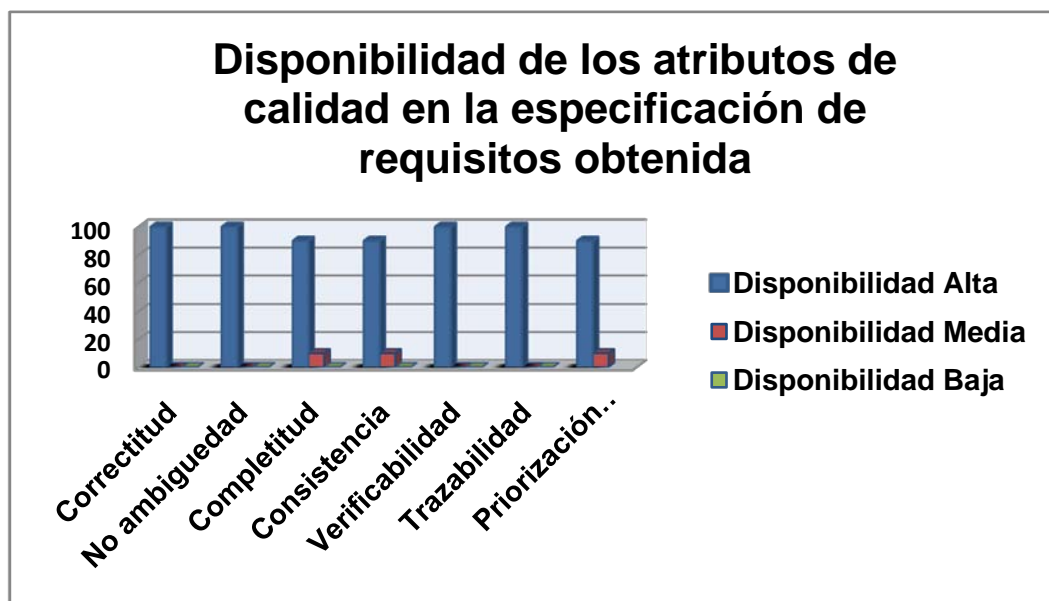


Figura 3.8: Disponibilidad de los atributos de calidad en la especificación de requisitos obtenida en el simulador quirúrgico luego de aplicada la propuesta.

En la misma encuesta se comprobó también el positivo efecto, propiciado con la aplicación de la propuesta, en algunos factores o actividades realizadas inicialmente en el proyecto cuando aun no se aplicaban las técnicas y herramientas propuestas y que atentaban contra la calidad del producto y el tiempo de desarrollo del mismo. Los elementos analizados son: necesidad del equipo de desarrollo de realizar búsquedas en internet o consultar otras bibliografías, solicitudes de cambios en los requerimientos y comunicación entre el equipo de desarrollo y los analistas.

De los 10 desarrolladores encuestados solo el 20% tuvo necesidad de realizar búsquedas en internet o consultar otras bibliografías para una total comprensión de las funcionalidades a desarrollar (Ver Figura 3.9). Lo que corrobora la disponibilidad de los atributos de completitud y claridad / no ambigüedad en la especificación de requisitos realizada.

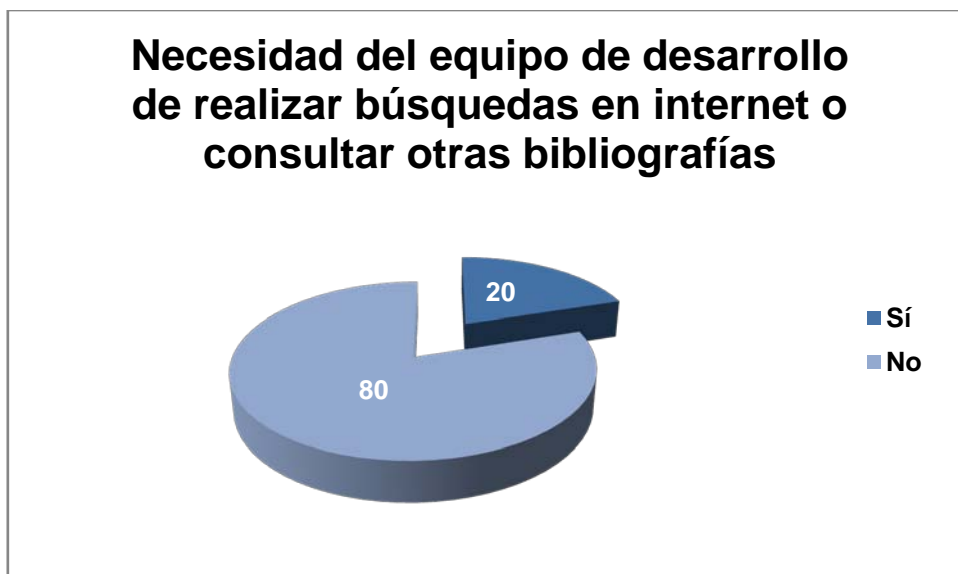


Figura 3.9: Necesidad del equipo de desarrollo del simulador quirúrgico de realizar búsquedas en internet o consultar otras bibliografías luego del empleo de la propuesta.

