

**HERRAMIENTA PARA EL ANÁLISIS DE REQUERIMIENTOS
DENTRO DE LA PEQUEÑA EMPRESA DESARROLLADORA DE
SOFTWARE EN BOGOTÁ**

ANTONIO NICOLÁS CAMACHO ZAMBRANO

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA DE SISTEMAS
JUNIO, BOGOTÁ D.C.**

2005

**HERRAMIENTA PARA EL ANÁLISIS DE REQUERIMIENTOS
DENTRO DE LA PEQUEÑA EMPRESA DESARROLLADORA DE
SOFTWARE EN BOGOTÁ**

ANTONIO NICOLÁS CAMACHO ZAMBRANO

**Proyecto De Grado Presentado Para Optar Al Título De Ingeniero
De Sistemas**

**Ingeniero Miguel Eduardo Torres Moreno MSc.
Profesor Investigador Área de Ingeniería de Software
Director de la Investigación**

**PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERÍA
CARRERA DE INGENIERÍA DE SISTEMAS
JUNIO, BOGOTÁ D.C.**

2005

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA DE SISTEMAS

Rector Magnífico: Padre Gerardo Remolina Vargas S.J.

Decano Académico Facultad de Ingeniería: Ingeniero Francisco Javier
Rebolledo Muñoz

Decano del Medio Universitario Facultad de Ingeniería: Padre Antonio
José Sarmiento Novoa S.J.

Director Carrera de Ingeniería de Sistemas: Ingeniera Hilda Cristina
Chaparro López

Director Departamento de Ingeniería de Sistemas: Ingeniero Germán
Alberto Chavarro Flórez

Nota de Aceptación

Firma del Director del Proyecto

Firma del Jurado

Firma del Jurado

BOGOTÁ D.C., JUNIO DE 2005

Artículo 23 de la Resolución No. 1 de Junio de 1946:

"La Universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado.

Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque no contengan ataques o polémicas puramente personales. Antes bien, que se vean en ellos el anhelo de buscar la verdad y la Justicia"

DEDICATORIA:

Este trabajo de grado es el esfuerzo conjunto de muchas personas. Por tanto, a todas ellas por su apoyo y amabilidad, les dedico este gran logro de mi vida. A mis padres, Consuelo y Antonio, que con su esfuerzo, cariño, dedicación y fortaleza; me han permitido convertirme en la persona que soy. Muchas Gracias.

A mi hermana, Carolina, quien en los momentos difíciles siempre fue una luz que iluminó la habitación oscura. A mis amigos, que saben quienes son; por su apoyo, por su amistad, por sus risas y por la alegría que siempre han compartido conmigo, muchas gracias.

A mis abuelos maternos, Lucía y Rafael (q.e.p.d.), por su cariño, apoyo y enorme interés en mí, y mi éxito académico y profesional; de igual forma a mis abuelos paternos, Ana (q.e.p.d) y Antonio (q.e.p.d), cuyos valores e ideales acerca de la dedicación y determinación estuvieron presentes todos los días de mi vida a través de los ojos de mi padre.

A Maribel, por ser una parte muy importante de mi vida, agradezco tu sonrisa y apoyo a través de todos estos momentos. Gracias por todo este tiempo. A todas aquellas personas que creyeron en mí, y aún hoy lo siguen haciendo, muchas gracias.

Finalmente, dedico el fruto de este trabajo a Dios, porque siempre he sentido su apoyo, y sin Él, nada de esto hubiera sido posible.

AGRADECIMIENTOS:

Quisiera agradecer a mi director de proyecto de investigación, Miguel Eduardo Torres Moreno, por su apoyo incondicional, entusiasmo, guía, paciencia y perseverancia, durante este año y medio de investigación. Le agradezco por ser un gran educador y principalmente, una excelente persona. Así mismo quiero expresar mi reconocimiento a todos mis profesores por sus enseñanzas y experiencias, apoyo y dedicación en estos años de estudio.

Igualmente, deseo expresar mi gratitud a todas aquellas empresas que participaron en la investigación, por su interés y tiempo. Agradezco particularmente a SWONE y AXESNET, empresas que me brindaron acceso a la información de los proyectos de software con los cuales realicé la investigación. Agradezco en especial a Ana María Rodríguez de SWONE y Bibiana Alexandra Lara de AXESNET por participar del proyecto con su tiempo y retroalimentación, por abrirme las puertas en su correspondiente empresa. También deseo agradecer a Dawid Junnco por participar y coordinar la comunicación con la empresa SEFT, y permitirme la obtención de información.

Agradezco de igual forma, a la Ingeniera y docente María Mercedes Corral, por su colaboración y su perspectiva, experiencia y guía para conducir la investigación en las primeras etapas de la misma. También agradezco al Ingeniero y docente Rafael González Rivera por su colaboración y comentarios oportunos alrededor de la investigación.

CONTENIDO

INTRODUCCIÓN.....	15
1. JUSTIFICACIÓN.....	20
2. OBJETIVOS.....	21
2.1 OBJETIVO GENERAL	21
2.2 OBJETIVOS ESPECÍFICOS	21
3. MARCO TEÓRICO.....	23
3.1 DEFINICIONES BÁSICAS.....	23
3.1.1 ¿Qué es un requerimiento?.....	23
3.1.2 ¿Cómo se clasifican los requerimientos?.....	24
3.1.3 Niveles de descripción de un requerimiento.	25
3.1.4 Características de un Buen Requerimiento.....	28
3.2 GENERALIDADES DE LA INGENIERÍA DE REQUERIMIENTOS.....	31
3.2.1 ¿Qué es la Ingeniería de Requerimientos?.....	31
3.2.2 Herramientas, técnicas y software.....	47
3.2.3 Consideraciones para el proceso de ingeniería de requerimientos.....	50
4. LA PEQUEÑA EMPRESA DESARROLLADORA DE SOFTWARE. 52	
4.1 LA SITUACIÓN ACTUAL	52
4.1.1 Economía.....	54
4.1.2 La Ingeniería de Requerimientos.....	55
5. EL PROCESO DE ANÁLISIS DE REQUERIMIENTOS.....	59
5.1 MODELOS TRADICIONALES	59
5.1.1. Descomposición Funcional.....	59
5.1.2. Análisis Estructurado.....	60
5.1.3. Especificación Operacional	61
5.1.4. Análisis Orientado a Objetos	62
5.2 OTROS MODELOS	63
6. PROPUESTA CONCEPTUAL	66
6.1 INGENIERÍA DE REQUERIMIENTOS DEL MODELO	66
6.1.1 Recolección de requerimientos.....	69
6.1.2 Análisis de requerimientos.....	80
6.2 ESTRUCTURA COMPUTACIONAL DEL MODELO	84
6.2.1 Definición del Modelo.....	84

6.3	MAXIMIZANDO LA INFORMACIÓN DE UN REQUERIMIENTO	93
6.3.1	<i>Análisis De Riesgos.</i>	93
6.3.2	<i>Estimaciones.....</i>	98
6.3.3	<i>Elementos Especiales y Fusión de Elementos Interconectados o Interdependientes.</i>	102
6.4	IMPLEMENTACIÓN DE LA FUNCIONALIDAD DEL MODELO.....	105
6.4.1	<i>Componentes Fuertemente Conectados.</i>	105
6.4.2	<i>Organización topológica.</i>	106
6.4.3	<i>Búsqueda en profundidad.....</i>	106
6.5	CONSIDERACIONES DEL MODELO	107
6.6	HERRAMIENTA COMPUTACIONAL.....	110
6.6.1	<i>Precondiciones de uso.....</i>	110
6.6.2	<i>Objetivos de la herramienta.....</i>	111
6.6.3	<i>Implicaciones de uso.....</i>	112
7.	ESTUDIOS.....	114
7.1	EVALUACIÓN DEL MODELO	114
7.1.1	<i>Criterios de evaluación</i>	114
7.1.2	<i>Características a evaluar en el modelo.....</i>	114
7.1.3	<i>Ejecución de la evaluación</i>	115
7.2	EVALUACIÓN DE LA HERRAMIENTA	117
7.3	CASO DE ESTUDIO.....	118
7.3.1	<i>Entradas de la herramienta</i>	119
7.3.2	<i>Operaciones y análisis realizados.....</i>	120
7.3.3	<i>Análisis del caso de estudio</i>	125
7.4	APLICABILIDAD DEL MODELO Y ANÁLISIS DE RESULTADOS...	125
7.4.1	<i>El análisis de riesgos asociados a un requerimiento como factor para el incremento de la calidad en desarrollos de software.....</i>	126
7.4.2	<i>¿Cómo aplica a las empresas?.....</i>	132
8.	RESULTADOS DE LA INVESTIGACIÓN.....	134
9.	CONCLUSIONES.....	136
	BIBLIOGRAFÍA.....	140

LISTA DE TABLAS

		Pág.
Tabla 1	Herramientas que se utilizan para las diferentes fases de la ingeniería de requerimientos.	48
Tabla 2	Características de las herramientas de Administración de requerimientos más utilizadas del mercado	49
Tabla 3	Relación de CMM con el estado del proceso de requerimientos al interior de una empresa	57
Tabla 4	Plantilla para la descripción de requerimientos funcionales	74
Tabla 5	Plantilla para la descripción de requerimientos no funcionales	78
Tabla 6	Tabla en la cual se tabulan las entradas y salidas por requerimiento y se anexaron los atributos de riesgo y dificultad estimados.	95
Tabla 7	Clasificación de los requerimientos funcionales en orden de riesgo creciente.	97
Tabla 8	Especificación de requerimientos de software proyecto MCI	119
Tabla 9	Traducción de la especificación de requerimientos en la correspondiente entrada a la herramienta.	120
Tabla 10	Datos correspondientes a la definición de requerimientos de la figura 24.	131
Tabla 11	Orden por factores de riesgo de los requerimientos de la tabla 10.	132

LISTA DE FIGURAS

		Pág.
Figura 1	Relación entre los diferentes artefactos, tipos de requerimientos y atributos de los mismos en un proyecto de software	28
Figura 2	Estructura de la Ingeniería de Requerimientos	34
Figura 3	Proceso de la Ingeniería de Requerimientos.	36
Figura 4	Actividades del Ciclo de Vida de Los Requerimientos	37
Figura 5	Proceso de Ingeniería de software definido para esta investigación.	68
Figura 6	Modelo de proceso para Recolección de Requerimientos.	70
Figura 7	Modelo de proceso para análisis de requerimientos.	81
Figura 8	Representación básica del modelo propuesto, donde los nodos del grafo son los requerimientos funcionales, y los trazados entre ellos son las relaciones de dependencia.	86
Figura 9	Representación de la relación de dependencia (Precedencia) entre dos requerimientos.	87
Figura 10	Representación de la relación de dependencia mutua (Interdependencia) entre dos requerimientos.	87
Figura 11	Estructura de servicios y funcionalidades que se presentan en un sistema modular. Muestra como los servicios que se encuentran más arriba en la estructura dependen de los servicios que se encuentran debajo de ellos.	89
Figura 12	Desarrollo incremental de los requerimientos. El conjunto de requerimientos crece, de manera que se puede presentar dependencia entre los elementos de alguno de los subconjuntos.	89
Figura 13	Estructura Jerárquica de requerimientos funcionales. Los requerimientos de alto nivel se descomponen hasta ser específicos.	91

Figura 14	Requerimientos y sus dependencias derivadas del negocio para el sistema de una calculadora. En este caso las funcionalidades de Multiplicación y División son traducidas como extensiones de otras funcionalidades.	91
Figura 15	Pequeño conjunto de requerimientos funcionales y sus dependencias para el sistema de subastas.	94
Figura 16	Diagramas de Actividades, correspondientes al de desarrollo de un conjunto de requerimientos de la figura (14).	99
Figura 17	Posibles casos de interdependencia. Interdependencia cíclica (a) y (c). Interdependencia básica (b) y (c).	103
Figura 18	Proceso de ejecución del modelo.	111
Figura 19	Grafo Inicial de la especificación sin operar	121
Figura 20	Grafo diferenciado por sus componentes fuertemente conectados	122
Figura 21	Grafo reducido a través de la fusión de los requerimientos que conforman componentes fuertemente conectados	123
Figura 22	Grafo diferenciado de acuerdo a los módulos de software a los cuales pertenece cada requerimiento	124
Figura 23	Relación de las variables riesgo y predictibilidad en el desarrollo de software	126
Figura 24	Diferentes Estructuras que se conforman de una definición de requerimientos, traducida al modelo y sus correspondientes operaciones	128

LISTA DE ANEXOS

- | | |
|---------|---|
| Anexo A | Análisis de resultados Encuesta pequeña empresa desarrolladora de software en Bogotá D.C. |
| Anexo B | Especificación de requerimientos de software para la herramienta de software. |
| Anexo C | Documento de diseño de la herramienta de software. |
| Anexo D | Documento de pruebas realizadas sobre la herramienta de software. |
| Anexo E | Documento de pruebas de campo realizadas al modelo y la herramienta de software. |
| Anexo F | Características técnicas de la herramienta y manual de usuario. |

GLOSARIO

- A -

Actor: Es una entidad externa al sistema que se modela y que puede interactuar con él.

Atributo De Requerimiento: Cada uno de los atributos que se especifican para un requerimiento de software.

- D -

Diagrama De Actividad: Diagrama que hace parte del lenguaje UML. Permite modelar el flujo entre un conjunto de objetos que cooperan entre sí. Son similares a los diagramas de flujo de otras metodologías diferentes a la orientada a objetos.

Definición de Requerimientos de Software: Documento, o conjunto de documentos en el cual se consignan de manera preliminar a la fase de especificación los requerimientos de software de un sistema.

Documento De Casos De Uso: Documento que contiene la especificación de casos de uso definidos para un sistema. Así mismo, puede contener un diagrama de casos de uso.

Grafo: Representación gráfica y matemática de los datos de una situación particular, a través de elementos particulares llamados nodos y enlaces.

- M -

Modelo Conceptual: Modelo que define vistas que representan la organización de los componentes, agentes o elementos de software que participan para lograr la funcionalidad requerida por el sistema.

- S -

Stakeholder: Persona interesadas o involucradas en el desarrollo de un sistema, bajo una perspectiva. Esta puede ser económica o relacionada otro beneficio por el desarrollo del sistema.

- U -

UML (Unified Modeling Language): Lenguaje De Modelamiento Unificado. Es un lenguaje para especificar, construir, visualizar y documentar los artefactos o ítems de un sistema o software orientado a objetos (OO).

INTRODUCCIÓN

En el ámbito de los proyectos de software siempre ha existido una constante preocupación acerca del posible éxito de los mismos, y una de las inquietudes más importantes de la Ingeniería de Software es el garantizar ese éxito. Así mismo, a través de la experiencia, se han identificado ramas y tópicos de especial relevancia dentro del desarrollo de software, y cuyo tratamiento es de suma importancia si se desea obtener éxito dentro de este campo.

Uno de estos tópicos es el concerniente a los requerimientos. Estos, sometidos a diferentes análisis y debates; se han mantenido en el ojo del huracán debido a su fuerte repercusión dentro del éxito o fracaso de proyectos de software. Recientemente, en la publicación emitida por la revista electrónica **The Rational Edge**, acerca del estado y las prácticas recomendadas para el desarrollo de software y sistemas, se le otorga una sección a los requerimientos en la cual se enmarca reiterativamente que el precisarlos es una parte esencial de la fórmula para los proyectos de software exitosos¹.

Este enfoque se puede apreciar en la importancia que la corporación IBM y otras compañías han otorgado al desarrollo de herramientas para el tratamiento, administración y desarrollo de requerimientos. Es así que para nombrar una muestra, en el mercado podemos conseguir

¹**MCEWEN**, Scott. Requirements: An Introduction. The Rational Edge [online]. Abril 2, 2004 [consultada mayo de 2005]. Disponible en Internet: <<http://www-106.ibm.com/developerworks/rational/library/4166.html>>.

herramientas para administración de requerimientos como lo son **Rational Requisite Pro®**, **Web Requisite®** o **CaliberRM®**; herramientas **CASE** que permiten especificar requerimientos como **Together®**, y otras herramientas de compañías que realizan soluciones adaptables a las necesidades requeridas.

Así mismo, y desde otra perspectiva, se han difundido ampliamente otros elementos como los son lenguajes de especificación de requerimientos, **frameworks** para el análisis de requerimientos, y metodologías que indican como llevar a cabo los procesos de requerimientos dentro de los desarrollos de software. Ejemplos de esto lo constituyen el **Zachman Framework**², una herramienta utilizada para analizar las características y requerimientos envueltos dentro de la arquitectura de cualquier sistema de información; y los documentos sugeridos por el RUP(**Rational Unified Process**)³ para manejo y administración de requerimientos.

Por otra parte, el desarrollo de software se perfila como una de las industrias con mayor proyección de crecimiento en nuestro país. Así mismo, el Gobierno Nacional está impulsando éste como una alternativa para el desarrollo nacional, y ha brindado grandes prebendas con el fin

²**THE ZACHMAN INSTITUTE FOR FRAMEWORK ADVANCEMENT**, ZIFA. Página oficial [consultada enero de 2004]. Disponible en Internet: < <http://www.zifa.com> >

³**RATIONAL CORPORATION** IBM. Página de Rational Unified Process [consultada enero de 2004] Disponible en Internet:
< <http://www-306.ibm.com/software/awdtools/rup/> >

de impulsar el progreso y maduración del gremio⁴. Este hecho se relaciona mucho con el proceso evolutivo que ha tenido el desarrollo de proyectos de software en el mundo, incitándonos así a aplicar, recrear y construir las mejores prácticas de desarrollo de software como ventaja competitiva dentro del mercado mundial.

Sin embargo, los estudios que se tienen acerca de las metodologías y prácticas que se realizan actualmente en Colombia acerca del desarrollo de software, no son muy claros ni especifican de manera profunda el tratamiento de requerimientos que se le da a los proyectos recientemente realizados. Los aspectos más relevantes que se han tratado dentro de estos estudios están relacionados principalmente con temas como los estados financieros de la industria, las metodologías para desarrollo de aplicaciones, los paradigmas de programación, la gestión de proyectos de software y la evolución de las plataformas en el proceso de desarrollo de software⁵; pero no se profundiza en el área de los requerimientos.

A nivel educativo, se puede apreciar que ya se han realizado primeras aproximaciones al estudio de requerimientos dentro del ámbito social y educativo. Dos ejemplos de esto son las realizadas en dos trabajos de grado pertenecientes a estudiantes de la Universidad Javeriana y la

⁴**FEDERACIÓN COLOMBIANA DE LA INDUSTRIA DEL SOFTWARE Y TECNOLOGÍAS INFORMÁTICAS RELACIONADAS**, Fedesoft. Página oficial [consultada enero de 2004]. Disponible en Internet: <<http://www.fedesoft.com>>

⁵**CORTÉS B.**, Gloria C. Los retos actuales para nuestra industria de software En: Sistemas: El entorno colombiano en procesos modernos de desarrollo de software, N° 86, agosto-octubre de 2003 [consultada mayo de 2005]. Disponible en Internet: <<http://sitio.acis.org.co/Paginas/publicaciones/editorial86.html>>

Universidad de los Andes respectivamente. El primero de estos trabajos, es el desarrollo de una herramienta para la administración de requerimientos de proyectos de software⁶. La otra aproximación, es un estudio de la ingeniería de requerimientos desde una perspectiva social⁷.

El campo de aplicación de los requerimientos es muy amplio, y áreas como lo son el análisis y verificación de requerimientos, son elementos de vital importancia dentro del proceso de desarrollo de software que tienen un campo de aplicación y acción que a la fecha ha sido muy poco explorado desde la perspectiva de herramientas computacionales que permitan llevar a cabo este proceso.

Si nos centramos en la situación actual colombiana, es fácil denotar que las pequeñas y medianas empresas son el motor del país. Según un estudio del Centro de Investigaciones de la Escuela de Finanzas y Comercio Exterior de la Universidad Sergio Arboleda, estas generan más del 50% del empleo nacional, significan el 36% del valor agregado industrial, el 92% de los establecimientos comerciales y el 40% de la producción total del país⁸.

⁶**ROJAS MARIN**, Obdulio. Herramienta para la administración de los requerimientos en los proyectos de ingeniería de software y procesos productivos. Bogotá, 2000. Trabajo de Grado (Ingeniero de Sistemas). Pontificia Universidad Javeriana. Facultad de Ingeniería. Área de Ingeniería de Software.

⁷**BARRERA FUENTES**, William Eduardo. Ingeniería de requerimientos desde una perspectiva social. Bogotá, 2002. Trabajo de Grado (Ingeniero de Sistemas). Universidad de los Andes. Facultad de Ingeniería. Área de Ingeniería de Software.

⁸**PUYANA SILVA**, David Guillermo. La Problemática De Las PYMES en Colombia: Internacionalizarse o Morir [online]. Centro de investigación de de la Escuela de finanzas y Comercio Exterior, Universidad Sergio Arboleda. Agosto 2002 [consultada

Por tanto, la pequeña empresa es un campo de investigación muy importante para el desarrollo de nuestro país. Así mismo, la ciudad de Bogotá es de especial importancia, no sólo por ser la capital de Colombia, sino por poseer el mayor porcentaje de empresas desarrolladoras de software del país; lo que la convierte en un foco de especial atención a la hora de analizar los problemas de desarrollo de software en nuestro contexto.

Con fundamento en lo anterior, se puede plantear la interrogante de generar elementos que favorezcan a la pequeña empresa, en áreas relacionadas con el manejo de los requerimientos, especialmente, las que no han sido ampliamente exploradas. Son estos pequeños elementos, los que pueden brindar ventajas competitivas a la pequeña empresa y serán de vital importancia en su futuro inmediato, y la responsabilidad que tenemos es el permitir que estos pequeños elementos estén a la mano de quienes más los necesitan.

1. JUSTIFICACIÓN

Esta investigación tiene como fundamento la necesidad de explorar un vacío en dos aspectos de la realidad contemporánea dentro de los proyectos de software: el primero, es el hecho de que la pequeña empresa desarrolladora de software necesita ventajas competitivas para llevar a cabo sus procesos de análisis de requerimientos; el segundo, es que este proceso puede ser mejorado con herramientas específicamente enfocadas al aspecto del análisis de requerimientos, y la mayoría de las PyMEs no están en capacidad de adquirir una solución sofisticada dentro de este campo.

Desde estas perspectivas, la investigación se justifica debido a que la necesidad de llevar a cabo un estudio del estado del proceso de análisis de requerimientos puede ser una contribución al desarrollo de buenas prácticas de ingeniería de software dentro de los proyectos en Bogotá, y a su vez en Colombia, y esto permitiría desarrollar elementos de ayuda para afrontar la problemática nacional de fortalecer la industria colombiana frente a la extranjera.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Contribuir con el mejoramiento del proceso de análisis de requerimientos en proyectos que involucren desarrollo de Software aplicado a las pequeñas empresas que tengan como fin el desarrollo de software en la ciudad de Bogotá.

2.2 OBJETIVOS ESPECÍFICOS

- Modelar y elaborar el prototipo de una herramienta computacional como elemento de ayuda dentro del proceso de análisis de requerimientos para proyectos de software.
- Identificar un proceso que facilite la recolección de datos que se manejen dentro del proceso de análisis de requerimientos.
- Utilizar este proceso de recolección como base de la información que se maneja dentro de la herramienta computacional.
- Identificar y definir un proceso de análisis de requerimientos dentro de los proyectos de software.
- Utilizar este proceso como base para la elaboración del prototipo.

- A través de esta herramienta agilizar y facilitar el proceso de análisis de requerimientos en los proyectos de Software.
- Contribuir al incremento de la calidad de los procesos de desarrollo de software que se llevan a cabo dentro de la pequeña empresa desarrolladora de software.
- Profundizar mis conocimientos en el área de la ingeniería de requerimientos y particularmente en el área de análisis de requerimientos.

3. MARCO TEÓRICO

3.1 DEFINICIONES BÁSICAS

3.1.1 ¿Qué es un requerimiento?

El concepto fundamental para entender los elementos que componen este trabajo de grado, es el concepto de *requerimiento*. La definición más general alrededor de esta noción es la que brinda el Instituto de Ingeniería Electrónica y Eléctrica (**IEEE**)⁹:

- (1) Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.
- (2) Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.
- (3) Una representación documentada de una condición o capacidad documentada como las descritas en (1) y (2).

Esta definición expresa la perspectiva clásica de los requerimientos como elementos de un producto, o criterios para acuerdos. Sin embargo, otros autores son más específicos frente a la relación de los

⁹**INSTITUTE FOR ELECTRONICS AND ELECTRICAL ENGINEERS.** Glosario estándar de la terminología de la ingeniería de software estándar 610.12-1990. s.l.: La institución, 1997.

requerimientos con relación al sistema que van a representar: "... Los requerimientos son una especificación de lo que debe ser implementado. Estos son descripciones de cómo el sistema se debe comportar, de las propiedades y atributos del mismo. Deben ser una restricción del proceso de desarrollo del sistema..."¹⁰.

Esta definición está muy ligada a lo que constituye el desarrollo de un sistema. Otra definición, que justifica la necesidad de los requerimientos frente a las perspectivas del usuario y del sistema es: "... Un requerimiento es algo que el producto debe hacer o una cualidad que el producto debe tener. Un requerimiento existe ya sea porque el tipo de producto demanda ciertas necesidades o cualidades, o porque el cliente desea que ese requerimiento sea parte del producto entregado..."¹¹.

3.1.2 ¿Cómo se clasifican los requerimientos?

Existen diferentes clasificaciones de los requerimientos, representativas de distintos autores; sin embargo, en este marco teórico se hará referencia a una de las clasificaciones más aceptadas. Esta clasificación se relaciona directamente con la noción de sistema o solución basada en software, por tanto se enfoca a establecer y diferenciar las propiedades de los requerimientos dentro de estos sistemas.

3.1.2.1 Requerimientos funcionales. Los requerimientos funcionales son aseveraciones de los servicios que el sistema debe proveer, como el

¹⁰**SOMMERVILLE**, Ian y **SAWYER**, Peter. Requirements engineering: A good practice guide. 3 ed. Chinchester, Inglaterra: John Wiley & Sons Ltd., 2000.

¹¹**ROBERTSON**, Suzanne y **ROBERTSON**, James. Mastering the requirements process. Londres: Addison - Wesley, 1999.

sistema debe reaccionar a entradas particulares y como el sistema debe comportarse bajo situaciones particulares. En algunos casos los requerimientos funcionales deben describir de manera explícita, lo que el sistema no debe hacer¹².

3.1.2.2 Requerimientos no funcionales. Estos requerimientos son restricciones sobre los servicios y funcionalidades ofrecidos por el sistema. Estos incluyen restricciones en el tiempo que se debe demorar un proceso, restricciones sobre el proceso de desarrollo y estándares. Los requerimientos no funcionales aplican usualmente sobre el sistema como un todo. Estos normalmente no aplican a características o servicios particulares del sistema¹³.

3.1.2.3 Requerimientos de dominio. Estos son requerimientos que provienen del dominio de aplicación del sistema y reflejan características y restricciones de ese dominio. Estos pueden ser funcionales o no funcionales¹⁴.

3.1.3 Niveles de descripción de un requerimiento.

Los niveles de descripción de un requerimiento permiten hacer una clara separación entre los diferentes tipos de requerimientos que se pueden concebir en un documento de requerimientos. Son necesarios para evitar errores y mejorar la descripción de los mismos. El clasificar los requerimientos en estos niveles facilita su entendimiento y su descripción.

¹²**SOMERVILLE**, Ian. Ingeniería de software. 7 ed. México: Addison – Wesley, 2004.

¹³Ibid.

¹⁴Ibid.

Los diferentes niveles de descripción son útiles porque comunican la información a diferentes tipos de lectores¹⁵.

3.1.3.1 Descripción a nivel de negocio. Se llaman requerimientos del negocio a aquellos requerimientos que representan objetivos de alto nivel para la organización o el cliente que requiere el producto¹⁶.

Estos requerimientos son la necesidad principal por la cual se empieza la construcción o mejora del producto. Estos requerimientos se caracterizan por ser descritos de manera muy generalizada en términos de beneficios o necesidades de la organización; y se expresan en un lenguaje *natural*¹⁷. En ocasiones son llamados los objetivos del software.

3.1.3.2 Descripción a nivel de usuario. Los requerimientos que describen tareas que los usuarios deben estar en capacidad de cumplir con el producto de software que se está describiendo, son conocidos como requerimientos del usuario¹⁸.

Estos requerimientos son descritos con frases usando lenguaje natural complementado con diagramas, describiendo las expectativas acerca de

¹⁵**SOMERVILLE**, Op. cit.

¹⁶**WIEGERS**, Karl. Software Requirements. 2 ed. Washington: Microsoft Press, 2003.

¹⁷Se considera lenguaje natural al lenguaje utilizado a diario entre los integrantes de la organización. Se caracteriza por estar orientado a una descripción más humana y generalizada, y no regido por consideraciones técnicas específicas.

¹⁸**WIEGERS**, Op. cit.

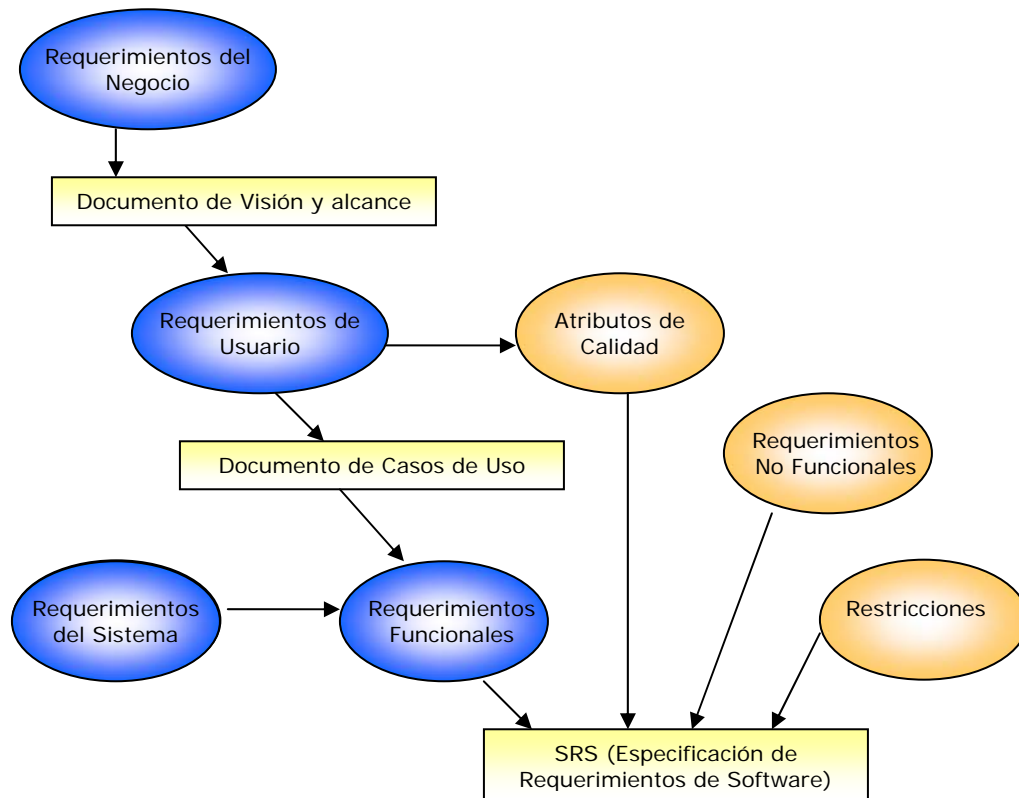
lo que el sistema debe proveer y las restricciones sobre las cuales debe operar¹⁹.

3.1.3.3 Descripción a nivel de sistema. Los requerimientos del sistema hacen referencia a la funcionalidad que debe ser construida para permitir al producto realizar sus tareas, en términos de las necesidades del sistema.

Los requerimientos del sistema se enfocan en las funciones del sistema, los servicios y las restricciones de operabilidad en detalle. El documento que contenga los requerimientos del sistema debe ser sumamente preciso y definir de manera exacta lo que va a ser implementado. Debe ser parte del contrato entre el comprador o cliente del sistema y desarrollador del mismo (para entender mejor la relación entre la descripción de requerimientos y los artefactos que se relacionan con los mismos Véase Figura 1...).

¹⁹**SOMERVILLE**, Op. cit.

Figura 1. Relación entre los diferentes artefactos, tipos de requerimientos y atributos de los mismos en un proyecto de software, desarrollado con Análisis y diseño orientado a objetos. En este caso se utilizan tecnologías particulares como los casos de uso.



Extraída de: **WIEGERS**, Op. cit. p. 23.

3.1.4 Características de un Buen Requerimiento.

Las características de un requerimiento son sus propiedades principales. Un conjunto de requerimientos en estado de madurez, deben presentar una serie de características tanto individualmente como en grupo.

3.1.4.1 Características de la Descripción de un Requerimiento. Las características que tiene una buena descripción individual de un requerimiento, que lo diferencian de uno mal descrito, son:

- Completo: Cada requerimiento debe describir de manera completa la funcionalidad que debe cumplir. Debe contener toda la información necesaria para que el desarrollador diseñe e implemente tal funcionalidad.
- Correcto: Cada requerimiento debe describir de manera precisa la funcionalidad que se debe construir. Un requerimiento correcto no debe entrar en conflicto con otro requerimiento. Sólo los usuarios más representativos del sistema pueden determinar de manera precisa si un requerimiento es correcto o no.
- Realizable: Debe ser posible implementar cada requerimiento de acuerdo a las capacidades y limitaciones del sistema y el medio que lo rodea. Para garantizar que no se determinen requerimientos no realizables, se recomienda contar con personal al interior del equipo de analistas de requerimientos que pueda establecer las limitaciones técnicas y de costos.
- Necesario: Cada requerimiento debe documentar algo que los clientes realmente necesiten, algo que sea para conformidad de un sistema externo con el que se tenga interacción, o para satisfacer un estándar. Para determinar si un requerimiento es

necesario se debe determinar quien lo propuso, es decir, conocer su origen.

- Priorizable: Es importante asignar una prioridad para cada requerimiento que indique que tan esencial es el mismo para la realización del producto. Se pueden perder elementos de juicio para el desarrollo del sistema si se asigna el mismo grado de prioridad a todos los requerimientos.
- No Ambiguo: Todos los lectores de un requerimiento deben llegar a una misma y consistente interpretación del mismo. El lenguaje usado en su definición, no debe causar confusiones al lector.
- Verificable: Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas²⁰.

3.1.4.2 Características de la Especificación de Requerimientos. Así mismo, las características que debe poseer en conjunto una buena especificación de requerimientos son:

- Completa: Una especificación de requerimientos está completa si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.

²⁰ **WIEGERS**, Op. cit.

- Consistente: Una especificación de requerimientos es consistente si no existen requerimientos que se contradigan.
- Modificable: Una especificación de requerimientos debe permitir ser revisada y mantener un historial de cambios hechos sobre cada requerimiento. Esto requiere que cada requerimiento sea etiquetado de manera única y expresado de manera separada de otros requerimientos para permitir referirse a él de manera no ambigua.
- Trazable: Cada requerimiento debe poder permitir trazar una línea del tiempo en la cual indique sus orígenes, y permita ser extendido a otras etapas del desarrollo del producto²¹.

3.2 GENERALIDADES DE LA INGENIERÍA DE REQUERIMIENTOS

3.2.1 ¿Qué es la Ingeniería de Requerimientos?

El marco conceptual sobre el cual se desarrolla este proyecto de grado está constituido principalmente por el proceso de Ingeniería de Requerimientos. Este se rige bajo un enfoque tradicionalista de la Ingeniería de Software; y tiene como objetivos el identificar, analizar, documentar, validar y administrar los requerimientos que van a ser desarrollados para un sistema o producto de software. Algunas de las definiciones más generales del mismo son:

²¹WIEGERS, Op. cit.

"Ingeniería de Requerimientos es la disciplina para desarrollar una especificación completa, consistente y no ambigua, la cual servirá como base para acuerdos comunes entre todas las partes involucradas y en dónde se describen las funciones que realizará el sistema"²².

"Ingeniería de requerimientos es un enfoque sistémico para recolectar, organizar y documentar los requerimientos del sistema; es también el proceso que establece y mantiene acuerdos sobre los cambios de requerimientos, entre los clientes y el equipo del proyecto"²³

"La Ingeniería de Requerimientos es la ciencia y disciplina a la cual le concierne el establecer y documentar los requerimientos."²⁴

Como se puede apreciar en cada una de estas definiciones, todos los procesos involucrados con la Ingeniería de Requerimientos están relacionados con identificar, modelar, comunicar y documentar los requerimientos de un sistema o producto de software y los contextos en los cuales este sistema o producto está envuelto.

Los requerimientos deben describir lo que se debe hacer y cómo se debe llevar acabo. Esto en la vida real es algo muy difícil de realizar. Por esto

²²**BOEHM**, Barry. Software Engineering Economics. New Jersey: Prentice Hall, 1981.

²³**OBERG**, Roger; **PROBASCO**, Leslee y **ERICSSON**, Maria. **RATIONAL SOFTWARE**. Applying requirements management with use cases [online]. Rational Software Corporation, 2003 [consultada mayo de 2005]. Disponible en Internet: <<http://www.uml.org.cn/RequirementProject/pdf/apprmuc.pdf>>

²⁴**THAYER**, Richard y **DORFAM**, Merlin. Software Requirements Engineering. 2 ed. Los Alamitos, California: IEEE Computer Science Press, 2000. p. 1.

existen muchas técnicas disponibles para la aplicación de la Ingeniería de Requerimientos, con el fin de asegurar que los requerimientos obtenidos cuenten, al final del proceso de Ingeniería de Requerimientos, con las características necesarias para ser implementados. Por tanto, lo que se busca al aplicar un proceso de Ingeniería de Requerimientos es el ayudar a la totalidad de los participantes del proyecto a conocer que desean construir antes de empezarlo a construir.

Esta práctica trae beneficios en dos aspectos:

- Minimiza los riesgos de fracaso del proyecto.
- Contribuye a cumplir aspectos de calidad, tiempo y presupuesto.

Estas afirmaciones se basan en las siguientes premisas de la Ingeniería de Software:

- El costo de encontrar un error en el desarrollo de un proyecto de software se incrementa a medida que el proyecto avanza fases, y en el tiempo^{25 26}.
- Es posible establecer un mínimo conjunto estable de requerimientos antes de empezar fases de diseño e implementación dentro de un proyecto de software²⁷.

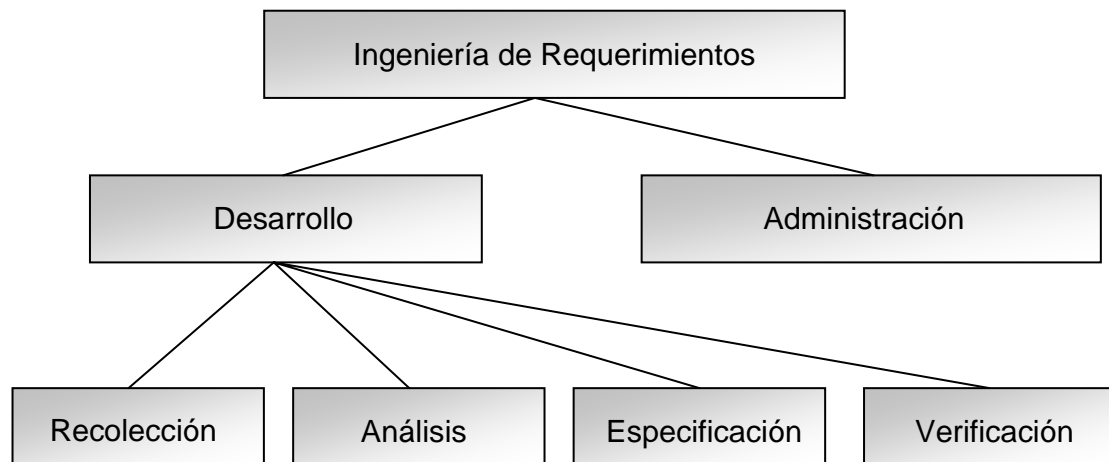
²⁵**BOEHM**, Op. cit.

²⁶**PAETSCH**, Fachhochschule; **EBERLEIN**, Armin and **MAURER**, Frank. Requirements engineering and agile software development [online]. Disponible en Internet: <<http://citeseer.ist.psu.edu>>

²⁷**BOEHM**, Op. cit.

Por esto, el proceso de Ingeniería de Requerimientos describe de manera detallada y precisa cada uno de los aspectos del ciclo de vida de un conjunto de requerimientos. Este proceso presenta dos grandes ramas: El *Desarrollo* de requerimientos, y la *Administración* de requerimientos (...Véanse Figuras 2 y 3...).

Figura 2. Estructura de la Ingeniería de Requerimientos.



WIEGERS, Karl. When telepathy won't do: Requirements engineering key practices.

En: Cutter IT Journal. Vol. 13. No. 5 (may 2000). Disponible en Internet:

<<http://www.processimpact.com/articles/telepathy.pdf> >

Cada una de estas actividades que conforman el Desarrollo de Requerimientos consisten en:

- **Recolección (Elicitation):** Es el Proceso a través del cual los clientes (compradores y/o usuarios) y el desarrollador (contratista) de un sistema de software; descubren, revisan,

articulan, y entienden las necesidades de los usuarios del sistema y las restricciones que se dan sobre el software y el desarrollo del mismo.

- Análisis (**Analysis**): Es el proceso de analizar las necesidades de los clientes y los usuarios para llegar a una definición de los requerimientos de software.
- Especificación (**Specification**): Consiste en el desarrollo de un documento que de manera clara y precisa contenga y especifique cada uno de los requerimientos del sistema de software.
- Verificación (**Verification**): Es el proceso de asegurar que la especificación de requerimientos de software sea acorde con los requerimientos del sistema, conforme a los estándares de documentación de la fase de requerimientos, y que a su vez este documento sea una base sólida para la arquitectura y el diseño²⁸.

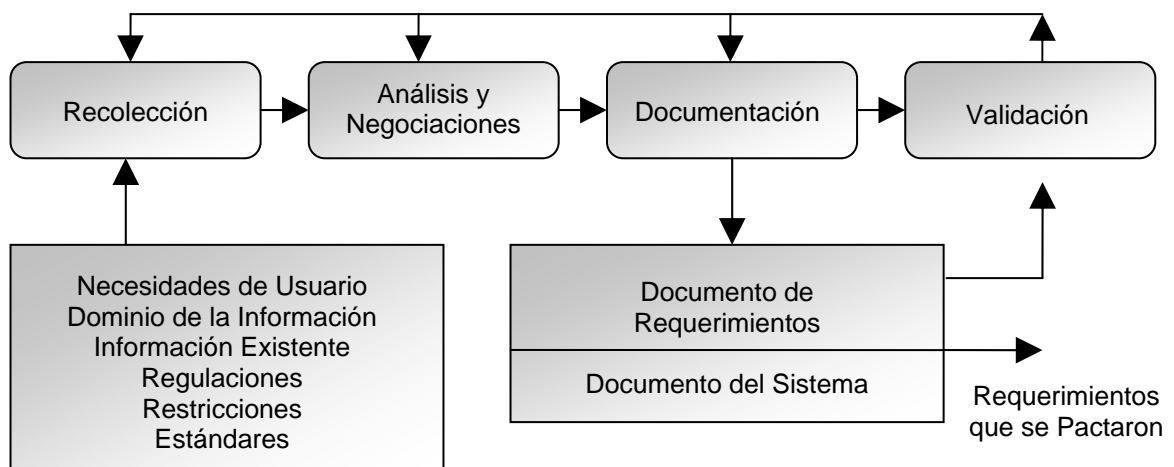
Cada una de estas actividades está enfocada a permitir el análisis y documentación de los requerimientos de un sistema (...Véase Figura 4...). Esto envuelve el transformar las necesidades operativas en una descripción del sistema, parámetros de funcionamiento del mismo y su configuración. Este proceso solo puede ser alcanzado a través de un proceso iterativo de análisis, diseño, estudios de alcance y construcción de prototipos²⁹.

²⁸THAYER, Op. cit. p. 29.

²⁹Ibid.

Por otra parte, la necesidad de recrear un proceso iterativo sobre el desarrollo de requerimientos nos conduce a la necesidad de ejercer control y establecer una línea base para la administración de los requerimientos; esto con el fin de mantener la consistencia de lo que se especifica respecto a lo que se desarrolla. Estas son las tareas de la Administración de requerimientos³⁰.