Los parámetros solicitudes de cambios en los requerimientos y comunicación entre el equipo de desarrollo y los analistas pueden ser interpretados en la (Figura 3.10) de la siguiente manera:

- A: Disminución de las solicitudes de cambios en los requerimientos.
- B: Mejora en la comunicación entre el equipo de desarrollo y los analistas.

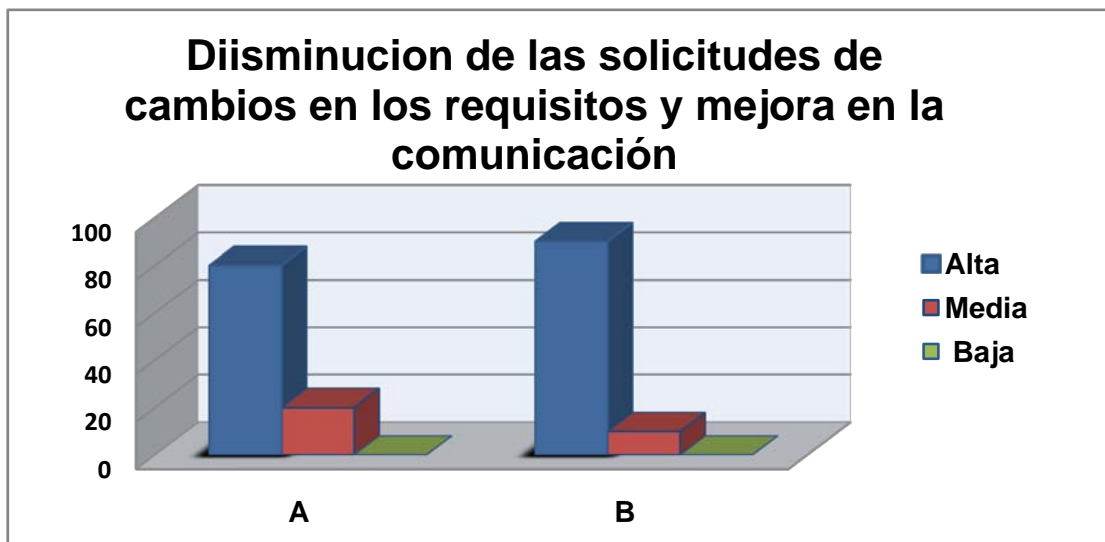


Figura 3.11: Disminución de las solicitudes de cambios en los requisitos y mejora en la comunicación entre analistas y el resto del equipo de desarrollo.

Como se muestra en la gráfica anterior, el experimento arrojó también que el 80% de los encuestados reflejan, en alta medida, que la aplicación de la propuesta tuvo una influencia positiva en la disminución en las solicitudes de cambios en los requerimientos y solo un 20% lo considera en valor medio. De igual forma se obtuvo que el 90% valora, en alta medida, el efecto de mejora en la comunicación entre el equipo de desarrollo y los analistas y solo un 10% lo concibe en valor medio.

Con el análisis de los resultados presentados en este apartado se puede concluir que los objetivos definidos para la propuesta fueron asequibles mediante la aplicación de las técnicas y herramientas propuestas. Dichos objetivos iniciales fueron establecidos teniendo en cuenta los indicadores definidos para la variable dependiente **mayor calidad del producto de trabajo de la ingeniería de requisitos para proyectos de simulación virtual**. El hecho de afirmar que con la aplicación de la propuesta se alcanzan los objetivos señalados, permite validar el cumplimiento de la hipótesis definida.

Consideraciones del capítulo

- La aplicación del método de expertos permitió validar la necesidad e importancia del empleo de las técnicas y herramientas propuestas en proyectos de simulación virtual, así como su contribución a la mejora en el desarrollo de los procesos de la IR para los simuladores virtuales, lo que viabiliza la aplicación práctica de la propuesta en dichos proyectos.
- Con el empleo de la propuesta en el proyecto simulador quirúrgico se obtuvo como resultado una especificación de requisitos validada mediante un prototipo no funcional que posibilitó una mejor comprensión y claridad a los desarrolladores del proyecto.
- La aplicación práctica de la propuesta en el proyecto simulador quirúrgico arrojó resultados favorables a la disponibilidad de los atributos de calidad en la especificación de requisitos obtenida en el proyecto.

Conclusiones

- Se obtuvo una propuesta de técnicas y herramientas para la IR en proyectos de simulación virtual que propician una mayor calidad del producto de trabajo de la IR para este tipo de sistemas.
- La propuesta fue validada por expertos en ambas áreas del conocimiento, simuladores virtuales e ingeniería de requisitos, quienes por mayoría coincidieron en que su aplicación en proyectos de simulación virtual resulta posible e importante y que con su implantación y ejecución se logrará alcanzar los objetivos definidos en cada uno de los procesos que comprenden la IR.
- Se realizó una aplicación práctica de la propuesta en uno de los simuladores del polo de realidad virtual de la UCI (simulador quirúrgico), la cual arrojó una total disponibilidad de los atributos que miden la calidad de la especificación de requisitos realizada. Lo que significa una mejora en la calidad del producto de trabajo de la IR obtenido.
- Las técnicas y herramientas propuestas para la IR en simuladores virtuales favorece el trabajo de los ingenieros que enfrentan la ausencia de procedimientos formalmente descritos y adaptados a sus necesidades en esta área de la IS.

Recomendaciones

- Aplicar las técnicas y herramientas propuestas en el desarrollo y gestión de requisitos de proyectos de producción de simuladores virtuales.
- Mantener la vigilancia tecnológica sobre las herramientas libres de gestión de requisitos para detectar mejoras en las existentes o nuevas, que soporten con eficiencia la automatización de la técnica propuesta.
- Promover la preparación de especialistas en pre y postgrado dedicados al desarrollo de simuladores virtuales en temas de ingeniería de software y particularmente en ingeniería de requisitos.

Referencias Bibliográficas

1. **Jhon, F. Von, D.** "Virtual Reality, VPL Research". s.l.: Disponible en <http://www.icc.uji.es/assignatura>, 2006.
2. **Nerys, Yisel.** *Interfaz para el manejo de los dispositivos de entrada y salida en Sistemas de Realidad Virtual.* Trabajo de Diploma. Universidad de Ciencias Informáticas. : s.n., 2007.
3. **Institute for. University of Central Florida.** *Just what is "simulation" anyway?* simulation. Disponible en: <http://www.ist.ucf.edu/background.htm> : s.n., 2005.
4. **Chover, M.** *Introducción a la Informática Gráfica. Capítulo 9. Aplicaciones de la Informática Gráfica.* Buscarlo.
5. **Jacobson, Ivar. Booch, Grady. Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* s.l. : S.I Pearson Education, S. A, , 2000.
6. **Bahamonde, Jose Manuel. Rossel, Richard.** *Un acercamiento a la Ingeniería de Requisitos.* Universidad técnica Federico Santa María. : s.n., 2003.
7. **Presman, Roger S.** *Ingeniería de Software. Un enfoque práctico. Sexta Edición.* . s.l. : Mexico: Mcgraw-Hill., 2006.
8. **RUP.** *Rational Unified Process.* . 2003.
9. **Presman, Roger.** *Ingeniería de Software. Un enfoque práctico. Quinta Edición.* . España: Mcgraw-Hill. : s.n., 2002.
10. **IEEE.** *IEEE-STD-830-1998 : Especificaciones de los requisitos del Software.* 1998.
11. *Prácticas recomendadas para la especificación de requisitos de software. software., Dirección de calidad de.* Infraestructura productiva. Universidad de Ciencias Informáticas. : s.n.
12. **Escalona, M. Koch, N.** *Ingeniería de requisitos en aplicaciones para la web. Un estudio comparativo.* . Universidad de Sevilla. : s.n., 2002.
13. **García, E. Martínez, M. Desiree, A. Saavedra, M.** *Sobre Realidad, Surrealismo y realidad virtual. IES Príncipe de Asturias de Lorca (Murcia)* . Disponible en: www.lacavernadeplaton.com/articulosbis/sobrerealidad0506.htm : s.n., 2005.
14. **Pacheco, Y. Díaz, A.** *Propuesta de Algoritmo para la Simulación de las explosiones aéreas para los simuladores de tiro.* . Trabajo de Diploma. Universidad de Ciencias Informáticas. : s.n., 2007.

15. **Peirano, M.** *Metaversos aéreos: mundos virtuales para los simuladores de vuelo.* . Disponible en: <http://www.consumer.es/web/es/tecnologia/software/2007/04/27/161875.php> : s.n., 2007.
16. **Consulting, Michelsen.** *Empresa de consultoría productora de simuladores de negocio.* Disponible en: <http://gerentevirtual.com/quienes.asp> : s.n., 2006.
17. **Del Olmo, A. Martínez, M. Carretero A.** *Desarrollo de un simulador para realizar prácticas virtuales de topografía utilizando gráficos Web3D. Universidad politécnica de Madrid.* . Disponible en: <http://www.cartesia.org/geodoc/ingegraf2005/T15.pdf> : s.n., 2005.
18. **Stern, J. Zeltser IS. Pearle MS.** *Simbionix. Compañía productora de simuladores médicos.* . Disponible en: <http://www.simbionix.com> : s.n., 2007.
19. **Softonic., Compañía.** *Madrid, España.* Disponible en: <http://www.softonic.com/s/simulador-conduccion> : s.n., 2005.
20. **Ibarreche, Daily. Macías, Manuel.** *Modulo de seguimiento de terrenos para sistemas de realidad virtual.* . Trabajo de Diploma.Universidad de Ciencias informáticas. : s.n., 2007.
21. **Pérez, L.** *Fragmentos del Periódico Granma.* . Disponible en: <http://granma.co.cu/2007/06/08/nacional/artic01.html> : s.n., 2007. .
22. **Dorfman, M. Richard, H.** *Standards, guidelines and Examples of System and Software Requirements Engineering.* . Los Alamitos: IEEE Computer Society Press, : s.n., 1990.
23. **Larman, Craig.** *UML y patrones. Introducción al análisis y diseño orientado a objetos.* . s.l. : Prentice Hall Hispanoamericana, S.A., 1999.
24. **Society, IEEE Computer.** *SWEBOOK Guide to the Software Engineering Body of Knowledge.* 2004.
25. **Wieggers, Karl.** *More About Software Requirements: Thomy Issues and Practical Advice.* 2006.
26. **Leffingwell, Dean. Widrin, Don.** *Managing Software Requirements: A Use case Approach.* s.l. : Second Edition. SI: Addison Wesley. , 2003.
27. *Development Chaos Today. Report., Standish Group.* 2007.