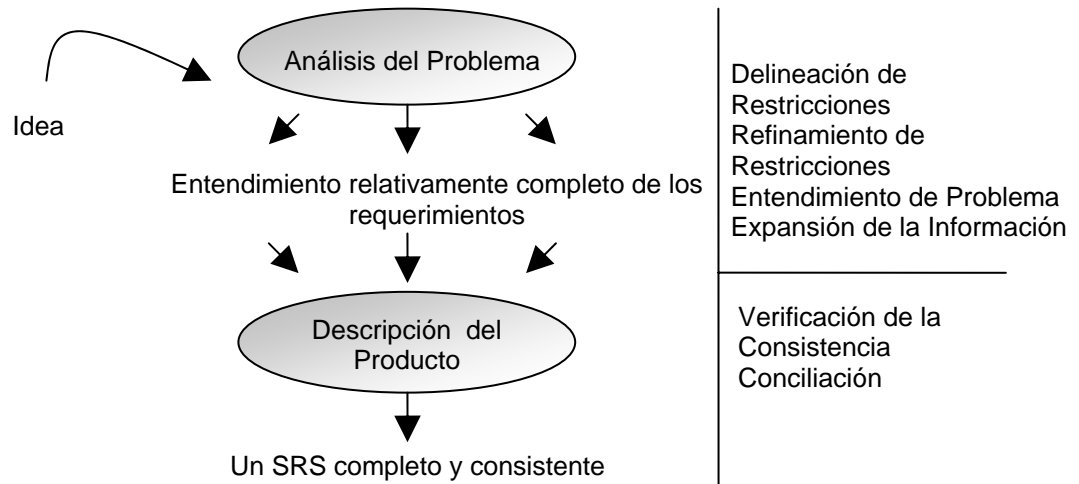
Figura 3. Proceso de la Ingeniería de Requerimientos.



MEAD, Nancy R. Requirements management and requirements engineering: You can't have one without the other. En: Cutter IT Journal. Vol. 13. No. 5 (may 2000).

³⁰**MEAD**, Nancy R. Requirements management and requirements engineering: You can't have one without the other. En: Cutter IT Journal. Vol. 13. No. 5 (may 2000).

Figura 4. Actividades del Ciclo de Vida de Los Requerimientos.



MEAD, Nancy R. Op. cit.

3.2.1.1 Recolección. La recolección es la fase inicial en la cual se trata de descubrir los requerimientos e identificar los límites del sistema a través de la consulta a los participantes del sistema (**stakeholders**).

Dentro de los objetivos de esta fase se encuentran el entender el dominio de la aplicación, las necesidades del negocio, las restricciones del sistema, a los participantes del sistema y al problema en sí, para entender de manera inicial lo que se debe desarrollar.

Algunas de las técnicas y herramientas más importantes para llevar a cabo la recolección de requerimientos son:

Entrevistas:

La entrevista es un método para descubrir hechos y opiniones que tienen los posibles usuarios y otros participantes dentro del sistema que

se está desarrollando. Los errores y malentendidos pueden ser detectados y corregidos a través de este método, por lo cual resulta muy útil dentro de esta actividad de la ingeniería de requerimientos. Las entrevistas pueden ser clasificadas en dos grandes grupos.

- Las entrevistas **cerradas**, donde el entrevistador (ingeniero de requerimientos) prepara un conjunto de preguntas antes del encuentro con el entrevistado, y se buscan respuestas para las preguntas formuladas.
- Las entrevistas **abiertas**, en las cuales no se preparan preguntas concretas, y, por el contrario, se discute con el entrevistado las expectativas que este tiene del sistema³¹.

No existe en realidad una delimitación entre los dos tipos de entrevistas en el momento de llevarlas a cabo. Pueden ser utilizadas de manera conjunta y no necesariamente exclusiva ni excluyente. La ventaja de este método es que permiten que el entrevistador obtenga una colección rica en información, que le puede ser útil al desarrollador. La desventaja que tiene este método, es que la información que se recolecta puede ser difícil de organizar y analizar, y diferentes participantes dentro del desarrollo del sistema pueden proveer información conflictiva y contradictoria, esto consecuencia de la gran cantidad de información que es obtenida durante la ejecución de este método.

³¹ **GOGUEN**, Joseph A. y **LINDE**, Charlotte. Techniques for requirements elicitation. En: **THAYER**, Richard y **DORFAM**, Merlin. Software Requirements Engineering. 2 ed. Los Alamitos, California: IEEE Computer Science Press, 2000. p. 140-152.

Casos de Uso y/o Escenarios.

Los casos de uso describen interacciones entre los usuarios y el sistema, enfatizando en lo que el usuario necesita del sistema. Un caso de uso describe la posible secuencia de interacciones que se dan entre el sistema y uno o más actores como respuesta a un estímulo inicial por parte de alguno de los actores³². De igual manera, debe ser incluida dentro de esta interacción, la descripción de las variantes y extensiones que el sistema debe soportar.

Los casos de uso representan los requerimientos funcionales del software y pueden ser utilizados dentro de las primeras etapas del proceso de desarrollo. Así mismo, los casos de uso están escritos en lenguaje natural y son descripciones expresadas de manera informal. Las descripciones expresan lo que sucede desde el punto de vista del usuario. Los detalles de cómo el sistema debe funcionar internamente son irrelevantes al caso de uso³³.

Por otra parte, los escenarios son ejemplos de sesiones de interacción entre el sistema y el usuario, donde un solo tipo de interacción entre los dos participantes es simulada y descrita. Los escenarios deben incluir una descripción del estado del sistema antes y después de la

³²**RUMBAUGH**, James. Getting started: Using use cases to capture requirements. En: **THAYER**, Richard y **DORFAM**, Merlin. Software Requirements Engineering. 2 ed. Los Alamitos, California: IEEE Computer Science Press, 2000. p. 153 - 157.

³³**RUMBAUGH**, Op. cit.

culminación del escenario, que actividades deben ser simultaneas, el flujo normal de los eventos y las excepciones a esos eventos³⁴.

Observación y análisis social.

Los métodos de observación involucran a dos participantes: el investigador observando al usuario mientras trabaja y tomando notas de las actividades que se llevan a cabo, y al trabajador (usuario) llevando a cabo las actividades cotidianas que su trabajo le implica realizar. La observación puede ser realizada de manera directa, es decir que el investigador este presente mientras el usuario realiza sus actividades; o indirecta, cuando la observación se lleva en otro escenario, instante, o a través de otro medio que permita que el observador no este presente durante la realización de las actividades que esta observando (como lo permitiría el uso de una cámara de video). Este método es muy útil cuando se busca estudiar las actividades y procesos que se están llevando a cabo en una organización en el momento. La observación permite a los investigadores observar lo que los usuarios hacen actualmente en un determinado contexto. Esto permite superar problemas con los participantes del proyecto que realizan descripciones idealizadas o demasiado simplificadas de los procesos que se llevan a cabo en sus trabajos³⁵.

Lluvia de Ideas.

³⁴**KONTOYA**, Gerald y **SOMMERVILLE**, Ian. Requirements engineering: Processes and techniques. Chinchester, Inglaterra: John Wiley & Sons Ltd, 1998.

³⁵**PAETSCH**, Op. cit.

Las lluvias de ideas son sesiones donde todos los participantes brindan sus ideas para obtener una solución a una problemática. Una lluvia de ideas está compuesta de dos fases: la fase de generación y la fase de evaluación. Durante la generación las ideas son recolectadas y es importante que no sean criticadas. Durante la evaluación de las ideas, las propuestas de solución deben ser evaluadas desde diferentes perspectivas.

Algunas de las características que tienen estas sesiones, es que las ideas deben ser generadas de manera rápida y abierta. De igual manera, es importante que el ambiente de la sesión fomente la creatividad de los participantes y esté enfocado a una problemática específica. Todas estas consideraciones permiten que este método conlleve a un mejor entendimiento del problema, y permita que los participantes de la sesión adquieran un sentido de propiedad sobre la solución que se debe llevar a cabo³⁶.

Prototipos.

En la ingeniería de software, un prototipo es programa de computador que implementa algunos de los requerimientos de un sistema. Este prototipo puede ser usado para colaborar con la definición de los requerimientos, o para facilitar la evaluación de alternativas de implementación de un sistema³⁷.

³⁶PAETSCH, Op. cit.

³⁷INSTITUTE FOR ELECTRONICS AND ELECTRICAL ENGINEERS. Op. cit.

Existen dos grandes tipos de prototipos. Los prototipos no funcionales o desechables (**Throw away**), que sirven para entender la dificultad y aclarar los requerimientos; y los prototipos funcionales o evolutivos (**Evolutionary**) que permiten construir una aproximación del sistema de manera que se pueda proveer cierta funcionalidad del sistema final y usualmente se convierten en parte del mismo³⁸.

En general, los prototipos se consideran herramientas muy valiosas para clarificar los requerimientos que son confusos durante el desarrollo de un sistema. Los prototipos actúan de manera similar a los escenarios, debido a que proveen un contexto en el cual los usuarios pueden entender mejor la información que ellos deben proveer a los desarrolladores para que se pueda construir el sistema³⁹.

3.2.1.2 Análisis. El Análisis de Requerimientos de Software es el proceso en el cual se estudia las necesidades del usuario para obtener una definición de los requerimientos de software. Una Especificación de Requerimientos de Software es un documento en el cual se describe de manera clara y precisa cada uno de los requerimientos esenciales (funcionalidad, rendimiento, restricciones de diseño y atributos de calidad) de un software y sus interfaces externas⁴⁰.

³⁸ **PAETSCH**, Op. cit.

³⁹ **SAWYER**, Peter y **KONTOYA**, Gerald. Software requirements. En: Software Engineering Book Of Knowledge (SWEBOK), capítulo 2. p.18. Disponible en Internet: <<http://www.swebok.org>>

⁴⁰ **THAYER**, Op. cit. p. 137.

Dentro de las actividades que se llevan a cabo en el análisis de requerimientos se encuentran: la descomposición de los requerimientos de alto nivel en requerimientos funcionales detallados, construcción de modelos gráficos de requerimientos, construcción de prototipos, priorizar los requerimientos, establecer atributos asociados con los requerimientos como puede ser su costo, o el beneficio que puede representar para el negocio, detectar y resolver conflictos entre requerimientos, descubrir el alcance del sistema y cómo interactúa en su ambiente, etc.^{41 42}.

Dentro de las prácticas principales para el análisis de requerimientos se encuentran:

JAD (Joint Application Development)

Esta práctica se basa en la creación de espacios que permitan celebrar sesiones o reuniones en donde los participantes y directos interesados dentro del desarrollo del proyecto buscan obtener o generar conocimiento alrededor del desarrollo que se va a llevar a cabo. En estas sesiones se trabaja bajo un enfoque común que permite el fácil entendimiento de los temas expuestos por parte de los invitados a la sesión (usualmente un enfoque de análisis estructurado), y se persiguen como propósito diferentes aspectos: definir niveles de detalle del proyecto, diseñar una solución, monitorear el proyecto, etc.^{43 44}.

⁴¹**WIEGERS**. When telepathy won't do: Requirements engineering key practices. Op. cit. p. 14.

⁴²**SAWYER**. Op. cit. p. 19.

⁴³**GOGUEN**, Op. cit. p. 143.

Las personas que pueden ser partícipes de estas reuniones pueden variar desde ejecutivos, líderes de proyectos, usuarios, expertos en el sistema y personal técnico externo⁴⁵. Este tipo de herramientas buscan obtener y maximizar el potencial de la cooperación y el trabajo en equipo entre los diferentes tipos de participantes. Dentro de las consideraciones que se deben tener en cuenta cuando se utiliza esta herramienta de trabajo es el nombrar un líder de las sesiones que impida que el rumbo de las sesiones se salga del propósito establecido de la sesión. Así mismo, la información que se obtiene en estas sesiones proviene de diferentes fuentes y representa los intereses de diferentes aspectos del sistema que se desea desarrollar, por lo tanto, es importante consignar esta información como base para la retroalimentación del proceso de ingeniería de requerimientos⁴⁶.

Priorización de Requerimientos

Para mejorar la toma de decisiones en el momento de diseño y desarrollo, así como en otros aspectos de desarrollo del sistema es importante establecer un procedimiento que facilite esta importante actividad. Un potencial remedio para este dilema es la priorización de requerimientos, que permite manejar la situación descrita

⁴⁴PAETSCH, Op.cit

⁴⁵Ibid.

⁴⁶Ibid.

anteriormente. Esto permite controlar las decisiones que se realicen teniendo en cuenta a la fuente generadora de las necesidades⁴⁷.

Modelos

La definición más general que se puede encontrar para la palabra modelo es:

- Representación en pequeño de alguna cosa.
- Esquema teórico, generalmente en forma matemática, de un sistema o de una realidad compleja, como la evolución económica de un país, que se elabora para facilitar su comprensión y el estudio de su comportamiento.⁴⁸

En ingeniería de software, existen dos tipos de modelos básicos: el modelo conceptual y el modelo de comportamiento.

- *Modelo conceptual:* Es el utilizado en la especificación del sistema, representa los conceptos más significativos en el dominio del problema.... Nos describe la parte estática del problema, es una fotografía del mundo real.
- *Modelo de Comportamiento:* Utilizado en la parte de diseño del sistema, define la parte dinámica, es decir, cual debe ser el

⁴⁷**WIEGERS**, Karl. First things first: Prioritizing requirements. En: Software Development Magazine [online], September 1999. Disponible en Internet: <www.processimpact.com/articles/prioritizing.pdf>

⁴⁸**Real Academia de la Lengua Española (RAE)**. Definición de la Palabra "modelo". Disponible en Internet: <<http://www.rae.es>>

comportamiento en cada situación y la forma de proceder. Los diagramas de secuencia y de estados son parte de este modelo⁴⁹.

Los modelos en ingeniería de software son aquellos que nos permiten representar características de un sistema, aportando información al proceso de análisis y diseño. Dentro de los modelos más populares para representar sistemas se encuentran: diagramas de flujo, modelos de datos como el diagrama entidad-relación y aproximaciones orientadas a objetos como los diagramas de clases⁵⁰.

3.2.1.3 Documentación. El propósito de la documentación de requerimientos es el comunicar los requerimientos a los diferentes participantes del desarrollo de software. El documento de requerimientos de software es una herramienta que puede servir como base para la evaluación de los requerimientos en el producto, y para la evaluación misma del proceso que se llevó a cabo (evaluar las actividades de diseño, implementación, las pruebas realizadas y la verificación y validación de estas actividades). De igual manera este documento es producto que debe ser controlado y administrado, ya que contiene los requerimientos que finalmente se van a llevar a cabo.

Las características de un buen documento que especifica los requerimientos fueron analizadas anteriormente (...Ver numeral 3.4.1...);

⁴⁹**MICROSOFT DEVELOPERS NETWORK (MSDN)**. Sitio Web para estudiantes Latinoamérica. Disponible en Internet:
<<http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/ingenieria/especif.asp>>

⁵⁰**KONTOYA**, Op. cit.

pero son relevantes porque este tipo de documento podría hacer parte de un contrato⁵¹.

3.2.1.4 Verificación. Es la etapa final de desarrollo de requerimientos. Su objetivo es el verificar que todos los requerimientos que aparecen en el documento especificado describan y representen de manera clara, lo que se desea implementar para el sistema. Esto implica verificar que los requerimientos sean consistentes y que estén completos.

Esta actividad representa un punto de control interno y externo; interno, porque se debe verificar internamente lo que se está haciendo, y externo, porque se debe validar con el cliente⁵².

3.2.2 Herramientas, técnicas y software

Como se analizó en la sección anterior, existen diferentes herramientas que se utilizan para llevar a cabo las fases de la ingeniería de requerimientos. Sin embargo, estas herramientas no son restrictivas para una única actividad dentro de este gran proceso (...Véase Tabla 1...).

Tabla 1. Herramientas que se utilizan para las diferentes fases de la ingeniería de requerimientos.

⁵¹IEEE. Op. cit.

⁵²DÁVILA, Nicolás Davyt. Ingeniería de requerimientos: Una guía para extraer, analizar, especificar y validar los requerimientos de un proyecto. Artículo Técnico (Licenciado en Análisis de Sistemas). Universidad ORT, Uruguay. Facultad de Ingeniería. Área de Sistemas.

Herramientas	Extracción	Análisis	Especificación	Validación
Entrevistas y cuestionarios	X			
Sistemas existentes	X	X		
Grabaciones de video y de audio	X	X		
Brainstorming (lluvia de ideas)	X	X		
Arqueología de Documentos	X	X		
Observación	X			
Prototipo Throw Away (no funcional)	X	X	X	
Prototipo Evolutionary (funcional)	X		X	X
Análisis DOFA		X		
Cadena de Valor		X		
Modelo Conceptual		X	X	
Diagrama de Pescado	X	X	X	
Glosario	X	X	X	X
Diagrama de Actividad		X	X	
Casos de uso	X	X	X	X
Casa de Calidad o QFD				X
Checklist	X		X	

DÁVILA, Nicolás Davyt. Ingeniería de requerimientos: Una guía para extraer, analizar, especificar y validar los requerimientos de un proyecto. Artículo Técnico (Licenciado en Análisis de Sistemas). Universidad ORT, Uruguay. Facultad de Ingeniería. Área de Sistemas.

También existen herramientas de software que facilitan la realización de cada una de esas actividades (...Véase Tabla 2...).

Tabla 2. Características de las herramientas de Administración de requerimientos más utilizadas⁵³.

CARACTERÍSTICAS	DOORS	RTM Workshop	Caliber - RM	RequisitePro
Permite la conversión de documentos fuente para cargar los requerimientos a una Base de Datos (BD)	✓	✓	✓	✓
Importa los requerimientos desde tablas de word hacia BD	✓	✓	✓	
Incorpora objetos no textuales como hojas de trabajo de excel e imágenes dentro de BD	✓	✓	✓	
Sincroniza un SRS con el contenido de BD		✓		✓
Define diferentes atributos para diferentes tipos de requerimientos y ajusta atributos para requerimientos individuales	✓	✓	✓	✓
Define directivas para requerimientos	✓	✓		
Notifica a los participantes afectados del proyecto vía e-mail acerca de los cambios en los requerimientos	✓	✓	✓	
Define relaciones de trazabilidad y vínculos entre los requerimientos individuales y entre estos y otros elementos del sistema	✓	✓	✓	✓
Permite definir opciones de usabilidad para requerimientos no funcionales	✓	✓	✓	✓
Incluye elementos para el aprendizaje como tutoriales y proyectos de ejemplo	✓	✓	✓	✓
Se integra con otras herramientas como: prueba, diseño y administración de proyectos	✓	✓	✓	✓
Define usuarios y grupos, con sus respectivos permisos de acceso	✓	✓	✓	✓
Habilita discusiones de requerimientos		✓	✓	✓
Incluye interfaces web para consultas a la BD, discusiones y tal vez la actualización de atributos de requerimientos	✓	✓	✓	✓
Incluye un sistema para desarrollos de sistemas basados en propuestas de cambios	✓	✓		
Incluye una completa documentación de manuales de usuario	✓		✓	✓

⁵³Lista de herramientas seleccionada por el autor como las más utilizadas por las empresas de E.E.U.U.

WIEGERS, Karl. Automating Requirements Management [online]. Disponible en Internet: <http://www.processimpact.com/articles/rm_tools.pdf>.

Estas herramientas están orientadas a soportar algunas de las actividades de la ingeniería de requerimientos, pero principalmente se orientan a permitir la administración de los requerimientos en relación a estas actividades. Por esta razón se les considera herramientas administrativas, y no herramientas que pertenezcan a alguna fase específica de la ingeniería de requerimientos.

3.2.3 Consideraciones para el proceso de ingeniería de requerimientos

A pesar que la ingeniería de software y la tecnología han tenido avances muy grandes en los últimos años, existen procesos de software que son informales, parciales y en algunos casos no confiables. Puede sonar ilógico si se analiza que el número creciente de herramientas automatizadas que han surgido para ayudar a aplicar un proceso de desarrollo de software efectivo. Entonces, ¿cuáles son las razones por las cuales los existen tantos proyectos de software víctimas de retrasos, presupuestos sobregirados y con problemas de calidad?

Estudios realizados por el Standish Group's CHAOS⁵⁴, muestran que la mayoría de las causas por las cuales los proyectos de software fracasan se relacionan directamente con los requerimientos. La ingeniería de requerimientos es una disciplina muy inmadura y constituye un campo

⁵⁴**OBBERG**, Op. cit. p. 1.

de batalla ocupado por un sinnúmero de métodos comerciales enfocados a cada una de sus actividades⁵⁵.

Las actividades de la ingeniería de requerimientos conllevan una serie de problemas, dentro de los cuales están:

- Los requerimientos no son obvios y vienen de muchas fuentes.
- Son difíciles de expresar en palabras (el lenguaje es ambiguo).
- Existen muchos tipos de requerimientos y diferentes niveles de detalle.
- La cantidad de requerimientos en un proyecto puede ser difícil de manejar.
- Los requerimientos están relacionados unos con otros, y a su vez se relacionan con entregables y otras partes del proceso.
- Cada requerimiento tiene propiedades únicas y abarcan áreas funcionales específicas.
- Un requerimiento puede cambiar a lo largo del ciclo de desarrollo.
- Existen diferentes interesados y responsables de los mismos....
- Son sensibles al tiempo⁵⁶.

Lo anterior sugiere que existen muchos campos de investigación y acción en la ingeniería de requerimientos, que abarcan diferentes aspectos y variables que influyen directa e indirectamente en el éxito o fracaso de los proyectos de software.

⁵⁵**GOGUEN**, Op. cit. p. 141.

⁵⁶**OBERG**, Op. cit. p. 3.

4. LA PEQUEÑA EMPRESA DESARROLLADORA DE SOFTWARE

4.1 LA SITUACIÓN ACTUAL

Las pequeñas empresas forman parte del porcentaje más alto de empresas que mueven el motor de la economía nacional. Esto se debe, a que junto con la mediana empresa, constituyen el 90% del parque nacional empresarial⁵⁷. La población de las mismas es (5) veces mayor que la población de medianas empresas. Por estas razones, constituyen una oportunidad de investigación muy importante cuando se trata de establecer elementos que colaboren al mejoramiento de la realidad económica nacional.

Estudios realizados alrededor de las PyMEs han mostrado el verdadero panorama que enfrentan las pequeñas empresas. Entre los diferentes aspectos que rodean al mismo, se puede destacar que: El 50% de las PyMEs ha tenido reducciones en inversión, el 43% no han solicitado créditos en los últimos dos años, enfrentan cerca de sesenta modalidades de tributación diferente, el 87% de PyMEs *no exportan* y el 50% afirma que en los últimos dos años sus utilidades se han reducido⁵⁸.

⁵⁷**RODRÍGUEZ**, Astrid Genoveva. La realidad de la PyME Colombiana: Desafío para Colombia. Bogotá: Fundes, 2003. p. 1.

⁵⁸Ibid.

Estos datos demuestran que las PyMEs, y en nuestro caso de análisis, las pequeñas empresas; poseen una gran cantidad de necesidades, y las mismas, pueden satisfacerse mediante el estudio de los diferentes factores que las afectan.

Uno de estos factores, es la distribución geográfica de las empresas. Bogotá posee la mayor concentración de pequeñas empresas en el país, con el 50% del total de las PyMEs de Colombia, lo cual la convierte en un importante foco de investigación alrededor de la pequeña empresa⁵⁹.