28. **Duran Toro, Amador.** *Metodología para la elicitación de requisitos de software.* Sevilla, Universidad de Sevilla. . 2002.
29. **Leffingwell, Dean. Widrin, Don.** *Managin Software requirements. A Use Case Approach, Second Edition.* 2003.
30. **Wiegers, Karl.** *Software Requirements, Second Edition.* . 2003.
31. **Rangel Carrillo, Adriana.** *Análisis comparativo de técnicas de obtención de requerimientos para el módulo de facturación del aplicativo gestasoft hospitalario para lmsalud.* Trabajo de Diploma Universidad de Pampolona. Colombia. : s.n., 2007.
32. **SEI.** *Software Engineering Institute.* [Citado el: 6 de 3 de 2008.]. s.l. : <http://www.sei.cmu.edu/str/indexes/glossary/requirements-engineering.html>., 2008.
33. **CMMI, Product Team.** *Software Engineering Institute. Carnegie Mellon. CMMI® for Development, Version 1.2.* [En línea] Agosto de 2006.
34. **9001:2000., ISO.** *Organización Internacional para la Estandarización. ISO/FDIS 9001:2000 (E). Sistemas de Gestión de la Calidad-Requisitos.* 2000.
35. **Arias Chaves, Michael.** *La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de Software.* . Julio del 2006. .
36. **Process., Rational Unified.** *Ayuda de Rational Unified Process.* 2003.
37. **Robertson, Suzanne. Robertson, James.** *Mastering the Requirements Process.* s.l. : Second Edition. SI: Addison Wesley Professional., 2006.
38. *Requirements Prioritization Introduction.* **R. Mead, Nancy.** Software Engineering Institute. Carnegie Mellon University. : s.n., 2008.
39. *Comunicación: El aspecto humano de la Ingeniería de Requisitos.* **Rojas Poblete, Cristian.** Departamento de Ciencias de la Computación. Universidad de Chile. : s.n., 2007.
40. **Pérez, Karina.** *Modelo de referencia de IR para proyectos de Bioinformática.* . Tesis para optar por el grado científico de Master en Ciencias. : s.n., 2007.
41. **Hadad, Graciela. Doorn, Jorge H y Kaplan, Gladys N.** *Enfoque Middle-Out en la construcción e Integración de Escaneros.* s.l. : [En línea] http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER99/hadad.pdf, 1999.
42. **Ferré Grau, Xavier.** *Marco de Integración de la usabilidad en el proceso de desarrollo de software.* Tesis Doctoral. Universidad Politécnica de Madrid : s.n., 2005.
43. **Lauesen, Soren.** *Software Requirement. Styles and Techniques.* 2002.

44. **Spector, David HM.** *CASE Tools: Large System Development*. . s.l. : [Citado el: 17 de 09 de 2008.]. Disponible en: <http://www.linuxdevcenter.com/pub/a/linux/2002/08/01/enterprise.html>, [En línea] 08 de 01 de 2002.
45. **Lam, Dora y Achrafi Rabi.** *Volere. Requirements Tools*. . [En línea] Febrero de 2009. [Citado el: 9 Abril 2009] Disponible en: <http://www.volere.co.uktools.htm> : s.n., Febrero de 2009.
46. **ArgoUML.** *ArgoUML Project home. Tigris.org Open Source Software Engineering Tools*. [En línea] : <http://argouml.tigris.org/>., 2006.
47. *Visual Paradigm for UML Model-Code-Deploy platform. Visual Paradigm*. . s.l. : <http://www.visual-paradigm.com/product/vpuml/>., 2007.
48. **Mcdonald Landazuri, Bárbara A.** *Definición de Perfiles en Herramientas de Gestión de Requisitos*. Madrid, España. : s.n., 2005.
49. **INCOSE.** *INCOSE Requirements Management Tools Survey*. [En línea]: http://www.paper-review.com/tools/rms/response.php?vendor=OSRMTv1_3#1 : Consultado [29-Septiembre-2008], 2006.
50. **Fraga Filho, Clayton V y Dos Reis, José M.** *CONTROLA: Herramienta de apoyo al proceso de desarrollo de software en las pequeñas compañías*. Universidad Federal de Viçosa-MG - Brasil : Revista Ingeniería Informática, Edición 12,, Abril 2006.
51. **Torres, Andrea . Guerrero, Restrepo y Hernández, Tatiana.** *Realidad Virtual*. [En línea en:ftp.eia.edu.co/Sitios%20Web/bioinstrumentacion/docs/signals/2006/exposiciones/RealidadVirtual_documento.pdf] : s.n., 2006.
52. **Jounghyun Kim, Gerard.** *Designing Virtual Reality Systems the Structured Approach*. 2005.
53. **Jounghyun Kim, Gerard.** *Designing Virtual Reality Systems*. Department of Computer Science and Engineering. Pohang University of Science and Technology (POSTECH). Korea. : s.n., 2005.
54. *Requirements Prioritization.* **Cheng, Betty.** Computer Science and Engineering. Michigan State University. : s.n., 2007.
55. **Singh, Jagjit.** *Use of AHP in Options Analysis*. 2004.
56. **exactas., Facultad de ciencias.** *Ingeniería de requisitos. Escenarios*. Universidad Nacional del centro de Buenos Aires. Argentina. : [En línea]

- www.exa.unicen.edu.ar/catedras/ingrequi/index_archivos/Notas-Escenarios.pdf..., 2007.
57. **Flynt Jhon P, Salem, Omar.** *Software Engineering for Game Developers.* 2005.
58. **Zielczynski, Peter.** *Requirements Management Using IBM Rational RequisitePro.* Diciembre, 2007.
59. **Seo, Jinseok y Oh, Sei-woong.** *PVoT: An Interactive Authoring Tool for Virtual Reality.* Dept. of Game Engineering, Dong-eui Univ., Busan, Korea : International Journal of Computer Science and Network Security, VOL.7 No.4,, 2007.
60. *Teaching Structured Development of Virtual Reality Systems using P-VoT.* **J. Kim Gerard y Seo, Jinseok.** s.l. : Dept. of Computer Science and Engineering, Korea Univ, Seoul, Korea, 2007.
61. **INCOSE.** *Requirements Management Tools Survey.* . [En línea] [Citado: 30 Abril de 2009] : <http://www.paper-review.com/toolsrmsread.php>, 2008.
62. **Román Durán, Maypher.** *PSM: Una propuesta para la medición de Software en la UCI.* . s.l. : Tesis de maestría. Universidad de Ciencias informáticas., 2007.
63. **Santos Urquiaga, Daylin y La Rosa Castro, Liena.** *Propuesta de proceso de desarrollo de software del simulador quirurgico.* Trabajo de Diploma. Universidad de Ciencias Informáticas. : s.n., 2008.

Anexos

Anexo 1: Lista de chequeo de especificaciones de requisitos.

Lista de comprobación para evaluar la calidad de las especificaciones de requisitos.

Empresa:

Producto:

Nivel	Evaluación	E	NP	Comentario
Correctitud				
	Cada requisito contenido representa algo requerido para la construcción del sistema.			
	Existen errores que puedan afectar el diseño del sistema.			
	Es favorable el criterio del usuario/cliente sobre la especificación.			
Claridad/No ambigüedad				
	Los requerimientos están expresados en un lenguaje comprensible para el cliente/usuario.			
	Cada requerimiento admite tan solo una única interpretación.			
	Los requerimientos podrían ser comprendidos, implementados y verificados por un grupo independiente.			
Complejidad (Interna)				
	Todos los requerimientos están definidos.			
	Todos los términos están definidos.			
	Todas las unidades de medida están definidas.			
	Se han identificados las áreas para las que falta información o es incompleta.			
Complejidad (Externa)				
	Están incluidos todos los requerimientos relacionados con:			
	➤ Funcionalidad			

	➤ Rendimiento			
	➤ Restricciones de Diseño			
	➤ Hardware			
	➤ Interfaces Externas			
	Están especificados los posibles cambios a los requerimientos			
	La probabilidad de cambios está especificada para cada requerimiento			
Concisión				
	Cada requerimiento se especifica en un único lugar.			
	Cada frase aporta a la especificación.			
	Siempre que es posible se recurre a un diagrama o tabla.			
Consistencia (Interna)				
	No hay dos requerimientos en conflicto debido a que describen:			
	➤ el mismo objeto con distinta terminología			
	➤ el mismo objeto como si tuviera distintas características			
	➤ dos o más acciones incompatibles lógicamente			
Consistencia (Externa)				
	Los requerimientos son consistentes con el contexto de negocio y las justificaciones del proyecto.			
Clasificados por importancia y/o estabilidad				
	Cada requerimiento tiene asignado un nivel de prioridad para abordar el desarrollo.			
Verificabilidad				
	Cada requerimiento es implementable.			
	Para cada requerimiento existe un procedimiento que, ejecutado por una persona o máquina, permite verificar que el sistema construido satisface lo descrito en el propio requisito.			
Modificabilidad				