Otro factor para analizar, es la baja tasa de exportaciones. A pesar de los enormes esfuerzos realizados por el Estado para promoverlas, apenas el 13% de las PyMEs colombianas han exportado en los últimos dos años, aunque no en todos los casos de manera continua. Del 87% de PyMEs que no exportan, únicamente el 7% lo ha intentado. Estas proporciones son mayores para las empresas medianas que para las pequeñas. Ahora bien, el mercado exterior es muy relevante para estas empresas. La exportación de las mismas se cataloga como no tradicional, por lo cual representa una oportunidad muy grande para el crecimiento económico del país⁶⁰.

Por tales motivos, es necesario establecer elementos que permitan que las pequeñas empresas sean más competitivas, y mejoren sus condiciones de manera que puedan afrontar los retos del futuro, y permitan el crecimiento económico del país.

⁵⁹ **RODRÍGUEZ**, Op. cit.

⁶⁰ Ibid.

4.1.1 Economía

Del numeral anterior, se puede concluir que las pequeñas empresas enfrentan un proceso, a través del cual, deben buscar elementos que les permitan explotar su potencial exportador.

La situación actual de las empresas desarrolladoras de software de Colombia, tiene características muy particulares. En primer lugar, La producción de software local, presenta un crecimiento promisorio. Esto significa que las empresas solicitan un número mayor de desarrollos de software, y las empresas nacionales, pueden cubrir esa demanda⁶¹. En segundo lugar, se debe recalcar que existe una alta volatilidad de este tipo de empresas en el mercado. Es decir que estas empresas nacen y desaparecen a una razón bastante alta. En tercer lugar, se puede identificar que en el mercado se encuentra mano de obra muy calificada, y una gran oferta de ingenieros, sin embargo, hay una muy baja participación de certificaciones internacionales por parte de los mismos. En cuarto lugar, y no menos importante, cabe destacar que las empresas están emprendiendo un proceso para la aplicación de programas de calidad. Esto contribuye a que las empresas sean más competitivas y se superen barreras en nuevos mercados, como los internacionales. En este aspecto, sólo 42 empresas cuentan en la actualidad con la certificación en ISO 9000, incluyendo a grandes, medianas y pequeñas empresas. Estos resultados son muy bajos

⁶¹**VALDES CÁRDENAS**, Luis Eduardo. Situación actual de la informática en Colombia. Bogotá: Centro de Apoyo a las Tecnologías Informáticas (CATI), abril de 2004.

comparados con otros sectores industriales, los cuales han sido forzados a aplicar modelos de Calidad que eleven el nivel del sector. Las Empresas Colombianas están en desventaja frente a niveles de competitividad basados en la aplicación de programas de Calidad. Estados Unidos cuenta con aproximadamente 1700 empresas que aplican programas de CMM y CMMI. En Colombia solo hay una⁶².

Analizando estas particularidades de las empresas desarrolladoras de software, surge la interrogante ¿qué elementos pueden ayudar a las empresas desarrolladoras de software en la situación actual en que vivimos?

4.1.2 La Ingeniería de Requerimientos

En el campo de las pequeñas empresas existen muchos interrogantes. No se conocen estadísticas concretas de cómo se llevan los procesos relacionados con el desarrollo de software, en empresas desarrolladoras de software y, más concretamente, no se conoce el estado de ingeniería de requerimientos al interior de las mismas.

Analizar estos procesos, su calidad, y cómo mejorar dichos procesos, en la pequeña empresa desarrolladora de software, es un campo de investigación muy amplio. Esta investigación se centra en el proceso que tiene mayor incidencia y relevancia, la ingeniería de requerimientos. Esto formula un interrogante ¿cómo se llevan a cabo las actividades de la ingeniería de requerimientos al interior de este tipo de empresas?

⁶²VALDES CÁRDENAS, Op. cit.

4.1.2.1 El estado de la ingeniería de requerimientos en las pequeñas empresas. Existen muchos métodos útiles que se podrían implementar para determinar el estado de la ingeniería de requerimientos en las pequeñas empresas. En esta investigación se utilizarán dos: análisis de escenarios y pruebas de campo.

Existen diferentes razones por las cuales se utilizaron estos métodos. En primer lugar, uno de los métodos más confiables para determinar cuál es el comportamiento de una población, es hacer un extenso análisis estadístico de casos representativos. Sin embargo, por motivos de recursos de tiempo, personal y dinero; llevar a cabo una investigación a través de este método sería un objetivo irrealizable.

Por tanto, es necesario plantearse hipótesis de cómo podría ser el estado del análisis de requerimientos, y en general, la ingeniería de los mismos en las empresas sujeto de la investigación. Para esto, es necesario formular las posibles condiciones en que se pueden encontrar las mismas y establecer cuáles son las posibles características que se poseen los procesos de ingeniería de requerimientos en estas empresas.

Por esta razón se formularon escenarios representativos, a través de los cuales analizar el estado de este tipo de empresas. Esto se realizó a través de la comparación directa entre la situación de la ingeniería de requerimientos de las pequeñas empresas, y los niveles definidos en CMM. Los niveles de CMM son representativos para cualquier empresa involucrada en el desarrollo de software, de manera que el análisis de los distintos escenarios que se pueden presentar en la pequeña empresa

es transparente y tiene relación directa con los diferentes niveles de clasificación con estos diferentes niveles (...Ver Tabla 3...):

Tabla 3. Relación de CMM con el estado del proceso de requerimientos al interior de una empresa.

NIVEL	ESTADO DE MADUREZ (CMM)	REQUERIMIENTOS
Nivel 0	No existe proceso	Caos
Nivel 1	Informal	Definición informal de procesos
Nivel 2	Repetible	Control de requerimientos
		Planeación y seguimiento de Proyectos
		Administración de la configuración
		Aseguramiento de la calidad
Nivel 3	Definido	Administración de software íntegra
		Definición formal de los procesos
Nivel 4	Administrable	Control de la calidad
		Administración cuantitativa de procesos
Nivel 5	Optimizado	Control de cambios
		Prevención de defectos

SOFTWARE ENGINEERING INSTITUTE (SEI). Capability Maturity Model [online]. Disponible en Internet: <<http://www.sei.cmu.edu/cmm>>

Cada una de las características definidas para cada uno de los niveles de la tabla anterior, son los elementos principales con los que debe contar una empresa desarrolladora de software para ser clasificada dentro de alguno de los respectivos niveles. Por esta razón, se puede extrapolar esta misma situación a la empresas sujeto de manera que se puedan conocer algunas características del proceso de ingeniería de requerimientos al interior de los diferentes tipos de pequeñas empresas desarrolladoras de software bogotanas que se encuentran actualmente en el mercado.

4.1.2.2 Características de la pequeña empresa desarrolladora de software. Para entender mejor cuales son las necesidades de las empresas estudiadas, es necesario describir de manera detallada cuales son las características de este tipo de empresas.

Para poder ser clasificada como una pequeña empresa, se debe cumplir con las siguientes características. En primer lugar, el orden de activos de una pequeña empresa, corresponde a un valor superior a un millón de pesos (\$1'000.000.00) e inferior a mil cien millones de pesos (\$1100'000.000.00).⁶³

Por otra parte, en lo que se refiere a personal, las pequeñas empresas poseen un número inferior de 60 personas como trabajadores de planta⁶⁴.

Ahora, para que una empresa pueda ser clasificada como empresa desarrolladora de software, sus actividades económicas deben estar relacionadas con la producción de software como objeto social de la empresa, es decir que las actividades de la empresa se relacionan con la producción de software para clientes externos a la organización. Dentro de este concepto aplican los diferentes tipos de empresas como las de asesoría y consultoría externa (**outsourcing**), y las fábricas de software (**software factory**), entre otras.

⁶³**CAMARÁ DE COMERCIO DE BOGOTÁ.** Estadísticas generales sobre las empresas colombianas.

⁶⁴Ibid.

5. EL PROCESO DE ANÁLISIS DE REQUERIMIENTOS

5.1 MODELOS TRADICIONALES

Para entender mejor las técnicas que son utilizadas para llevar a cabo el análisis de requerimientos, hay que diferenciar 4 grandes perspectivas sobre las cuales se puede llevar a cabo este proceso.

5.1.1. Descomposición Funcional

Esta estrategia consiste en definir el comportamiento requerido (requerimientos) como una relación entre entradas y salidas de software. Se procede idealmente con una estructura **top-down** (arriba hacia abajo), identificando primero la funcionalidad del sistema como un todo. Después se procede a descomponer esta funcionalidad en un conjunto de funciones y subfuncionalidades. El resultado es una estructura jerárquica y de las funciones o funcionalidades y la definición de las interfaces funcionales. La ventaja de la descomposición funcional es que la especificación es escrita en el lenguaje y concepto de quienes implementan. Esto fomenta una buena comunicación de los requerimientos hacia los diseñadores y codificadores. La traducción al diseño y la codificación es sencilla debido a que la especificación de los

requerimientos está escrita en términos del espacio de la solución que se necesita⁶⁵.

5.1.2. Análisis Estructurado

Este modelo está basado en la premisa que expone que las dificultades accidentales pueden ser enfrentadas con una aproximación sistemática del análisis del problema usando:

- Un modelo conceptual común para describir todos los problemas.
- Un conjunto de procedimientos que sugieran la dirección general del análisis y brinden un orden de pasos para el mismo.
- Una serie de pautas o soporte heurístico de decisiones acerca del problema y su especificación.
- Una serie de criterios para evaluar la calidad del producto⁶⁶.

Dentro de las prácticas comunes del análisis estructurado están los diagramas de flujo y los diccionarios de datos. Para tratar los problemas de comunicación y comprensión del espacio del problema, este tipo de herramientas utilizan un conjunto de estructuras conceptuales –una representación gráfica de la especificación en términos de de estas estructuras- basándose en la hipótesis que la descomposición del problema en términos de datos que el sistema maneja será más clara y

⁶⁵**FAULK**, Stuart R. Software requirements: A tutorial. En: **THAYER**, Richard y **DORFAM**, Merlin. Software Requirements Engineering. 2 ed. Los Alamitos, California: IEEE Computer Science Press, 2000. p. 158 - 179.

⁶⁶Ibid.

menos inclinada al cambio que otra basada en las funciones que el sistema debe realizar⁶⁷.

Esta técnica ha evolucionado y es ampliamente utilizada dentro del análisis de requerimientos pero también es criticada debido a sus falencias. Uno de los aspectos más criticados es que este tipo de análisis no provee suficiente asistencia ni guías. Los analistas tienen dificultad para decidir las partes del problema que deben ser modeladas y cómo deben ser modeladas. Por otro lado, mientras los pasos del proceso están definidos a grosso modo, las demás acciones que se deben seguir en el proceso son muy generales y difíciles de llevar a cabo, en especial si se aplican métodos heurísticos para obtener resultados. Esto conlleva a pensar que el análisis estructurado no facilita la formulación de un SRS (**Software Requirement Specification**) o documento de especificación de requerimientos de software que sea claro y con los atributos suficientes para satisfacer a todos los participantes del proyecto.

5.1.3. Especificación Operacional

Este modelo se enfoca principalmente en solucionar dos de los dilemas más importantes que rodean a los requerimientos. El primero, es que las personas que están involucradas en el proceso de desarrollo no saben que desean construir, sino hasta que lo construyen. El segundo dilema, es el problema que se encuentra inmerso en el paso que implica pasar de una especificación de requerimientos a un diseño que satisfaga esa especificación. Entre más cercana esté la especificación del diseño,

⁶⁷FAULK, Op. cit.

mejor y más fácil será la transición entre estas actividades, pero así mismo entre más cercanas son, las decisiones de diseño se convierten en decisiones prematuras.

Los elementos claves para una especificación operacional son:

- Un lenguaje de especificación formal.
- Un motor que permita obtener especificaciones correctas escritas en el lenguaje ya mencionado⁶⁸.

Esta aproximación también debería incluir un soporte automatizado para el análisis de las propiedades de la especificación formal y métodos para transformar dicha especificación en su correspondiente implementación.

5.1.4. Análisis Orientado a Objetos

Este modelo basa sus principios en realizar modelos de la información y el diseño orientado a objetos. Las técnicas del **AOO** (análisis orientado a objetos) difieren del análisis estructurado en la forma en que se descomponen los problemas en sus partes, y los métodos a través de los cuales se descubren las relaciones entre dichas partes.

En este enfoque, el analista descompone el problema en un conjunto de objetos que interactúan entre sí, basados en las entidades y relaciones que existen en el dominio del problema. Un objeto encapsula un conjunto de datos, procesos y estados relacionados. En general, los componentes del AOO son los objetos, sus datos y servicios que prestan, y las relaciones que tienen con otros objetos.

⁶⁸**FAULK**, Op. cit.

Este método representa muy bien el comportamiento del dominio de la aplicación que se desea realizar, pero no es soportado por un modelo conceptual que muestre el comportamiento del dominio del negocio. Otro serio problema, es que la generalidad de esta aproximación se desvía más en el desarrollo de la aplicación y no permite concluir el objetivo específico de obtener un SRS completo.

5.2 OTROS MODELOS

Existen aproximaciones basadas en preceptos de las diferentes perspectivas, que satisfacen algunas de las actividades del análisis de requerimientos. Cysneiros⁶⁹, propone la utilización de grafos, en este caso dirigidos; en el análisis y descomposición de requerimientos No Funcionales (RNF). En este caso, el autor sugiere que los RNF representen *metas* a ser satisfechas. Cada meta debe ser descompuesta en subsecuentes submetas y estas a su vez descompuestas de igual manera hasta que se encuentren *operacionalizaciones*; estas son elementos que representan acciones o atributos que claramente identifican lo que es necesario para satisfacer la meta principal. Esta estructura permite el análisis de interdependencias y la detección de conflictos entre los RNF.

⁶⁹**CYSNEIROS**, L. M. Requisitos Não Funcionais: Da Elicitação ao Modelo Conceitual. Tesis Ph. D., Departamento de Informática, Pontificia Universidad Católica de Río de Janeiro, Río de Janeiro, Brasil, 2001.

De igual manera, se ha estudiado el uso de grafos y redes para mejorar la comprensión y comunicación de los requerimientos entre los **stakeholders** a través de modelos mentales y estructuras de pensamiento. Kudikyala, Lu y Torres; proponen en sus respectivos trabajos^{70 71 72}, clasificar requerimientos previamente recolectados y definidos en un documento de especificación; en grupos relacionados de acuerdo a la percepción de cada uno de los participantes involucrados. Estos grupos de requerimientos son transformados en redes, y aplicando algoritmos para la búsqueda de caminos (**Pathfinder Networks**), se establecen nuevas redes con un valor informativo y grupos de requerimientos interrelacionados (**Cluster**). De esta manera se realizan comparaciones entre las diferentes redes generadas y se buscan similitudes y no similitudes entre las mismas, para determinar conjuntos de requerimientos que sean conformes a los participantes del proyecto, y conjuntos de requerimientos que no correspondan a una correcta definición de requerimientos de acuerdo a las perspectivas de los **stakeholders**. En general, todas estas aproximaciones recrean un modelo que permite llevar acabo alguna tarea específica de la ingeniería

⁷⁰**KUDIKYALA**, U. K. Reducing misunderstanding of software requirements by conceptualization of mental models using pathfinder networks, Disertación de Ph. D., Departamento de Ciencias de la Computación, Mississippi State University, Mississippi, U.S.A., Agosto de 2004.

⁷¹**LU**, X. Using pathfinder networks to analyze and categorize software requirements. Reporte Técnico MSU – 030922, Requisito para la obtención del título de Maestría en Ciencias de la Computación, Departamento de Ciencias de la Computación, Mississippi State University, Mississippi, U.S.A., Agosto de 2002.

⁷²**TORRES**, M. An automated tool for software requirement refinement and pathfinder networks generation. Reporte Técnico MSU – 030922. Requisito para la obtención del título de Maestría en Ciencias de la Computación, Departamento de Ciencias de la Computación, Mississippi State University, Mississippi, U.S.A., Agosto de 2003.

de requerimientos, aprovechando como recurso estructuras ampliamente conocidas como los grafos o redes.

Cabe recalcar que estas estructuras son tal vez la representación matemática más utilizada en los problemas de la vida real por su versatilidad para representar problemas.

6. PROPUESTA CONCEPTUAL

Para diseñar y elaborar la propuesta conceptual de una herramienta de análisis de requerimientos, se estudiaron las diferentes actividades y tareas que se llevan a cabo en esta fase, así como los diferentes aspectos que influyen en las mismas. Esto se realizó, con el propósito de identificar problemáticas y oportunidades, para recrear funcionalidades computacionales que facilitaran dichas actividades.

Además, en este capítulo se exponen los aspectos más relevantes del modelo de análisis de requerimientos resultado de la investigación, y cómo se pueden realizar algunos procedimientos de este modelo a través de técnicas computacionales. También se exponen las características que deben poseer las fases de recolección y análisis, de manera que satisfagan el modelo propuesto, y cómo este es aplicable de manera transversal, a cada uno de los diferentes niveles en los cuales puede residir el proceso de ingeniería de requerimientos de la pequeña empresa desarrolladora de software en Bogotá D.C. (para mayor información sobre las características de operabilidad de la herramienta de software véase Anexo F...).

6.1 INGENIERÍA DE REQUERIMIENTOS DEL MODELO

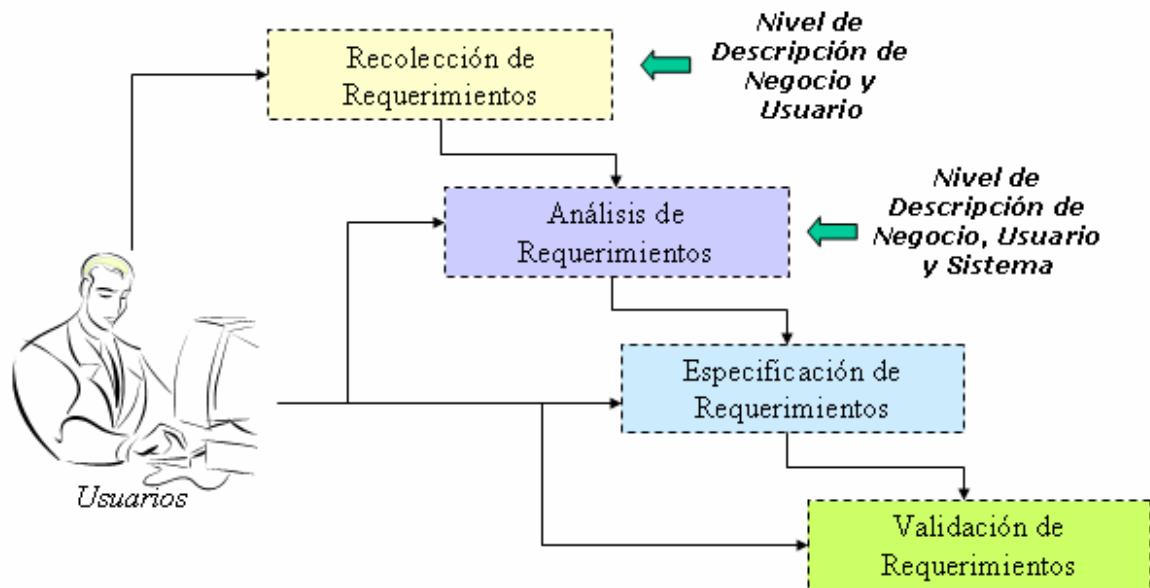
Un modelo de análisis de requerimientos, no es funcional sin la descripción de su relación con las otras fases de la ingeniería de

requerimientos. Por esta razón, es necesario extender el alcance del modelo a la fase de la cual depende: la recolección de requerimientos.

El modelo de análisis de requerimientos definido, parte de la estructura de un proceso estándar de ingeniería de requerimientos (...Véase figura 3...). En este modelo, se enfatizará en las dos primeras fases que corresponden al alcance definido para esta investigación, y en las características y niveles de descripción que deben tener los requerimientos en estas dos fases (...Véase figura 5...).

Es importante analizar estos aspectos, para poder establecer cómo mejorar la calidad del análisis de requerimientos (uno de los objetivos de la investigación). Para esto, se debe partir de la siguiente premisa: el factor más relevante en el éxito de un proceso de ingeniería de requerimientos, es la calidad con que se desarrolla el mismo. Por esto la herramienta de software se relaciona con el modelo, de manera que actúa como un elemento de soporte, para facilitar algunas actividades definidas en el modelo, únicamente en la fase de análisis de requerimientos.

Figura 5. Niveles de descripción de requerimientos utilizados en las diferentes fases del proceso de ingeniería de requerimientos.



Cada una de las fases de este esquema de ingeniería de requerimientos, es estándar, y no está relacionada con ninguna metodología ni tecnología en particular, de manera que se puede implementar para los diferentes niveles de madurez de las empresas, en los cuales el proceso sea al menos definido y repetible (niveles 2, 3, 4, 5, Tabla 3)⁷³. Esto es necesario debido a la gran cantidad de herramientas, metodologías, estándares y demás tecnologías que pueden aplicarse para llevar a cabo las actividades del proceso de ingeniería de requerimientos en las pequeña empresa desarrolladora de software en Bogotá D.C.⁷⁴.

⁷³Las empresas con un nivel de madurez inferior deben mejorar en primera instancia, TODA la definición de sus procesos para ser más competitivas, y como primer elemento para mejorar la calidad de sus desarrollos.

⁷⁴No se determinaron todas las herramientas ni métodos utilizados en la pequeña empresa, sin embargo, la encuesta realizada sobre las mismas, mostró que se utilizan diferentes métodos y herramientas durante los procesos de requerimientos.

6.1.1 Recolección de requerimientos

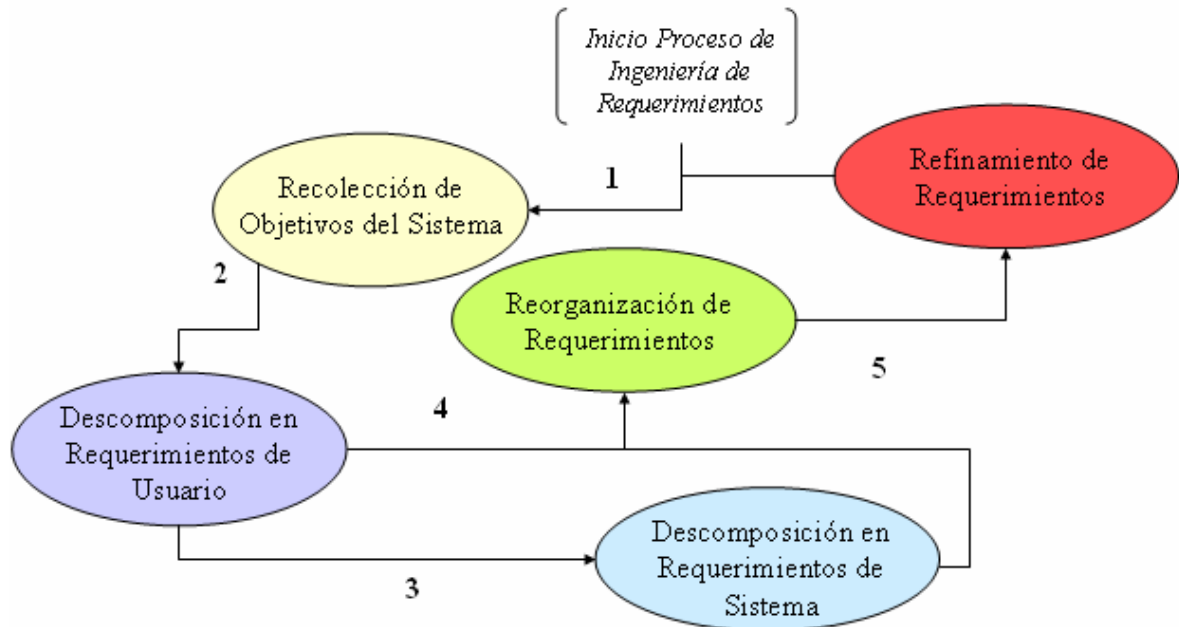
Como se explicó anteriormente, las actividades que se realizan en la fase de recolección de requerimientos, son determinantes a la hora de realizar el análisis de los mismos, ya que constituyen la base para dicho análisis.

Para la fase de recolección de requerimientos, se definieron un conjunto básico de actividades necesario para satisfacer las restricciones del análisis. Este conjunto de actividades, se basa en dos enfoques del proceso de recolección. El primero, propuesto por Marcia Carvalho y Zair Abdelouahab⁷⁵, y el segundo, propuesto por Ian Sommerville⁷⁶. Ambos enfoques son abiertos y modularizados, y favorecen el desarrollo de esta fase a través de actividades desarrolladas en etapas estándares. El conjunto de actividades que se debe llevar a cabo en esta fase son:

⁷⁵**CARVALHO**, Marcia y **ABDELOUAHAB**, Zair. Un metodo para elicitação e moderação de requisitos baseado em objetivos [online].

⁷⁶**SOMMERVILLE**. Ingeniería de software. Op. cit.

Figura 6. Modelo de proceso para Recolección de Requerimientos.



En la Figura 6, se puede observar un modelo de proceso para la recolección de requerimientos, constituido por un conjunto de actividades encaminadas a soportar y facilitar la comprensión del análisis de requerimientos.

No existe un único método, restrictivo a través del cual se deban llevar a cabo las actividades descritas en la Figura 6. Las empresas que cuentan con un proceso definido y maduro en el cual se realizan en la actualidad estas actividades, no poseen ningún inconveniente para aplicar el modelo de análisis de requerimientos. Sin embargo, las empresas que no cuentan con un proceso de recolección definido (nivel 0 ó 1), y por tanto, no realizan en la actualidad estas actividades; deben utilizar alguna herramienta, técnica o estrategia a través de la cual se lleven a cabo las actividades formuladas en el modelo.

A continuación, se exponen algunas sugerencias de estrategias para llevar a cabo cada una de esas actividades. Estas estrategias son representadas a través de los enlaces de la figura 6.

- **Recolección de objetivos del sistema:** es la primera actividad que se debe llevar a cabo en esta fase. En ella, se busca identificar los objetivos del sistema, que corresponden a los requerimientos del sistema descritos en un nivel de descripción de negocio (...Véase Sección 3.1.3...).

A continuación se explican algunas de las estrategias recomendadas para llevar a cabo esta actividad (...Véase el enlace **1** en la Figura 6...):

- Identificación de objetivos iniciales: consiste en extraer los objetivos del sistema, a partir del análisis de la documentación existente o de entrevistas realizadas a los **stakeholders**. El análisis de documentación existente se realiza examinando, ya sea las entrevistas realizadas a los usuarios del sistema en búsqueda de palabras (verbos) que representen una acción; o, analizando cómo expresan sus requerimientos en forma de operaciones, procesos o diagramas de flujo, en cualquier documento (normalmente informal) que exista acerca del dominio del negocio donde la aplicación será desarrollada.
- Preguntas específicas: consiste en establecer un conjunto de preguntas específicas previamente diseñadas por el ingeniero

de requerimientos (u otro participante que asuma ese rol), con el fin de aplicar en una entrevista o cuestionario a los diferentes **stakeholders**, de manera que se identifiquen los objetivos del sistema.

- Plantillas: esta estrategia propone el uso de una plantilla para la recolección de los objetivos del sistema. Esta, debe contener campos para ingresar información, como un número de identificación del objetivo, descripción del objetivo y otros definidos por el ingeniero de requerimientos.
- Estrategia lingüística: consiste en la formalización de un lenguaje sobre el cual se especifique y recolecte la información correspondiente a los objetivos del sistema.
- **Descomposición de los objetivos del sistema en requerimientos de usuario**: esta actividad persigue como objetivo el descomponer los objetivos previamente establecidos, en requerimientos más detallados (requerimientos de usuario), enfocados en la funcionalidad del negocio sobre el cual se está trabajando. Esta etapa es realizada para cada uno de los objetivos de negocio identificados en la fase anterior. Para las empresas que no cuenten con un proceso maduro, y se encuentren en los niveles inferiores (nivel 0 y 1); se recomienda el uso de escenarios y/o casos de uso. Para cada una de estas herramientas, se analizarán algunas de las estrategias que se pueden utilizar para llevarlas a cabo. (...Véase enlace **2** en la Figura 6...).

- Casos de uso.
 - Plantillas: esta estrategia propone especificar casos de uso por medio de una plantilla, al igual que lo descrito para los objetivos del sistema. Esta debe contener la información correspondiente al requerimiento que se desea especificar. En este documento se expone la plantilla que sugieren Durán, *et al.* Esta son plantilla contiene los campos estándar para los casos de uso (entendidos como requerimientos funcionales del sistema). Para esta plantilla, se utiliza una notación que permite implementar patrones lingüísticos a través una notación específica. Esta notación está compuesta así: Las palabras al interior de < > deben ser remplazadas de la manera apropiada, y las que se encuentran al interior de { } y separadas por coma, representan opciones excluyentes de manera que sólo una puede ser seleccionada (...Véase Tabla 4...).

Tabla 4. Plantilla para la descripción de requerimientos funcionales.

RF - <id>	<Nombre descriptivo>		
Versión	<Número de versión actual> (<Fecha de versión actual>)		
Autor	<Autor de versión actual> (<Compañía del autor>)		
Fuente	<Fuente de versión actual> (<Organización fuente del requerimiento>)		
Propósito	<Propósito del requerimiento>		
Descripción	El sistema debe comportarse como se describe en la siguiente secuencia de interacciones cuando <evento disparador>		
Alcance Y Nivel	<Que sistema es considerado caja negra en la perspectiva de diseño>		
Precondición	<Lo que se espera del entorno para la realización del requerimiento>		
Condición de Éxito	<Estado del entorno que indica que se realizó con éxito>		
Condición de Fracaso	<Estado del entorno cuando el requerimiento es abortado o falla>		
Actores	<Primario> { <Secundarios> }		
Evento Disparador	<Evento disparador o trigger>		
Secuencia Normal	Paso	Acción	
	
	n	{ el { <actor>. Sistema} <acción	
		n.1	{ el { <actor>. Sistema} <acción ejecutada por actor/sistema>, Pasos descritos en <caso de uso (RF-x)> es ejecutado}
	
	
Postcondición	<Postcondición del caso de uso>		
Excepciones Y Extensiones	Paso	Acción	
	p	if <condición de excepción>, { el {	
	
INFORMACIÓN RELACIONADA			
Prioridad	<grado de prioridad del requerimiento>		
Frecuencia	Este caso de uso se espera ser ejecutado <número de veces> número de veces/ <unidad de tiempo>		
Desempeño	Paso	Acción	
	q	m segundos	
	
Canales hacia los Actores	<Archivos, interactivo, base de datos, etc>		
Características Abiertas	<Lista de características que pueden afectar las decisiones sobre el caso de uso>		
Superiores	<Lista de requerimientos que incluyen este requerimiento>		
Subordinados	<Vínculos hacia subrequerimientos>		
Comentarios	<comentarios adicionales acerca del requerimiento>		

COCKBURN, Alistair. Basic Use Case Template [online]. Disponible en Internet:
< <http://members.aol.com/acockburn> >

La correspondiente definición de cada uno de los campos de esta plantilla es:

- **Versión:** De acuerdo a las recomendaciones de la IEEE⁷⁷, y de compañías líderes en procesos de software como Rational⁷⁸, un requerimiento debe permitir manejar sus distintas versiones de manera que se pueda analizar la evolución del mismo a través del tiempo. Para todo requerimiento, este campo debe estar ocupado por la versión actual, con su correspondiente número y fecha.
- **Autor, Fuente:** Contiene el nombre de la organización y el autor del requerimiento.
- **Propósito:** Contiene la descripción del porqué el requerimiento consignado es necesario para alcanzar los objetivos del negocio⁷⁹.
- **Descripción:** Para los requerimientos funcionales, esta plantilla contiene un patrón lingüístico que indica que debe ser llenado y los eventos que disparan el requerimiento.
- **Alcance y Nivel:** Cuál sistema debe ser considerado como caja negra para este requerimiento.

⁷⁷**INSTITUTE FOR ELECTRONICS AND ELECTRICAL ENGINEERS.** IEEE Guide for Developing System Requirements Specifications IEEE/ANSI Standard 1233–1996. *s.l.:* La institución, 1996.

⁷⁸**OBBERG**, Op. cit.

⁷⁹**DAVIS**, A. M. 201 Principles of Software Development. McGraw–Hill, 1995.

- **Precondición:** Se ingresan las condiciones necesarias que se deben tener para llevar a cabo el requerimiento que se está describiendo.
- **Condición de Éxito:** Condición que indica si la ejecución del requerimiento fue exitosa.
- **Condición de Fracaso:** Condición que indica que la ejecución del requerimiento fue abortada o falló.
- **Actores:** Actor primario, y/o lista de actores secundarios del requerimiento.
- **Evento Disparador:** Evento que dispara la realización del requerimiento.
- **Secuencia Ordinaria:** Aquí se deposita la secuencia de interacciones que tiene el usuario con el sistema, en orden de llevar a cabo una funcionalidad u operación. Los pasos que se llevan a cabo en pueden a su vez contener sub-pasos o secuencias anidadas, asumiendo que sólo una de estas se puede llevar a cabo.
- **Poscondiciones:** Son los estados a los cuales se debe llegar o los resultados que se deben obtener después de ejecutar la funcionalidad descrita en el requerimiento.
- **Excepciones Y Extensiones:** Durante la interacción descrita en la secuencia ordinaria se pueden presentar excepciones o extensiones condicionales, debido a flujos alternativos de dicha interacción entre el usuario y el sistema. Es este campo se especifica la secuencia que se tomaría si se presenta la excepción o se indica si la funcionalidad descrita en el requerimiento se aborta.

- **Prioridad:** Indica que tan importante es el requerimiento, que grado de prioridad tiene el requerimiento para los usuarios y clientes⁸⁰.
- **Frecuencia:** Frecuencia con la que se lleva a cabo la ejecución del requerimiento en un intervalo de tiempo. Se indica la tolerancia a fallos que tiene el requerimiento de acuerdo al número de eventos que se utiliza el requerimiento y el tiempo que el mismo debe estar disponible.
- **Desempeño:** Aquí se especifica un tiempo de tolerancia respecto al tiempo de respuesta del sistema para alguno o todos los pasos de la secuencia ordinaria.
- **Canales hacia los actores:** Canales como archivos, interactivo, base de datos a través de los cuales se expresa el resultado de la ejecución del requerimiento
- **Características Abiertas:** Lista de características que pueden afectar las decisiones sobre el caso de uso
- **Superiores:** Lista de requerimientos que incluyen este requerimiento
- **Subordinados:** Vínculos hacia subrequerimientos.
- **Comentarios:** En este campo se ingresa toda la información que no se pudo agregar a ninguno de los campos.

⁸⁰DAVIS, Op. cit.

Así mismo, se debe puede tener una plantilla para la descripción de requerimientos no funcionales (...Véase Tabla 5...).

Tabla 5. Plantilla para la descripción de requerimientos no funcionales.

RN-3	Sistema Operativo
Versión	1,0 (May, 25, 2005)
Autor	M. Ariza(Universidad de Javeriana)
Fuente	M. Torres (Proyecto Empresa de Teléfonos de Bogotá)
Descripción	El sistema debe operar sobre el Sistema Operativo Linux
Prioridad	Alta
Comentarios	Verificar la compatibilidad entre versiones diferentes de Linux

DURÁN TORO, A. et al. A requirements elicitation approach based in templates and patterns [online]. Disponible en Internet:
<http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER99/toro.pdf>

- Gráficas y Diagramas: Los casos de Uso son construidos conforme a la notación estándar de UML, donde la descripción de los casos de uso permiten múltiples actores, existiendo una visión única del caso para cada actor envuelto en él.
 - Estrategia Orientada a Actores: propone identificar como primer paso a los actores del sistema, como medio para identificar los casos de uso.
- Escenarios **(2 – Alternativa B)**.

Para la especificación de escenarios, se debe establecer un conjunto de reglas que permitan definir el contenido de los mismos, ya sea estableciendo una platilla como en los casos de uso.

- **Recolección de requerimientos funcionales:** se proponen dos estrategias para la recolección de este tipo de requerimientos (...Véase enlace **3** en la Figura 6...).

- Refinamiento: se aplica sobre los casos de uso, de manera que se descomponen originando Requerimientos Funcionales de Sistema.
- Composición: permite identificar los objetivos complementarios de un requerimiento particular, y ayuda a identificar un paquete de servicios para un sistema determinado.

- **Reorganización de requerimientos:** en esta actividad de reorganización se deben establecer las dependencias entre los requerimientos, de modo que se organicen de manera que en conjunto, su definición se concentre en mostrar como se van a satisfacer los criterios definidos para los mismos previamente (...Véase enlace **4** en la Figura 6...). No se define una estrategia particular sugerida para esta actividad y se deja a libre elección cómo se lleve a cabo esta actividad.

- **Refinamiento De Requerimientos.**

En esta etapa, lo que se busca es el mitigar los errores de la definición de requerimientos existente. Se realiza un proceso en cual se pueden realizar las siguientes actividades: eliminar requerimientos duplicados, unificar sinónimos, descubrir inconsistencias, redundancias y

requerimientos incompletos, etc. Este proceso puede ser manual, o poseer algún nivel de automatización (...Véase enlace **5** en la Figura 6...). Al igual que en la etapa anterior, no se recomienda ninguna estrategia particular, y se deja abierto al gusto de la empresa, cómo llevar a cabo estas actividades.

6.1.2 Análisis de requerimientos

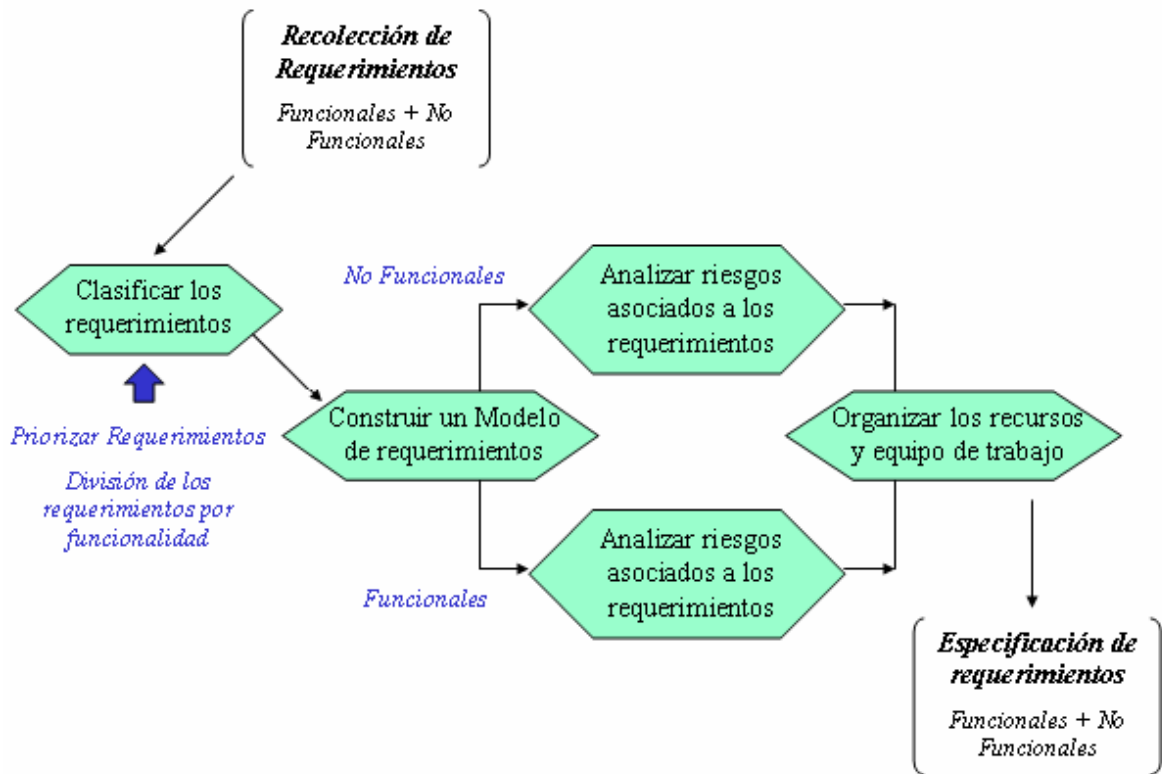
Las consideraciones iniciales, necesarias para el desarrollo de un concepto que cubra ampliamente las características indispensables para llevar a cabo un correcto y completo proceso de análisis de requerimientos, dentro de cualquier organización interesada en el desarrollo de un sistema o producto relacionado con software; están asociadas con los siguientes factores:

- Identificar y definir los requerimientos de software.
- Manipular estos requerimientos como ítems de configuración y administración, así como se realiza en otros productos del ciclo de vida de un proceso de desarrollo.
- Detectar y resolver conflictos entre requerimientos.
- Descubrir los límites del software y cómo este debe interactuar con su ambiente.
- Clasificar los requerimientos, en módulos, de acuerdo a su funcionalidad.

- Describir al máximo las características y atributos de los requerimientos⁸¹.

Para esta fase también se estableció un conjunto de actividades estándares a realizar dentro de un desarrollo de software (...Véase figura 7...), de manera que se cuente con un modelo transparente y flexible a los procesos ya establecidos dentro de las empresas, sin tener en cuenta el enfoque o perspectiva del mismo (...Véase sección 5.1...).

Figura 7. Modelo de proceso para análisis de requerimientos.



⁸¹ **SAWYER**, Peter y **KONTOYA**, Gerald. Software requirements. En: Software Engineering Book Of Knowledge (SWEBOK), capítulo 2. p.18. Disponible en Internet: <<http://www.swebok.org>>

Este modelo, define una serie de actividades que comparten las diferentes perspectivas sobre las cuales se puede llevar a cabo el análisis de requerimientos. Esto es relevante, debido a que una misma empresa puede utilizar cada una de las perspectivas, ya sea para un mismo proyecto, cómo para diferentes tipos de proyectos. Así mismo, se diferencia el trato que se le debe dar a los requerimientos funcionales, de los no funcionales. Esto se debe a las características propias de cada uno de los tipos de requerimientos.

Cada una de las actividades anteriormente nombradas, recopila una serie de tareas enfocadas a satisfacer los objetivos de la fase de análisis de requerimientos.

- **Clasificar los requerimientos:** esta actividad se encuentra orientada a facilitar la distinción de los diferentes requerimientos presentes en un proyecto de software. La clasificación de los requerimientos de un proyecto, representa el dividir los requerimientos por módulos del desarrollo, asociados por funcionalidad, y diferenciados por tipo, ya sea funcional o no funcional. También es necesario el establecer una medida de priorización de cada uno de los requerimientos, de manera que se facilite la toma de decisiones de desarrollo de los mismos.
- **Construir un modelo de requerimientos:** esta actividad se asocia a establecer una manera a través de la cual se puedan representar los requerimientos, sus características y propiedades. Esta actividad también representa escoger la perspectiva a través de la cual se

analizaran los requerimientos, y cómo se llevarán a cabo las actividades y tareas restantes del análisis de requerimientos.

- **Analizar riesgos asociados a los requerimientos:** con la realización de esta actividad se persigue el conocer los elementos de riesgo del desarrollo del sistema, antes de empezar a desarrollarlo. Los riesgos asociados a un requerimiento, pueden ser provenientes de algunas de sus características: costo de desarrollo, costo no desarrollo, tiempo de desarrollo, dificultad de implementación, dependencia de otra funcionalidad, etc. Estos diferentes aspectos deben ser evaluados de manera que se pueda asociar un riesgo determinado.

Es recomendable realizar el análisis de riesgos de los requerimientos, de manera separada para los requerimientos funcionales y no funcionales (...Véase Figura 7...). Esto se debe en gran medida, a que estos dos tipos de requerimientos poseen características, especificaciones y efectos sobre el sistema, diferentes entre sí.

- **Organizar los recursos y equipos de trabajo:** esta actividad implica establecer un orden de desarrollo de los requerimientos, establecer los recursos que se van a invertir en el desarrollo de los mismos (tiempo, dinero, desarrolladores).

6.2 ESTRUCTURA COMPUTACIONAL DEL MODELO

Las consideraciones y características del proceso de ingeniería de requerimientos anteriormente descrito se deben cumplir para satisfacer las restricciones de calidad, y funcionalidad antes de aplicar un modelo computacional de análisis de requerimientos. Ahora, es necesario analizar cómo relacionar el proceso de análisis de requerimientos, con elementos computacionales que faciliten, agilicen y mejoren su realización.

6.2.1 *Definición del Modelo*

El modelo trabaja sobre un grafo dirigido. Esta estructura fue seleccionada porque se considera que permite satisfacer algunas de las necesidades relacionadas con la fase de análisis. Los elementos de trabajo dentro de este marco los constituyen la definición y descripción de los requerimientos funcionales de un proyecto, y sus relaciones de dependencia (...Véase sección 6.1.1...).

Para entender el modelo hay que recurrir a la definición básica de grafo y los elementos que lo componen. La estructura tradicional de un grafo $G = (V, C)$, está constituida por un conjunto ordenado de elementos pertenecientes a los conjuntos V y C . Los elementos de V son conocidos como vértices (también llamados nodos o puntos). Los elementos de C son conocidos como caminos (también llamados arcos o líneas). Cada camino de C une dos diferentes vértices de V y es denotado como la

dupla (i, j) , donde i y j son dos diferentes vértices unidos por E . Cada uno de estos arcos puede tener una orientación (constituye un grafo dirigido) o puede no tenerla (grafo no dirigido). En el caso de que no sea dirigido la dupla (i, j) representa lo mismo que (j, i) . En caso de que se tratase de un grafo dirigido estas duplas constituyen caminos diferentes⁸².