	El documento de requerimientos está organizado de forma clara y lógica.			
	La estructura permite realizar cambios sobre los requisitos que contiene de forma sencilla, completa y consistente, manteniendo la estructura inicial del conjunto.			
Trazabilidad				
	Está claro el origen de cada requisito individual.			
	De cada requisito se puede rastrear hacia delante su incorporación en el resto de las actividades del ciclo de vida.			
Contenido general				
	Ninguno de los requerimientos realmente corresponde a:			
	➤ detalles de diseño			
	➤ detalles de gestión del proyecto			
	Están presentes:			
	➤ Todos los anexos necesarios			
	➤ Todas las figuras, tablas y diagramas necesarias			

Leyenda:

Nivel: Importancia del aspecto a evaluar

E: Evaluación

NP: No Procede

Comentario: Es obligatorio en las respuestas negativas

Las evaluaciones serán:

Bajo: Propiedad no disponible (B)

Medio: Propiedad parcialmente disponible (M)

Alto: Propiedad disponible (A)

Anexo 2: Encuesta realizada a líderes de proyectos de simulación virtual del país.

Encuesta de apoyo a la investigación para la obtención de una referencia para la ingeniería de requisitos en simuladores virtuales.

En los proyectos de Simulación Virtual en los que usted ha participado:

1. ¿Qué nivel de prioridad se le asigna a los requisitos dentro del proceso de desarrollo de software?
 Alta
 Media
 Baja

1. ¿Qué factores han influido negativamente en la calidad del proceso de desarrollo de los requisitos?
 Falta de comunicación con usuarios y/o clientes.
 Requisitos y especificaciones incompletas.
 Requisitos cambiantes.
 Falta de apoyo por parte de los directivos.
 Falta de recursos.
 Expectativas irreales.
 Falta de claridad en los objetivos.
 Estimaciones irreales.
 Falta de coordinación entre entidades desarrolladoras del proyecto.
 Consulta de documentos o búsquedas en internet como fuente de información.
 Otros. ¿Cuáles? _____.

2. ¿Qué actividades relacionadas con los requisitos se acometen durante el proceso de desarrollo de software?
 Obtención de requisitos
 Refinar requisitos
 Analizar requisitos
 Especificar requisitos
 Priorizar requisitos
 Derivar requisitos
 Clasificar requisitos
 Asignar requisitos por subsistema
 Dar seguimiento a los requisitos
 Probar y verificar requisitos
 Validar requisitos

3. ¿Qué técnica(s) utilizan para obtener los requisitos de los usuarios y clientes?
 Entrevistas
 Talleres
 Tormenta de ideas
 Aprendiz
 Storyboards
 Otra(s)
¿Cuál? _____

4. ¿Qué técnica(s) utiliza para especificar requisitos?
 Casos de Uso
 Escenarios
 Pseudocódigo

- Storyboards
 - Modelo Entidad Relación
 - Otra(s). ¿Cuál? _____
5. ¿Qué técnica(s) utiliza para validar los requisitos?
- Listas de chequeo
 - Casos de prueba
 - Construcción de prototipos
 - Otra(s). ¿Cuál? _____
6. ¿Ha empleado herramientas software para llevar a cabo los procesos de Ingeniería de Requisitos?
- Sí
 - No
7. En caso de ser afirmativa la respuesta anterior marque las herramientas utilizadas.
- Suite Rational Rose
 - Visual Paradigm
 - RSA
 - Enterprise Architect
 - Otra(s). ¿Cuál? _____
8. ¿Cómo califica la calidad de los requisitos y sus especificaciones obtenidas?
- Alta
 - Media
 - Baja

Anexo 3: Propuesta de plantilla para la realización de la primera entrevista con el cliente.

Guía de la primera entrevista con el cliente			
Fecha: <i>{Fecha de realización de la entrevista}</i>	Entrevistador: <i>{Nombre del analista que realiza la entrevista }</i>	Entrevistado: <i>{Dirigido a Jefe de Empresa}</i>	Cargo: <i>{Cargo del entrevistado}</i>
Posibles interrogantes			
¿Quién está detrás de la solicitud de este trabajo?			
¿Quién utilizaría la solución?			
¿Cuál sería el beneficio económico del éxito de una solución?			
¿Hay alguna otra alternativa para la solución que necesita?			
¿Cómo caracterizaría una buena salida (resultado) generada para una buena solución?			
¿A qué tipo de problema(s) va dirigida esta solución?			
¿Puede mostrarme (o describirme) el entorno en que se utilizará la solución?			
¿Hay aspectos o restricciones especiales del rendimiento que afecten a la manera de enfocar la solución?			
¿Es usted la persona adecuada para responder a estas preguntas? ¿Sus respuestas son oficiales?			
¿Hay alguien más que pueda proporcionar información adicional?			
¿Hay algo más que debería preguntarle?			