En el caso particular de este modelo, los requerimientos funcionales representan cada uno de los nodos del grafo, y los caminos entre estos nodos representan la dependencia que existe entre los distintos requerimientos. Esto obliga a considerar esta estructura como un grafo dirigido, debido a la relación inequívoca de la dependencia entre requerimientos (requerimiento B depende de Requerimiento A, no es congruente con Requerimiento A depende de Requerimiento B). Por tanto, el grupo de los requerimientos funcionales se define como el conjunto:

$$G = G_F = (R_F, D_F) \quad (1)$$

Donde R_F está conformado por el conjunto de requerimientos funcionales del proyecto y D_F representa el conjunto de todas las dependencias existentes entre los elementos de R_F .

Las relaciones que componen D_F hacen referencia a la dependencia que tiene un requerimiento o conjunto de requerimientos sobre otro conjunto de requerimientos, ambos pertenecientes a R_F .

⁸²**CORMEN**, T., y **LEISERNON**, C. Introduction to Algorithms. 2da edición. Mc Graw Hill. MIT Press 2002.

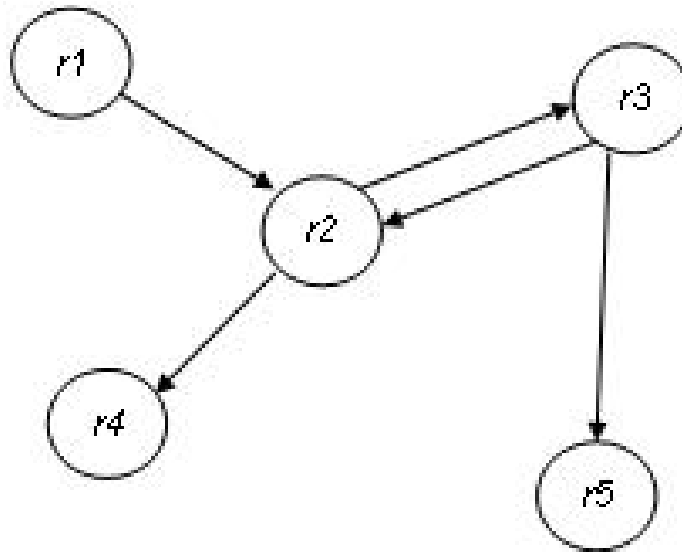
$$R_F = \{r_1, r_2, r_3, \dots, r_i, r_j, \dots, r_{N-1}, r_N\}$$

$$1 \leq i, j \leq N, i \neq j, N \in \mathbf{Z}^+$$

(2)

Luego, una relación de dependencia d_{ij} constituye un arco trazado desde el elemento r_i hacia el elemento r_j , indicando que para poder satisfacer la funcionalidad definida en r_j debe satisfacerse previamente la funcionalidad definida en r_i . Un requerimiento de software satisface una funcionalidad si al implementarla en el sistema sobre el cual se está trabajando, este cumple con los criterios de aceptación asociados al requerimiento (...Véase Figura 8...).

Figura 8. Representación básica del modelo propuesto, donde los nodos del grafo son los requerimientos funcionales, y los enlaces trazados entre ellos son las relaciones de dependencia.



Las relaciones de dependencia pueden clasificarse en dos tipos básicos: precedencia e interdependencia.

$$D_F = \{d_{ij} / d_{ij} = d(r_i, r_j), 1 \leq i, j \leq N, i \neq j, N \in \mathbf{Z}^+, \\ r_i, r_j \in R_F\}$$

(3)

Todas las parejas ordenadas de la forma $d_{ij} = d(r_i, r_j)$ constituyen relaciones de precedencia (...Véase Figura 9...). Asimismo, si la relación complementaria opuesta a una pareja d_{ij} , $d_{ji} = d(r_j, r_i)$ existe dentro del mismo conjunto D_F se habla de una relación de dependencia mutua (interdependencia) entre la pareja de requerimientos (r_i, r_j) (...Véase Figura 10...).

Figura 9. Representación de la relación de dependencia (Precedencia) entre dos requerimientos.

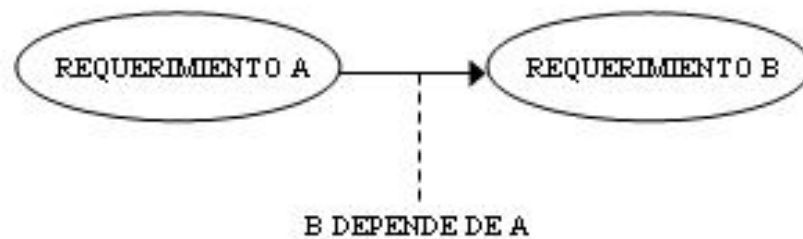
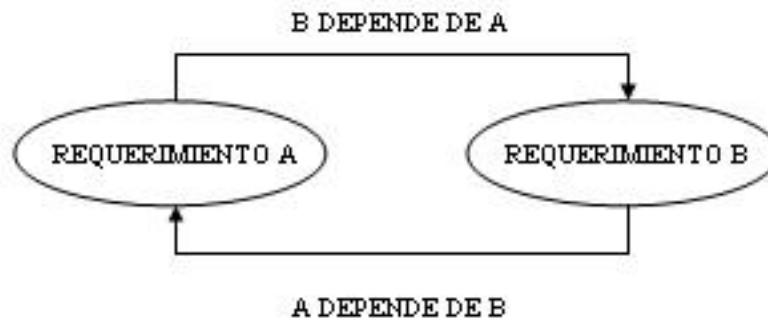


Figura 10. Representación de la relación de dependencia mutua (Interdependencia) entre dos requerimientos.



Ahora bien, esta representación es posible debido a la definición básica de los RF: constituyen aseveraciones de los servicios que el sistema debe proveer, como el sistema debe reaccionar a entradas particulares y como el sistema debe comportarse bajo situaciones particulares⁸³. Analizando los RF desde esta perspectiva, puede identificarse que la relación de dependencia entre dos requerimientos se da debido a alguno de estos tipos de restricciones: técnicas, de negocio y contractuales.

Las restricciones técnicas están relacionadas con las limitaciones de las plataformas y las arquitecturas de hardware y software que se utilizan dentro de un proyecto. Es debido a estas limitaciones que se pueden presentar dependencias entre elementos de una infraestructura tecnológica. Esta relación se da debido a que una funcionalidad puede depender de otra que la soporta a nivel tecnológico. Un ejemplo de esto lo constituyen las aplicaciones por capas. Las capas son independientes, pero las superiores no son funcionales si las inferiores que las soportan no existen. Por tal razón, en términos tecnológicos, existen dependencias para constituir funcionalidades, debido a la dependencia con los elementos, clases, componentes, infraestructura tecnológica y otras herramientas encargadas de soportar dicha funcionalidad (...Véase Figura 11...). Este modelo propone un concepto de desarrollo incremental e iterativo (...Véase Figura 12...), orientado a las características del sistema a desarrollar; para poder establecer la relación de dependencias entre requerimientos funcionales.

⁸³**SOMMERVILLE**, Ian. Software Engineering. Op. cit.

Figura 11. Estructura de servicios y funcionalidades que se presentan en un sistema modular. Muestra como los servicios que se encuentran más arriba en la estructura dependen de los servicios que se encuentran debajo de ellos.

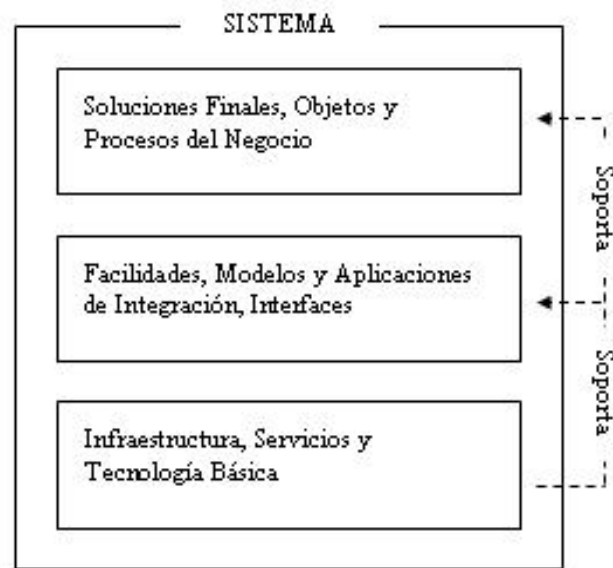
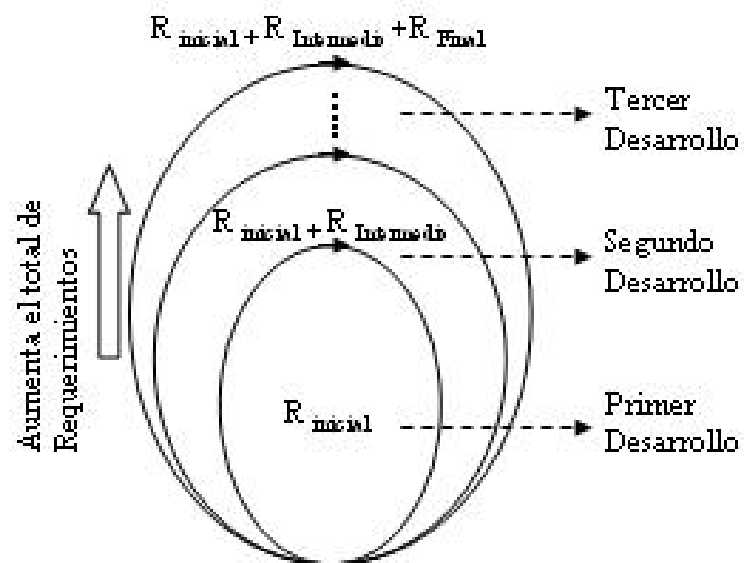


Figura 12. Desarrollo incremental de los requerimientos. El conjunto de requerimientos crece, de manera que se puede presentar dependencia entre los elementos de alguno de los subconjuntos.



Analizando esta estructura, se puede observar como los requerimientos funcionales pueden descomponerse en otras funcionalidades; es decir, otros requerimientos funcionales de menor jerarquía. Los requerimientos derivados de otros requerimientos se conocen como propiedades emergentes⁸⁴.

A nivel de negocio, también se pueden descomponer las funcionalidades de un sistema. En un conjunto de requerimientos puede establecerse una jerarquía en la cual los requerimientos se descomponen a su vez en otros requerimientos. Estos niveles en los cuales se descomponen los requerimientos se conocen como niveles de descripción (...Véase Figura 13...). A medida que un requerimiento se describe de manera más detallada nacen requerimientos funcionales y no funcionales asociados al mismo⁸⁵.

Por otro lado, se puede presentar dependencia entre dos requerimientos funcionales, de acuerdo a las necesidades del negocio que rodea a los mismos. Esta relación se da porque las funcionalidades que constituyen estos requerimientos son complementarias o, en otros términos, una funcionalidad enriquece a la otra. Esta afirmación se puede analizar a través de un sencillo ejemplo: Un sistema que permite realizar los cálculos básicos (suma, resta, multiplicación y división) entre dos números naturales (...Véase Figura 14...).

⁸⁴ **SAWYER**, P. y **KONTOYA**, G. Software requirements. En: (SWEBOK).Op. cit.

⁸⁵ **KONTOYA**, G. y **SOMMERVILLE**, I. Requirements engineering: Processes and techniques. Op. cit.

Figura 13. Estructura Jerárquica de requerimientos funcionales. Los requerimientos de alto nivel se descomponen hasta ser específicos.

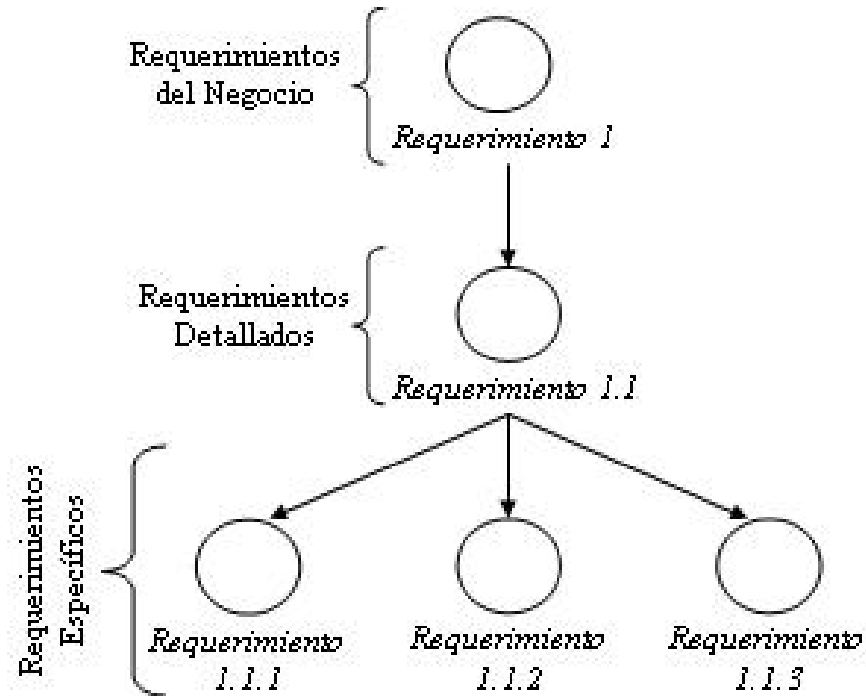
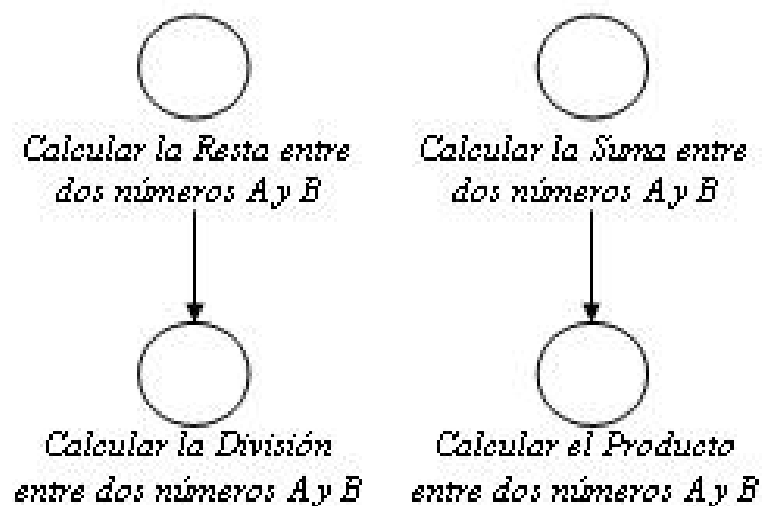


Figura 14. Requerimientos y sus dependencias derivadas del negocio para el sistema de una calculadora. En este caso las funcionalidades de Multiplicación y División son traducidas como extensiones de otras funcionalidades.



En este ejemplo, se presentan dependencias de precedencia entre requerimientos derivadas de la lógica de negocio. Esto sucede, debido a que el requerimiento que indica que el sistema debe permitir realizar el cálculo de el producto entre dos números naturales, se puede traducir en una funcionalidad que depende de que el requerimiento que permite llevar a cabo el cálculo de una Suma entre dos números naturales. Esto supone entonces una dependencia que implica que el requerimiento dependiente no se puede desarrollar hasta que el requerimiento precedente no haya sido desarrollado, o lo que es lo mismo, el requerimiento precedente debe ser desarrollado primero que el requerimiento que depende de su funcionalidad⁸⁶.

Finalmente, pueden establecerse dependencias entre requerimientos, si se relacionan estos a través de un contrato. Esta dependencia se basa en el precepto, que para algunos proyectos, se establecen objetivos (requerimientos funcionales de alto nivel) que deben ser satisfechos al término de la entrega del sistema. Para satisfacer las expectativas y condiciones estipuladas en el contrato, se estipulan subobjetivos que deben llevarse a cabo para poder aprobar si un requerimiento contractual se satisface con un sistema o no⁸⁷. Esto implica que el objetivo depende de la realización de sus subobjetivos, para satisfacer

⁸⁶ **Aclaración:** Pueden ser requerimientos independientes, o se puede empezar el desarrollo de requerimientos dependientes estableciendo una interfaz con sus respectivos datos compartidos, con el (los) requerimiento (s) que le precede (n), para no establecer una dependencia en el desarrollo. Sin embargo, la funcionalidad del requerimiento dependiente no podrá ser culminada si se presenta la inexistencia del (los) requerimiento (s) que le precede (n).

⁸⁷ **ANTÓN**, A. Goal Based Requirements Analysis. College of Computing, Georgia Institute of technology. Atlanta, Georgia; U.S.A. 1996.

sus condiciones de aprobación.

6.3 MAXIMIZANDO LA INFORMACIÓN DE UN REQUERIMIENTO

Para maximizar el valor informativo de los RF es necesario que en la fase previa al análisis de requerimientos se hayan obtenido como mínimo los siguientes datos o atributos para cada requerimiento: Prioridad, Costo, Volatilidad / Estabilidad, Dificultad, Riesgo, Tiempo Estimado de Desarrollo. Esto no restringe la recolección de otros atributos que pueden ser útiles en otras actividades, como: estado de desarrollo (estatus de desarrollo), persona encargada de su desarrollo, quién formuló el requerimiento, etc.

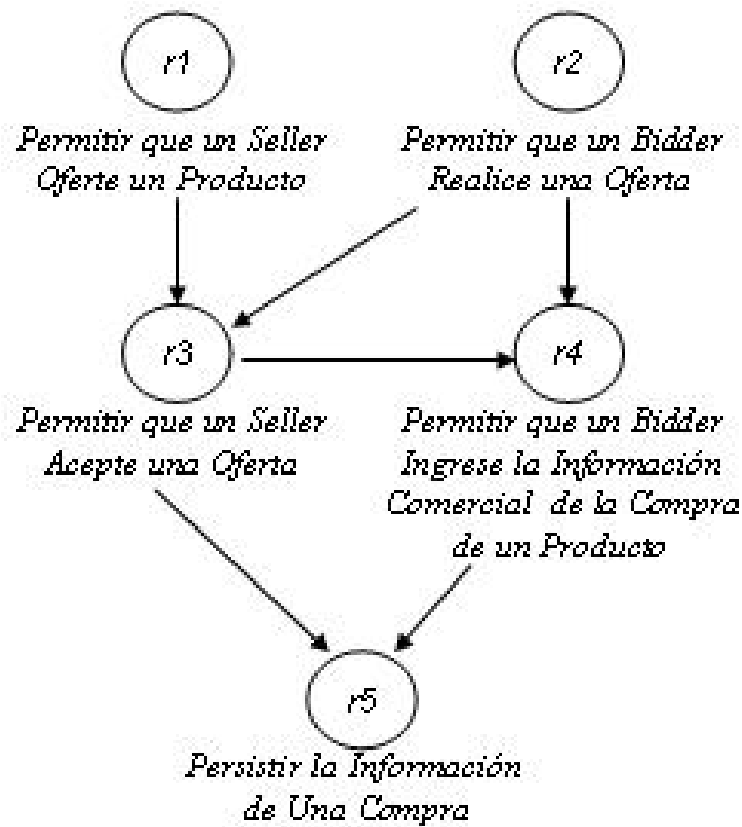
6.3.1 Análisis De Riesgos.

A través de la representación propuesta, se pueden identificar elementos de riesgo dentro del conjunto de Requerimientos Funcionales R_F . Esta operación tiene lugar gracias al análisis de las dependencias y atributos de cada requerimiento.

En este modelo, el riesgo asociado a un requerimiento se clasifica de acuerdo al número de requerimientos que le preceden directamente y al número de requerimientos que afecta este a su vez. De esta manera, un criterio para evaluar si un requerimiento es riesgoso, es determinar el número de requerimientos que le preceden directamente (Lista de antecesores directos) y/o el número de requerimientos que dependen de él. Considérese un sistema que permite realizar subastas, permitiendo la interacción entre la persona que oferta un producto (**Seller**) y la que realiza ofertas sobre el mismo (**Bidder**). Si se abstrae un pequeño

subconjunto de los requerimientos que este sistema debería tener, se obtiene una estructura como la siguiente (...Véase Figura 15...):

Figura 15. Pequeño conjunto de requerimientos funcionales y sus dependencias para el sistema de subastas.



Definiendo p_i^{in} como el número de arcos o dependencias que inciden o entran en un requerimiento r_i , y p_i^{out} como el número de arcos salientes del mismo; si se calculan estos valores para todos los requerimientos anteriormente mencionados, y además se tabulan los atributos de dificultad y riesgo de cada uno se obtiene lo siguiente:

Tabla 6. Tabla en la cual se tabulan las entradas y salidas por requerimiento y se anexaron los atributos de Riesgo y Dificultad estimados⁸⁸.

r_i	p_i^{in}	p_i^{out}	Prioridad	Dificultad
$r1$	0	1	Baja (1)	Baja (1)
$r2$	0	2	Media (2)	Media (2)
$r3$	2	2	Media (2)	Media (2)
$r4$	1	1	Alta (3)	Alta (3)
$r5$	2	0	Alta (3)	Bajo (1)

Si se analiza el riesgo de desarrollo de un requerimiento como una función de las dependencias, riesgos externos, y el grado de dificultad en el desarrollo, se puede expresar el riesgo de un requerimiento así:

$$riesgo(r_i) = f(p_i^{in}, p_i^{out}, prioridad(r_i), dificultad(r_i)) \quad (4)$$

En la fórmula anterior (4), los elementos p_i^{in} y p_i^{out} forman un mismo criterio de evaluación del riesgo asociado a un RF. Cabe resaltar, que el coeficiente p_i^{out} poseen más “peso” dentro de la función riesgo, que p_i^{in} . Esto se debe a una característica de los RF: si el desarrollo de un requerimiento funcional no se lleva a cabo, el sistema se ve “degradado” debido a que no permitiría realizar la funcionalidad descrita en el requerimiento⁸⁹. Por tanto se puede afirmar, que si de un requerimiento evaluado se desprenden otros requerimientos, entonces el no desarrollo del mismo degradaría la funcionalidad que le prosigue, o depende de él.

⁸⁸**Aclaración:** los datos de prioridad y grado de dificultad debieron haber sido establecidos previamente. Para el modelo, no es relevante el cómo se establecieron, sin embargo, la calidad y precisión de los mismos permitirá una mayor calidad en los resultados obtenidos.

⁸⁹**SOMMERVILLE**, Ian. Software Engineering. Op. cit.

Esto convierte en un factor de mayor riesgo el no desarrollo del requerimiento antecesor, frente a un requerimiento sucesor. Por otro lado, si un requerimiento depende de muchos otros requerimientos, el riesgo que se le asocia se valora de acuerdo a la consigna que su culminación dependerá de que se satisfaga(n) la(s) funcionalidad(es) que lo soporta(n).

Los otros dos criterios que se utilizan son atributos que se recolectan en la fase previa al análisis de requerimientos: la prioridad y la dificultad estimadas. Estos dos atributos son calificativos que se miden de manera cualitativa, pero son traducidos al modelo de manera cuantitativa. Para esto se establece una escala de valores entre los diferentes calificativos escogidos para clasificar los requerimientos. Para el ejemplo mencionado se seleccionaron valores de 1 a 3 y calificativos como Baja, Media, y Alta. Es importante seleccionar bien la escala no introduciendo demasiados calificativos, ya que el resultado de la fórmula se vería demasiado inclinado por estos dos elementos.

Para el caso anterior, la definición de una matriz de elementos de riesgos, es un elemento que permitiría clasificar los requerimientos según su riesgo asociado. Para esto, es necesario organizar los requerimientos teniendo en cuenta cómo primer elemento de clasificación a la prioridad asociada al requerimiento. Los demás atributos, constituyen los otros criterios a través de los cuales se clasifican los requerimientos en orden de riesgo. Su consecuente orden de importancia es: dificultad, p_i^{out} y p_i^{in} . Esto se debe a que la prioridad es atributo más acertado, estable y confiable de los cuatro atributos

mencionados, en la descripción de un requerimiento. Así mismo, pasa con el atributo dificultad. Y el orden de evaluación de riesgos relacionados con los enlaces entrantes o salientes a un requerimiento, se debe a la consideración analizada en la ecuación 4.

Tabla 7. Clasificación de los requerimientos funcionales en orden de riesgo creciente. El requerimiento $r1$ es el de menor riesgo, mientras que $r4$ tiene asociado el mayor riesgo.

$riesgo(r_i)$	<i>Prioridad</i>	<i>Dificultad</i>	p_i^{out}	p_i^{in}
$r1$	Baja (1)	Baja (1)	1	0
$r2$	Media (2)	Media (2)	2	0
$r3$	Media (2)	Media (2)	2	2
$r4$	Alta (3)	Bajo (1)	0	2
$r5$	Alta (3)	Alta (3)	1	1

Los resultados de aplicar los criterios de ordenamiento y evaluación de riesgo, con los datos de cada requerimiento nos indicaron que el requerimiento $r4$ es el requerimiento que posee mayores riesgos asociados, debido a sus precedencias, prioridad y dificultad de desarrollo. Por otro lado, el requerimiento $r1$ obtuvo el menor orden, lo cual lo convierte en el requerimiento más fácil de realizar (...Véase Tabla 7...). Esta medición del riesgo nos brinda otro criterio de decisión cuando se analizan los riesgos asociados a un proyecto, de manera que se puedan evaluar los efectos de los mismos, sus causas y posibles contingencias.

6.3.2 *Estimaciones*

Debido a las características del análisis sugerido en este modelo, se pueden realizar estimaciones que son importantes para el análisis y administración de un proyecto de software.

Esto es posible si se interpreta el desarrollo de cada requerimiento funcional como una actividad a realizar, u objetivo a satisfacer, dentro de un proyecto. Este paradigma se conoce como “Planeación de proyectos orientada por requerimientos” (**Requirements Driven Project Planning**)⁹⁰.

Dentro de las ventajas de este enfoque se encuentran: planear el desarrollo y otras etapas del proyecto a través de los requerimientos, se pueden hacer estimativos de las etapas de construcción del sistema y monitoreo del estado o porcentaje de desarrollo de los requerimientos. Aplicar este paradigma permite reaccionar al cambio en definiciones de requerimientos de una mejor manera. Por otra parte, el construir una estructura formalmente definida facilita la formalidad necesaria para establecer y medir el estado actual de un requerimiento.

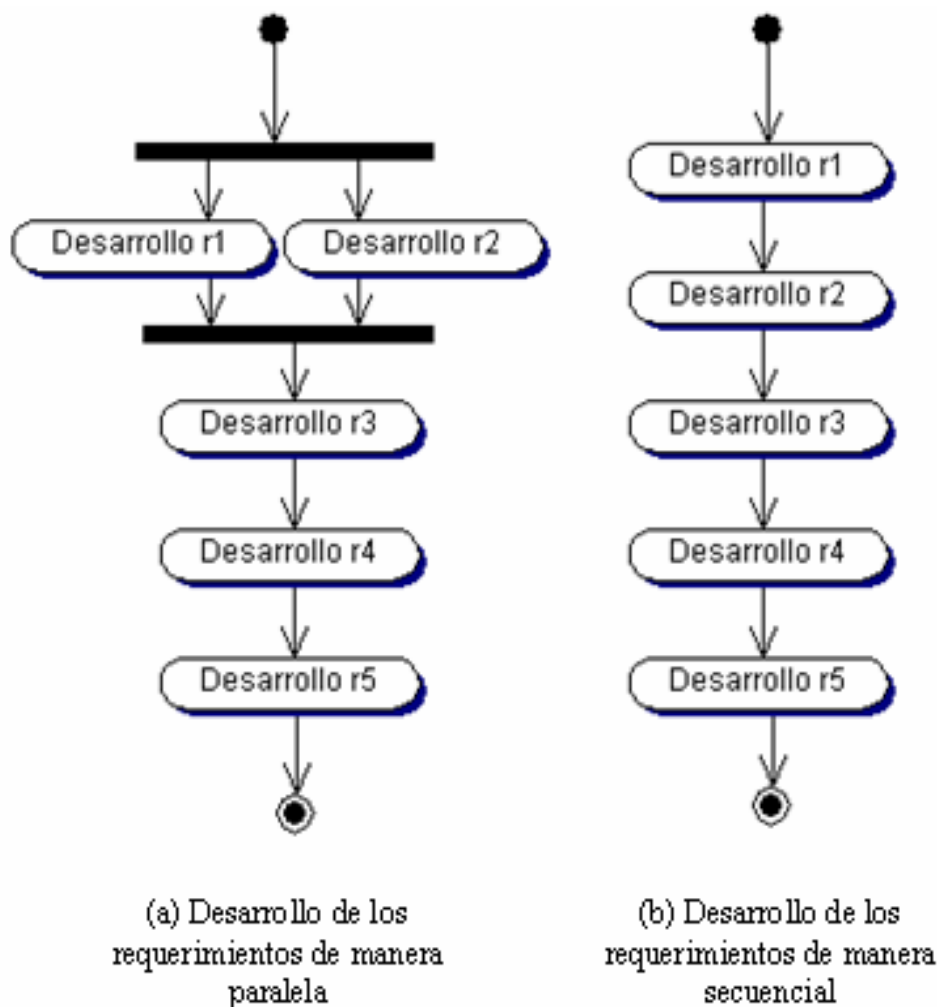
Analizando una de estas características, se pueden estimar los recursos necesarios para el desarrollo del proyecto. Se entiende por recursos, el tiempo, costo, herramientas y personal que se utilizarán para el desarrollo.

Para entender cómo puede realizarse la estimación del tiempo el

⁹⁰**Robertson**, Suzanne. Requirements Driven Software Planning.

desarrollo del proyecto como un conjunto de requerimientos, es necesario analizar las dependencias existentes entre los requerimientos, y el tiempo estimado de desarrollo de cada uno de los mismos. Interpretando estos datos, para el conjunto de requerimientos que se definió anteriormente (...Véase Figura 15...) se pueden conformar los diagramas (...Véase figura 16...):

Figura 16. Diagramas de Actividades, correspondientes al desarrollo de un conjunto de requerimientos de la figura (15).



Cada uno de estos diagramas constituye un orden de desarrollo sugerido, debido a que este modelo favorece el desarrollo de sistemas de manera incremental. El caso en que el desarrollo de requerimientos se lleva a cabo de manera paralela (...Véase Figura 16-a...), es decir, se cuenta con recursos para desarrollar todos los requerimientos que no dependientes, de manera paralela, en un determinado espacio de tiempo; debe interpretarse como un escenario favorable en el que el tiempo de desarrollo de todo el proyecto es mínimo (t_{min} , optimista). El segundo caso (...Véase Figura 16-b...), los requerimientos son desarrollados de manera secuencial; constituye el enfoque pesimista para el cual el tiempo de desarrollo del proyecto es máximo (t_{max}). El analizar cada una de estas posiciones extremas, nos permite inferir un rango estimado de tiempo, que tome el desarrollo del conjunto de requerimientos ($t_{Dev}(R_F)$), ver ecuación (5).

$$t_{min} \leq t_{Dev}(R_F) \leq t_{max} \quad (5)$$

Ninguno de estos órdenes indica o constituye una estructura estática, en la cual el conjunto de requerimientos a desarrollar sea estático a su vez. La interpretación correcta, nos indica cual sería el posible comportamiento del desarrollo del sistema en una determinada iteración, y fase de desarrollo, de acuerdo a los recursos que se tengan para llevar a cabo un desarrollo. Este análisis se basa en:

- Determinar cual(es) es (son) el(los) requerimiento(s) que no posee(n) ninguna precedencia, es decir, que no dependa(n) de

ningún requerimiento. Estos requerimientos constituyen la base sobre la cual se puede empezar el desarrollo de un sistema determinado (RF Iniciales).

- Establecer cual(es) RF no posee(n) ningún arco saliente, es decir que no sean precedencia de ningún otro requerimiento (RF Terminales).
- Analizar las dependencias de requerimientos de manera que se establezca la profundidad de cada uno de los requerimientos del grafo para permitir analizar si un conjunto de requerimientos se puede desarrollar de manera paralela o de manera secuencial. Para este caso, se entiende por profundidad de un requerimiento, el camino que recorra el mayor número de nodos del grafo sin repetir ninguno, partiendo desde cualquiera de los RF Iniciales. De esta manera, los requerimientos cuya profundidad sea igual se pueden desarrollar de manera paralela, y quienes no correspondan con esta comparación se desarrollan de manera secuencial.

Por otra parte, al definir actividades para el desarrollo de los requerimientos, estas pueden incorporarse en un diagrama de Gantt. Utilizando las propiedades de estos diagramas, se puede estimar el tiempo de desarrollo del conjunto total de requerimientos, así como el costo total de dicho desarrollo. En este caso, los elementos del conjunto de relaciones de precedencia representarían las actividades. Para esto es necesario utilizar los atributos previamente estimados de cada requerimiento funcional: tiempo estimado de desarrollo, y costo del

mismo.

6.3.3 Elementos Especiales y Fusión de Elementos Interconectados o Interdependientes.

Las funcionalidades anteriormente explicadas, aplicables al modelo, sólo se pueden llevar a cabo si la estructura del grafo que las soporta no posee elementos aislados ni interdependientes. Esto tiene como razón, el hecho que no se pueden establecer caminos ni actividades que pertenezcan a un ciclo o se encuentren interconectadas dentro de in diagrama de actividades o de de Gantt. Cómo se analizó anteriormente, esta situación se puede presentar, debido a las relaciones de interdependencia que se establezcan entre los RF, según la definición de requerimientos que se haya realizado previamente. Sin embargo, una definición de requerimientos que contenga este tipo de relaciones, no constituye una mala práctica, simplemente no cumple con las exigencias del modelo.

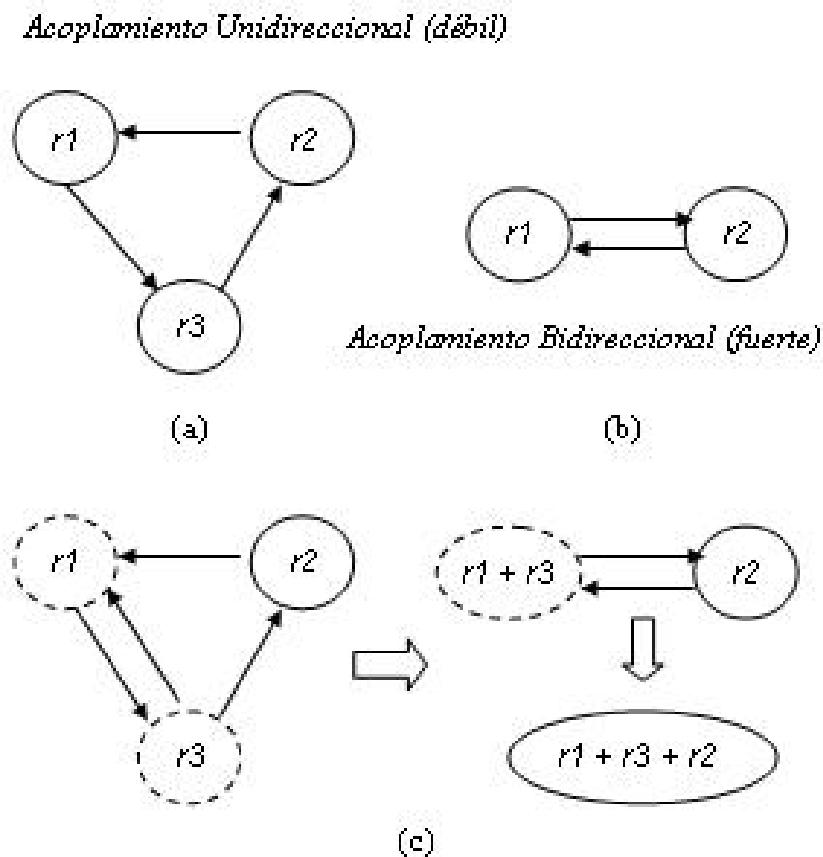
Por tanto, si se desea aplicar el modelo a estructuras que contengan estas interdependencias, es necesario “removerlas”. Para esto se utiliza la fusión de requerimientos. Se entiende por fusión de requerimientos, el tomar un conjunto de requerimientos, y agruparlos en uno, el cual, encapsularía toda la funcionalidad de los requerimientos que se fusionaron en él. Los siguientes, son los posibles casos de fusión de un conjunto de requerimientos:

- i. *Interdependencia Básica:* este caso se da cuando dos requerimientos son interdependientes entre sí de manera

directa. Para remover esta interdependencia, los dos requerimientos son fusionados en un único requerimiento (...Véase figura 17 b...).

- ii. *Interdependencia Cíclica:* Este caso se da cuando un subconjunto de requerimientos conforma un grafo cíclico a su vez. Todo ciclo debe ser reducido al menor conjunto de requerimientos posible, de manera que los mismos no presenten interdependencia (...Véase figura 17 a y c...).

Figura 17. Posibles casos de interdependencia. Interdependencia cíclica (a) y (c). Interdependencia básica (b) y (c).



Las figuras anteriores representan casos concretos de las posibles interdependencias (todas removibles). Para el caso (a) los tres requerimientos se consolidarían en un único requerimiento, y de igual forma para el caso (b). El caso (c) presenta los dos tipos de interdependencias, por lo tanto debe reducirse de manera que se obtenga un conjunto de requerimientos que no posean interdependencias; para este caso particular, un único requerimiento.

Estos procedimientos de fusión, también tienen una implicación dentro del concepto de requerimientos. Este tipo de componentes fuertemente conectados (interconectados) puede representar uno de los siguientes casos: un conjunto de requerimientos de alto acoplamiento entre sí, o una mala definición de requerimientos de software. Para cualquiera de los dos casos, el procedimiento de fusión permite mitigar los errores, conflictos e incongruencias de la definición de requerimientos de un proyecto. En el caso de un conjunto de requerimientos de alto acoplamiento; la fusión de sus requerimientos disminuye errores y problemas que se pueden presentar si no se considera a cada uno de esos requerimientos como parte de un elemento interconectado, esto, debido al riesgo asociado a su alto acoplamiento. En el caso de representar una mala definición de requerimientos, simplifica la definición de los mismos, facilitando a los analistas detectar cual fue el problema en la definición, ya sea por repetir requerimientos, definir requerimientos no realizables (incongruencia).

6.4 IMPLEMENTACIÓN DE LA FUNCIONALIDAD DEL MODELO

Para entender de manera clara y concreta, cómo estos procesos de análisis de requerimientos pueden ser asistidos por computador, es necesario analizar cuál es la estructura de los algoritmos a través de los cuales se pueden llevar a cabo los procedimientos mencionados para las actividades del análisis de requerimientos.

6.4.1 Componentes Fuertemente Conectados.

Este algoritmo (**Strongly connected components**), permite descomponer un grafo dirigido en sus componentes fuertemente conectados.

Un componente fuertemente conectado de un grafo dirigido $G = (V, C)$ es el máximo conjunto de vértices $SCC \subseteq C$ de manera que por cada pareja de vértices u y v presentes en SCC , existe un camino desde v hasta u , y uno desde u hasta v ; es decir, que tanto u como v pueden alcanzarse el uno desde el otro [Referencia Libro de Algoritmos]. Un grafo dirigido puede poseer más de un componente fuertemente conectado⁹¹.

La implementación de este algoritmo, es la necesaria para la detección de elementos interconectados, y fusión de los mismos, al interior del modelo. Cómo se puede apreciar (...Véase Figura 17...), cada uno de los gráficos concuerda con la descripción de un componente fuertemente

⁹¹CORMEN, T., y LEISERNON. Op. cit.

conectado. El orden de complejidad temporal de este algoritmo es lineal, lo que califica como una excelente solución para las actividades del modelo que se relacionan con él.

6.4.2 Organización topológica.

La organización topológica u orden topológico (**Topological sort**) de un grafo acíclico dirigido (GAD) $G = (V, C)$, es una ordenación lineal de todos los vértices de manera que si G contiene un camino (u, v) , entonces u aparece antes que v en el orden. Diferentes tipos de aplicaciones (incluyendo este modelo) utilizan los GAD para la representación de precedencias entre eventos⁹².

Este algoritmo sirve para la realización de algunas de las actividades descritas dentro de análisis de administración y análisis de recursos necesarios den del desarrollo (...Véase Figura 16...).

6.4.3 Búsqueda en profundidad

Como su nombre lo indica (**Depth-first search**), esta estrategia busca “el más profundo” siempre que es posible. Este algoritmo explora desde el vértice v descubierto más recientemente, que todavía posea vértices inexplorados. Cuando todos los vértices han sido explorados, la búsqueda realiza un proceso de **backtrack**, para explorar los caminos salientes del vértice desde el cual v fue descubierto. Este proceso continua hasta que se hayan descubierto todos los vértices que son

⁹²CORMEN, T., y LEISERNON. Op. cit.

alcanzables desde el vértice fuente u original. Si permanecen vértices en el grafo, sin ser descubiertos, entonces uno de ellos es seleccionado como la nueva fuente y la búsqueda se repite desde este origen. Todo este proceso es repetido de manera que no queden vértices en el grafo sin descubrir⁹³.

Este algoritmo es necesario para la elaboración de los algoritmos anteriormente seleccionados.

6.5 CONSIDERACIONES DEL MODELO

Para entender la estructura teórica, alcance y limitaciones de este modelo, es importante denotar que está orientado a integrar diferentes enfoques del análisis de requerimientos⁹⁴. Dentro de los diferentes objetivos del modelo se encuentran:

- 1) Descomponer el problema en partes manejables.
- 2) Organizar la información de los requerimientos de manera que pueda ser entendida y aprovechada.
- 3) Comunicar el problema a las diferentes partes involucradas, a través de un modelo intuitivo a todas las partes.
- 4) Establecer un conjunto de reglas que permitan orientar la dirección el análisis de requerimientos.
- 5) Definir un conjunto de guías y procedimientos heurísticos para soportar las decisiones de análisis sobre el problema.

⁹³**CORMEN**, T., y **LEISERNON**. Op. cit.

⁹⁴**THAYER**, Richard. Op cit.

Por otro lado, es importante entender el porqué de la definición particular del modelo, orientado exclusivamente a los RF. Partiendo del concepto de requerimiento funcional y sus características podemos inferir las siguientes afirmaciones:

- Expresan transformaciones que tienen lugar en el software.
- Por lo anterior, su construcción se puede medir y evaluar a través del tiempo, y se puede constatar su evolución y estado de desarrollo.
- Se refieren a rasgos particulares del sistema.
- Por lo anterior, el incumplimiento de alguno de estos sólo degradará el sistema.
- Se suele utilizar modelos conceptuales para su representación, es decir, se utilizan para describir de forma abstracta las funcionalidades y el dominio del problema sin entrar a evaluar consideraciones de tecnologías de la información o arquitectura⁹⁵.

Estas propiedades particulares de los RF, que los diferencian de los RNF; los habilitan como elementos válidos dentro del modelo propuesto. No se considera la aplicabilidad de los RNF, debido a una de sus principales características: afectan al sistema de manera transversalmente, de manera que su no realización puede inutilizar al mismo⁹⁶. Esto no permite establecer el mismo tipo de relaciones de dependencia entre RNF, por tanto el análisis de los riesgos asociados a dichos requerimientos no puede llevarse a cabo de la misma forma que con los RF.

⁹⁵Compendio de los diferentes autores de ingeniería de software.

⁹⁶**SOMMERVILLE**, Ian. Software Engineering. Op. cit.

De igual manera, el modelo hace un paréntesis respecto al tratamiento de los requerimientos funcionales que involucren características especiales como son los requerimientos relacionados con temas que están fuera de la priorización convencional. Estos requerimientos son aquellos que se encuentran relacionados con la seguridad de las personas o de los elementos del sistema, cuando estos se encuentran envueltos por ambientes especiales, que involucren e imprimen un alto riesgo al desarrollo de los mismos. Un ejemplo de estos ambientes, lo constituye un laboratorio farmacéutico o una planta nuclear⁹⁷.

Aun cuando las características fundamentales de los proyectos que envuelven este tipo de requerimientos son las mismas que en otro tipos de proyectos, los riesgos asociados a los requerimientos que comprometen la seguridad de las personas, ya sea los usuarios del sistema o los beneficiarios del mismo, establecen que el desarrollo se centralice en la importancia de garantizar el cumplimiento de estos requerimientos. Por tanto la aplicación del Modelo no es la más adecuada para priorizar el desarrollo de los requerimientos.

Por otra parte, el modelo se diferencia de otros modelos anteriormente propuestos, ya que integra diferentes enfoques y modelos de grafos, y particulariza en las ventajas de estos en el análisis de requerimientos. Además. Principalmente permite el análisis de riesgos y dependencias

⁹⁷**SOMMERVILLE**, I. *An Integrated Approach to Dependability Requirements Engineering*. Proc. 11th Safety-Critical Systems Symposium, Bristol. 3-15, Springer. 2003.

entre requerimientos y conjuntos de requerimientos, de manera que se encuentren elementos de riesgo y/o se detecten errores en la definición de requerimientos de manera que sea más fácil removerlos o corregirlos.

6.6 HERRAMIENTA COMPUTACIONAL

6.6.1 *Precondiciones de uso*

Apoyada sobre la definición del modelo, la estructura de la herramienta computacional orientada a satisfacer las necesidades del análisis de requerimientos comprende diferentes aspectos.