Anexo 4: Propuesta de plantilla para la descripción textual de un caso de uso.

Descripción Textual de Caso de Uso	
Caso de uso:	<Insertar nombre del caso de uso>
Actor(es):	<Insertar nombre de actor(es) que intervienen en el caso de uso>
Resumen:	<Insertar una breve descripción del caso de uso, especificando cómo y cuándo comienza y termina el caso de uso (primera y última acción a ejecutar)>
Precondiciones:	<Insertar condición que debe cumplirse para que se inicie el caso de uso (estado inicial)>
Referencias:	<Insertar requisitos asociados al caso de uso>
Prioridad:	<Insertar orden de prioridad del caso de uso >
Flujo normal de eventos	
Acción del actor	Respuesta del sistema
<Insertar acción realizada por el actor>	<Insertar respuesta emitida por el sistema>
Flujo alterno	
	<Insertar respuesta alternativa del sistema>
Poscondiciones:	<Insertar condiciones en que queda el caso de uso al concluir su ejecución (estado final)>

Anexo 5: Propuesta de plantilla para la descripción textual de un escenario.

Descripción de un escenario
Título: <Insertar título que identifica al escenario>
Objetivo: <Insertar objetivo que satisface el escenario>
Resumen: <Insertar un breve resumen de lo representado en el escenario>
Actores: <Insertar el nombre de los actores del sistema que están involucrados en el escenario>
Episodio: <Insertar el curso principal de acciones que se ejecutan para satisfacer el objetivo del escenario>
Excepciones: <Insertar excepciones que señalan una limitación para lograr el objetivo del escenario>
Precondiciones: <Insertar breve descripción del estado del sistema antes de ejecutado el escenario>
Poscondiciones: <Insertar Insertar breve descripción del estado del sistema después de ejecutado el escenario>
Restricción: <Referenciar elementos que caractericen requisitos no funcionales aplicados al contexto>
<Insertar storyboard y videos que permitan visualizar los requisitos involucrados en el escenario>

Anexo 6: Propuesta de plantilla para la generación de un caso de prueba.

Id de la funcionalidad	Funcionalidad	Variable 1	...	Variable n	Respuesta del Sistema	Resultado de la prueba
<Se asigna un identificador a cada una de las funcionalidades extraídas>	<Se coloca el nombre de cada funcionalidades >	<Se especifica los datos o variables necesarias para la prueba>	<Se describe la posible respuesta a emitir por el sistema y/o se coloca una imagen de la pantalla obtenida como respuesta del sistema en el prototipo no funcional realizado a partir de sus uso como técnica de validación de requisitos>	<Se especifican los resultados obtenidos al realizar la prueba>

Anexo 7: Primera encuesta realizada a los Expertos para obtener su nivel de competencia.

Encuesta No.1 a Expertos

Competencia de Expertos

Nombre: _____

Fecha: _____

Años de experiencia: _____

Cargo o Responsabilidad: _____

Grado científico: _____

Entidad: _____

Nota:

El objetivo de esta encuesta es solamente investigativa. El responsable de esta encuesta se compromete a mantener total privacidad de la información recopilada.

Pregunta 1

Marque con una X, en una escala del 1 al 10 el valor que se corresponde con el grado de conocimiento o información que usted considera que tiene respecto al desarrollo de Simuladores Virtuales y sobre la Ingeniería de Requisitos. El número 1 indica que no tiene ningún conocimiento sobre el tema y 10 indica que tiene pleno conocimiento sobre él.

	1	2	3	4	5	6	7	8	9	10
SV										
IR										

Pregunta 2

Señale con una X el nivel de influencia que ha tenido cada una de las fuentes indicadas en su conocimiento sobre el desarrollo de Simuladores Virtuales y la Ingeniería de Requisitos.

FUENTES DE ARGUMENTACION	Grado de influencia de cada una de las fuentes en sus criterios.		
	A (alto)	M (medio)	B (bajo)
Análisis teóricos realizados por usted			
Su experiencia obtenida			
Trabajos de autores nacionales			
Trabajos de autores extranjeros			
Su propio conocimiento del estado del problema en el extranjero			
Su intuición			

Anexo 8: Segunda encuesta realizada a los Expertos para evaluar el valor práctico de la aplicación de la propuesta y el cumplimiento de los objetivos propuestos.

Encuesta No.2 a Expertos

Valor práctico de la aplicación de la propuesta y cumplimiento de los objetivos.

Pregunta 1: Evalúe la aplicación práctica de la propuesta según las siguientes interrogantes.

Preguntas	Sí	No	No sé
¿Es necesario la definición de técnicas adecuadas para llevar a cabo los procesos de Ingeniería de Requisitos en proyectos de Simulación Virtual?			
¿Considera posible el empleo de las técnicas para la IR propuestas en el desarrollo de Simuladores Virtuales?			
¿Considera conveniente el uso de las herramientas propuestas para desarrollar los procesos de Ingeniería de requisitos durante el desarrollo de Simuladores Virtuales?			
	Muy importante	Medianamente importante	No importante
¿Considera importante la utilización de la propuesta en los proyectos de producción de Simuladores Virtuales?			

Pregunta 2: Evalúe el cumplimiento de los objetivos definidos en la propuesta teniendo en cuenta las siguientes interrogantes.

Preguntas	Sí	No	No sé
¿Considera que el empleo de las técnicas y herramientas propuestas en los proyectos de producción de Simuladores Virtuales mejoraría la comunicación entre equipo de desarrollo y clientes y otros involucrados?			
¿Considera que el empleo de las técnicas y herramientas propuestas en los proyectos de producción de Simuladores Virtuales permitiría un mejor desarrollo del proceso de obtención o captura de requisitos?			
¿Considera que el empleo de las técnicas y herramientas propuestas en los proyectos de producción de Simuladores Virtuales ayudaría a obtener una formalización y especificación de requisitos con suficiente calidad y detalles de manera que el personal técnico pueda proceder con el diseño y construcción de la solución?			
¿Considera que el empleo de las técnicas y herramientas propuestas en los proyectos de producción de Simuladores Virtuales permitiría gestionar o administrar los requisitos eficientemente?			

Anexo 9: Encuesta realizada al equipo de desarrollo del proyecto simulador quirúrgico para valorar la calidad de la especificación de requisitos realizada haciendo uso de las técnicas y herramientas propuestas.

Encuesta para valorar la calidad de la especificación de requisitos realizada en el simulador quirúrgico haciendo uso de las técnicas y herramientas propuestas.

Teniendo en cuenta el uso de las nuevas técnicas utilizadas para el tratamiento de los requisitos en el proyecto y la definición de los atributos para medir la calidad de una especificación de requisitos dada al final de este documento, responda las siguientes interrogantes.

Correctitud

1. ¿En qué medida considera correcta la especificación de requisitos realizada?

Alta Media Baja

No ambigüedad

2. ¿En qué medida considera clara/no ambigua la especificación de requisitos realizada?

Alta Media Baja

Compleitud

3. ¿En qué medida considera completa la especificación de requisitos realizada?

Alta Media Baja

Concisión

4. ¿En qué medida considera concisa la especificación de requisitos realizada?

Alta Media Baja

Verificabilidad

5. ¿En qué medida considera verificable la especificación de requisitos realizada?

Alta Media Baja

Trazabilidad

6. ¿En qué medida considera posible la trazabilidad de la especificación de los requisitos?

Alta Media Baja

Priorización según su importancia y/o estabilidad

7. ¿Con la especificación realizada se tiene el nivel de prioridad de cada requisito para abordar su desarrollo?

Sí No

Aspectos claves

8. ¿Tuvo necesidad de auxiliarse de búsquedas en Internet o consultar documentos por no ser suficiente la especificación de los requisitos dada?

Sí No

9. ¿En qué medida considera usted que se hayan disminuido los cambios en los requerimientos y sus especificaciones?

Alta Media Baja

10. ¿En qué medida las técnicas propuestas mejoraron la comunicación con el equipo de desarrollo del proyecto?

Alta Media Baja

Especificación correcta: todos los requisitos contenidos representan algo que es requerido para la construcción del sistema y no hay errores que afecten al diseño". La corrección depende del usuario final del sistema, que es el indicado para decidir si la especificación es correcta o no.

Especificación no-ambigua: un requisito puede ser interpretado de formas diversas por diferentes personas. El estándar IEEE 830 establece que un requisito es no-ambiguo "si y sólo si, puede estar sujeto a una única interpretación."

Especificación completa: una especificación es completa "si y sólo si, describe todos los requisitos relevantes para el usuario, incluyendo requisitos asociados con funcionalidad, rendimiento, restricciones de diseño, hardware o interfaces externas."

Especificación consistente: una especificación es consistente "si y sólo si, no hay ningún subconjunto de requisitos descrito dentro de ella que esté en conflicto con cualquier otro." El mayor problema en este ámbito es la repetición de información en los requisitos (requisitos repetitivos y redundantes).

Especificación verificable: una especificación es verificable "si lo son cada uno de los requisitos constituyentes". A su vez, se considera que un requisito individual es verificable "si existe un proceso acotado (en plazo y presupuesto) que permita determinar que el sistema construido satisface lo descrito en el propio requisito". La descripción detallada y prueba de los requisitos una vez implementados ayudan considerablemente a su verificación.

Especificación trazable: una especificación se considera trazable si el origen de cada requisito individual está claro y existe algún mecanismo que permita rastrear la incorporación de dicho requisito en el resto de las actividades del ciclo de vida.

Requisitos ordenados por importancia y/o estabilidad: permite establecer prioridades a la hora de abordar el desarrollo, contribuyendo al cumplimiento de los presupuestos y tiempo de entrega del producto.