Al igual que en la definición del proceso, la estructura de la herramienta es aplicable a cualquier tipo de empresa interesada en el desarrollo de software, siempre y cuando el proceso de desarrollo de requerimientos satisfaga las necesidades del modelo.

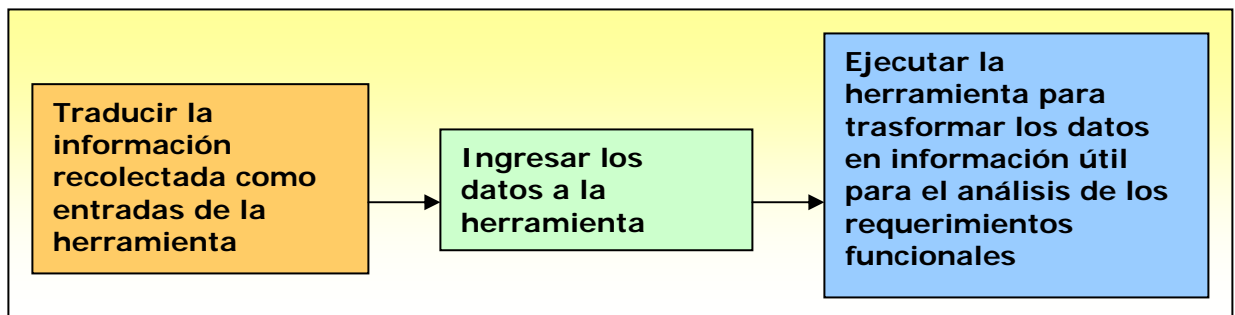
Debido a las limitantes de recursos inmersas al interior de la investigación, el desarrollo de la herramienta se realizó a través de un prototipo funcional evolutivo. Así mismo, la herramienta soporta las necesidades del modelo, así como las operaciones del mismo.

Por tanto, las condiciones previas que se deben dar para la utilización de la herramienta son:

- Realizar las actividades correspondientes a la fase de Recolección de requerimientos, de acuerdo a la estructura de proceso definida en la propuesta (...Véase sección 6.1.1...).

- Clasificar los requerimientos, de la manera definida por el modelo de proceso de la propuesta (...Véase sección 6.1.2...).
- Traducir la información recolectada en las actividades previamente mencionadas, en entradas de la herramienta.
- Ingresar la información a la herramienta.

Figura 18. Proceso de ejecución de la herramienta computacional.



6.6.2 Objetivos de la herramienta

Son varios los objetivos que se pretenden cumplir con el desarrollo de un prototipo de herramienta de análisis de requerimientos. Los primeros se encuentran relacionados con algunas actividades propias del análisis de requerimientos.

- Analizar riesgos asociados a los requerimientos funcionales.
- Permitir realizar estimaciones.
- Mejorar la comprensión, y definición de una especificación de requerimientos.

Estos objetivos están relacionados con las capacidades del modelo computacional definido para la herramienta. Sin embargo la herramienta está orientada a satisfacer otros objetivos relacionados con las necesidades de las herramientas más utilizadas dentro de las actividades de requerimientos (...Véase sección 3.2.2...).

- No cuentan con un método para analizar riesgos asociados a un requerimiento. La herramienta pretende implementar un método heurístico para llevar a cabo esta operación.
- No permiten visualizar una estructura gráfica que permita entender rasgos de un conjunto de requerimientos. La herramienta se orienta a permitir construir una representación gráfica de los requerimientos que sea de fácil comprensión.
- No se encuentran orientadas a la realización de actividades específicas de análisis de requerimientos.

6.6.3 Implicaciones de uso

El correcto uso de la herramienta trae consigo una serie de implicaciones alrededor del proceso de análisis de requerimientos dentro de proyectos de software, independientemente de su ámbito.

- Presupone la existencia de un proceso de ingeniería de requerimientos de alta calidad.
- El análisis de requerimientos asistido por herramientas computacionales e informáticas permite aprovechar los recursos inmersos en procesos de computación. Un claro

ejemplo de esto lo constituye el procesamiento de altos volúmenes de datos de manera más rápida que en un proceso manual.

- Permite realizar operaciones lógicas y matemáticas que implican una estructura de requerimientos definida, medición y comparación de los mismos. Esto es un elemento que permite elevar la calidad del análisis de requerimientos. *Todo aquello que se puede medir y evaluar está sujeto a mejorar su calidad.*
- Se debe imponer restricciones fuertes dentro del modelo para que el resultado de operar con el mismo sea el adecuado⁹⁸.

⁹⁸ **THAYER**, Richard. Op cit. p. 169.

7. ESTUDIOS

En esta sección se evalúa el modelo de manera teórica, y los resultados obtenidos al aplicar la herramienta dentro contexto de las pequeñas empresas desarrolladoras de software bogotanas.

7.1 EVALUACIÓN DEL MODELO

7.1.1 *Criterios de evaluación*

Para poder evaluar este modelo, se establecieron criterios para determinar su validez. El modelo, posee las características de un modelo conceptual, y por esta razón se establecieron los siguientes criterios de evaluación comunes a este tipo de modelos:

- Determinar que sea ***correcto***.
- Determinar que sea ***completo***.
- Determinar que sea ***consistente***.
- Determinar que sea ***comprensible***.

7.1.2 *Características a evaluar en el modelo*

Así mismo, se estableció un conjunto de características para evaluar la validez del modelo. Estas características son:

- Sintaxis del modelo
- Representación en el dominio del problema
- Trazabilidad.

Con la verificación de la sintaxis se pretende comprobar la conformidad del modelo a las reglas que se imponen a la hora de crearlo. Para que un modelo o diagrama sea conforme a las reglas, es importante que contenga información correcta y acertada.

Al evaluar el modelo, se debe analizar si el mismo describe y representa las funcionalidades del sistema que se obtuvieron en la recolección de requerimientos. Con esto se pretende comprobar si el modelo es correcto, y detectar si se omitieron necesarios para la representación de dichas funcionalidades. Para que el modelo sea correcto, es fundamental que represente adecuadamente la información del sistema proporcionada a través del enunciado.

La trazabilidad permite determinar el origen de cada elemento o componente en el modelo. Por tanto, la evaluación de la trazabilidad consiste en asegurar que siempre es posible determinar de donde provienen los elementos del modelo, o cual es su correspondiente representación en la información que lo alimenta. Esto también aplica a poder traducir la información en elementos del modelo (de la descripción del sistema al modelo y del modelo a la descripción del sistema).

7.1.3 Ejecución de la evaluación

Los criterios y características anteriormente descritas, fueron analizadas desde dos perspectivas: la teórica y la aplicada. Cada una de estas perspectivas fue analizada de una manera particular.

7.1.3.1 Perspectiva teórica. Desde esta perspectiva fácilmente se puede analizar la validez del modelo. Para cada una de las características, se puede establecer si se satisfacen los criterios de evaluación establecidos.

La sintaxis propuesta en el modelo, se basa en la definición matemática de una estructura ampliamente conocida en el mundo de la computación y la informática. Por esta razón, las reglas y comportamiento de dicha estructura son transparentes al modelo.

Así mismo, los grafos son la estructura matemática más ampliamente usada para representar los elementos de la realidad. Y esto, sumado a que las relaciones de precedencia entre requerimientos han sido propuestas por diferentes enfoques de la ingeniería de requerimientos, permite concluir que la representación de los objetos del dominio (en este caso los requerimientos) es satisfactoria en cuanto a la evaluación de los criterios de validez de un modelo.

Por otra parte, la trazabilidad de la estructura del grafo y los correspondientes requerimientos que lo constituyen es intuitiva debido a que la relación de los elementos de grafo es inequívoca con los requerimientos que representan y sus correspondientes atributos.

7.1.3.2 Evaluación práctica. La evaluación práctica de la validez del sistema se orientó a investigar, cada una de las características del modelo, cumplía con los criterios de validez desde las diferentes

perspectivas de los actores involucrados en las pequeñas empresas desarrolladoras de software en Bogotá D.C.

Por otro lado, la evaluación práctica del modelo se llevó a cabo realizando una entrevista a algunas empresas que cumplían con las diferentes características que las acreditaban como pequeñas empresas desarrolladoras de software. En esta encuesta se diligenció con la ayuda de integrantes de las correspondientes empresas, llenando una lista de comprobación (**checklist**) definida por cada criterio y característica del modelo.

Los resultados de ese trabajo de campo indican que las características del modelo cumplen con los criterios de evaluación definidos, de manera que el modelo constituye un elemento a través del cual representar un conjunto de requerimientos funcionales (...Véase Anexo E...).

7.2 EVALUACIÓN DE LA HERRAMIENTA

Para la evaluación de la funcionalidad de la herramienta se procedieron a analizar los resultados de aplicarla a proyectos previamente desarrollados en las empresas. Esto, con el fin de comparar los resultados y características del desarrollo de un proyecto de software con el uso de la herramienta, frente a su ausencia durante el proceso. Para esto, se seleccionaron proyectos de software que contaran con las siguientes características:

- Cumplieran con los requisitos de precedencia del uso de la herramienta (...Véase sección 6.6.1...).
- El proyecto debería tener información relevante para la comparación en el desarrollo (tiempo de desarrollo de los requerimientos, finalización del proyecto, recursos, definición de requerimientos).
- Se hubiese realizado un análisis de requerimientos para estos procesos de manera que se comparara el enfoque de los procesos involucrados con las actividades para esta fase.

De esta manera, se realizó una comparación entre los resultados obtenidos a través de los procesos y herramientas establecidas en la empresa, y los que se podrían obtener a través de la utilización de la herramienta.

Además se realizó una lista para verificar si la funcionalidad de la herramienta permitía llevar a cabo las funcionalidades del modelo, y cual es la percepción de la funcionalidad de la herramienta.

7.3 CASO DE ESTUDIO

El caso más representativo a través del cual se pueden analizar los diferentes resultados obtenidos durante la evaluación del modelo y la herramienta, se basó en el análisis de un proyecto de software anteriormente desarrollado en la Empresa SWONE Ltda.

Para analizar los resultados, se tomó como entrada, la especificación de requerimientos de software realizada por la ingeniera de desarrollo Ana María Rodríguez Arévalo (...Véase Anexo E...). Esta especificación, fue analizada en detalle para determinar si cumplía con los requisitos mínimos para considerarse una entrada válida para la herramienta.

7.3.1 Entradas de la herramienta

Se tabularon los distintos requerimientos funcionales que formaban parte de la especificación, para traducirlos en la correspondiente entrada al sistema. El proyecto se relaciona con la creación de un sistema para el registro y facturación de un congreso de iglesias del Ministerio Cristiano Internacional MCI.

Tabla 8. Especificación de requerimientos de software proyecto MCI.

ID	Nombre en la Especificación	Modulo	Prioridad	Dificultad	Tiempo
1	Consultar/ Modificar/ Ingresar Evento	1	5	3	2
2	Consultar/ Modificar/ Ingresar Lugar de Evento	1	3	3	2
3	Registrar Ingreso Evento	2	5	5	2
4	Consultar/ Modificar/ Ingresar Iglesia	1	3	3	1
5	Consultar/ Modificar/ Ingresar Personas a Iglesia	1	5	4	4
6	Consultar/ Modificar/ Ingresar Inscripción a Evento	1	5	5	5
7	Consultar/Modificar/Generar/Imprimir Recibo Pago Evento	1	5	5	6
8	Consultar/Generar/Imprimir Escarapela Evento	1	5	5	2
9	Consultar/Modificar/AsignarseccionesLugar/delegación	1	5	5	4
10	Consultar/Modificar/RegistrarPagoEvento	1	5	5	3
11	ConfirmarInscripciónEvento	1	5	5	1
12	Consultar/Modificar/IngresarEntradas/SalidasEvento	1	5	4	4
13	Consultar/Modificar/Ingresardatosatablasmaestras	1	2	3	4
14	Generar/ImprimirReportes	3	4	4	4
15	Consultar/Modificar/IngresarUsuariosSistema	1	2	3	2
16	IdentificarTerminal	2	2	2	1

Tabla 9. Traducción de la especificación de requerimientos en la correspondiente entrada a la herramienta.

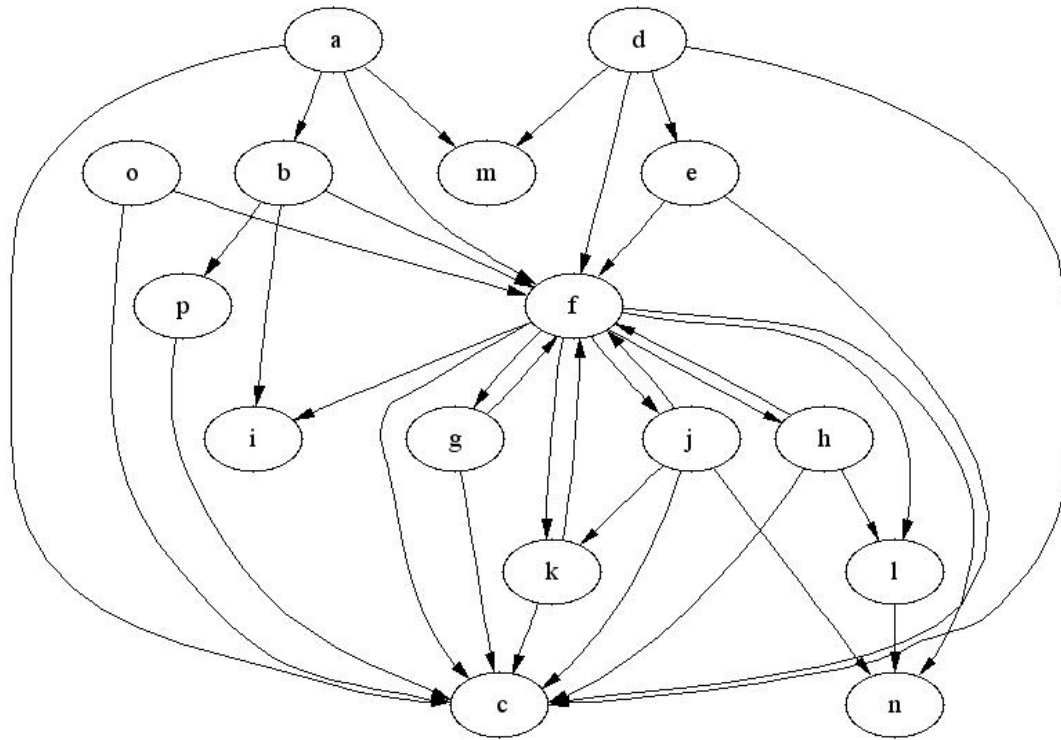
ID	ID Grafo	Modulo	Prioridad	Dificultad	Tiempo	Antecedentes
1	a	1	5	3	2	0
2	b	1	3	3	2	1
3	c	2	5	5	2	1,4,5,6,7,8,10,11,15,16
4	d	1	3	3	1	0
5	e	1	5	4	4	4
6	f	1	5	5	5	1,2,4,5,7,8,10,11,15
7	g	1	5	5	6	6
8	h	1	5	5	2	6
9	i	1	5	5	4	2,6
10	j	1	5	5	3	6
11	k	1	5	5	1	10,6
12	l	1	5	4	4	6,8
13	m	1	2	3	4	1,4
14	n	3	4	4	4	6,10,12
15	o	1	2	3	2	0
16	p	2	2	2	1	2

7.3.2 Operaciones y análisis realizados

Una vez ingresada la entrada (**input**, para ver el formato de entrada véase Anexo F) a la herramienta, se procedió a operar el modelo de manera que se pudiera obtener información relacionada al análisis de requerimientos del correspondiente proyecto.

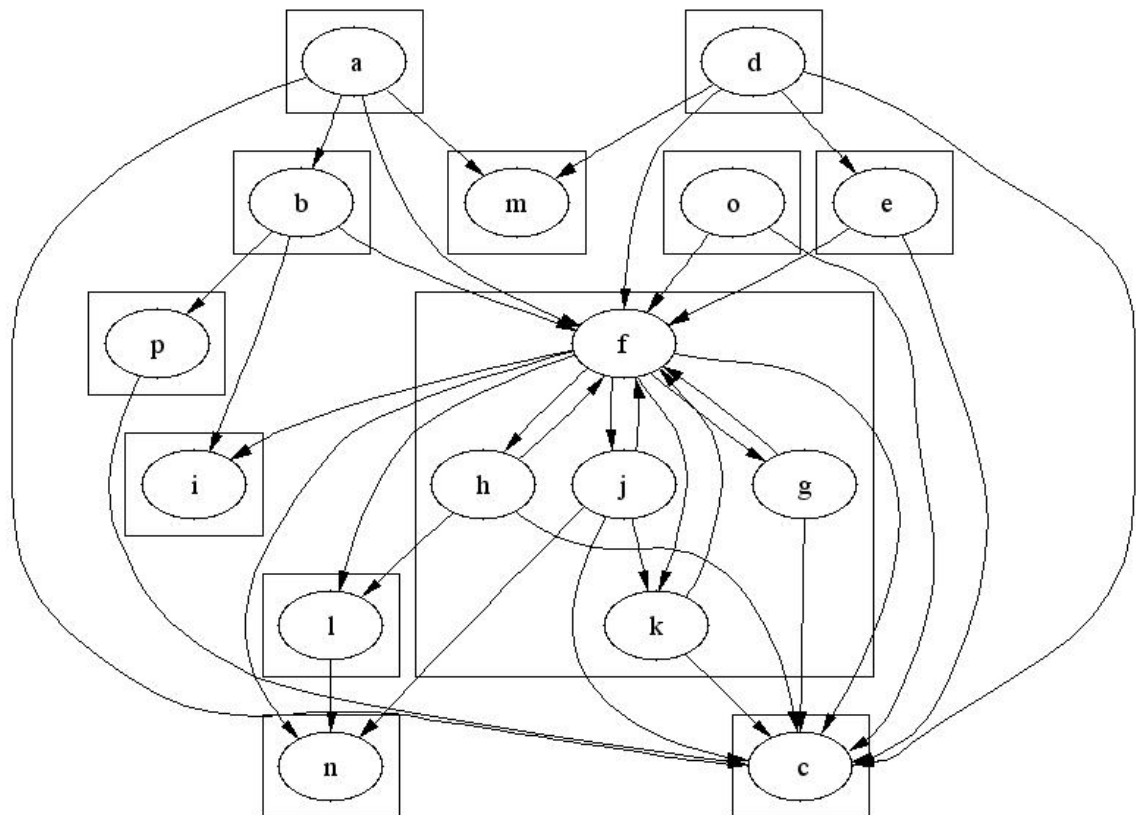
La herramienta aplicó los algoritmos descritos en el modelo (...Véase sección 6.4...), y las salidas arrojadas por la herramienta fueron las siguientes (...Véase Figura 19...):

Figura 19. Grafo Inicial de la especificación sin operar.



La Figura 19, corresponde a la traducción de la definición de requerimientos, en un grafo dirigido. Esta traducción se realizó sin llevar a cabo ninguna operación particular sobre el grafo. En esta figura se puede observar que el requerimiento identificado como *f* es un elemento que posee muchas interdependencias. Para facilitar la comprensión e identificación de este tipo de elementos, la herramienta generó el archivo necesario para construir la siguiente gráfica (...Véase Figura 20...).

Figura 20. Grafo diferenciado por sus componentes fuertemente conectados.

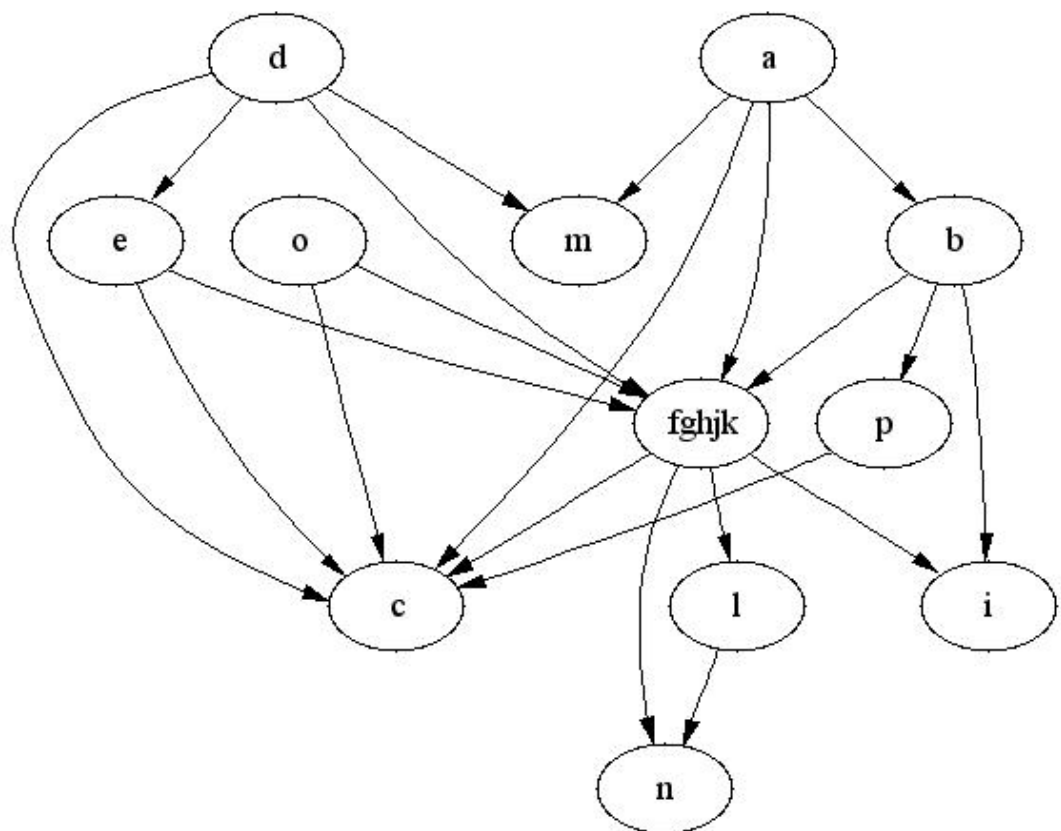


La figura 20, corresponde a la primera de las operaciones realizadas sobre el grafo: la detección de componentes fuertemente conectados. En esta oportunidad se puede diferenciar con claridad que elementos (requerimientos) del grafo son interdependientes, debido a que existe un recuadro que agrupa más de un requerimiento, indicando que ese conjunto de requerimientos constituye un elemento fuertemente conectado. En este caso, los requerimientos *f*, *g*, *h*, *j*; son

requerimientos que presentan una relación de interdependencia, y por tanto, un alto acoplamiento.

De igual manera, se aplicó una fusión de los requerimientos para facilitar la comprensión y manejo del grafo, de manera que se pudiese analizar más fácilmente las precedencias entre requerimientos (...Véase figura 21...).

Figura 21. Grafo reducido a través de la fusión de los requerimientos que conforman componentes fuertemente conectados.

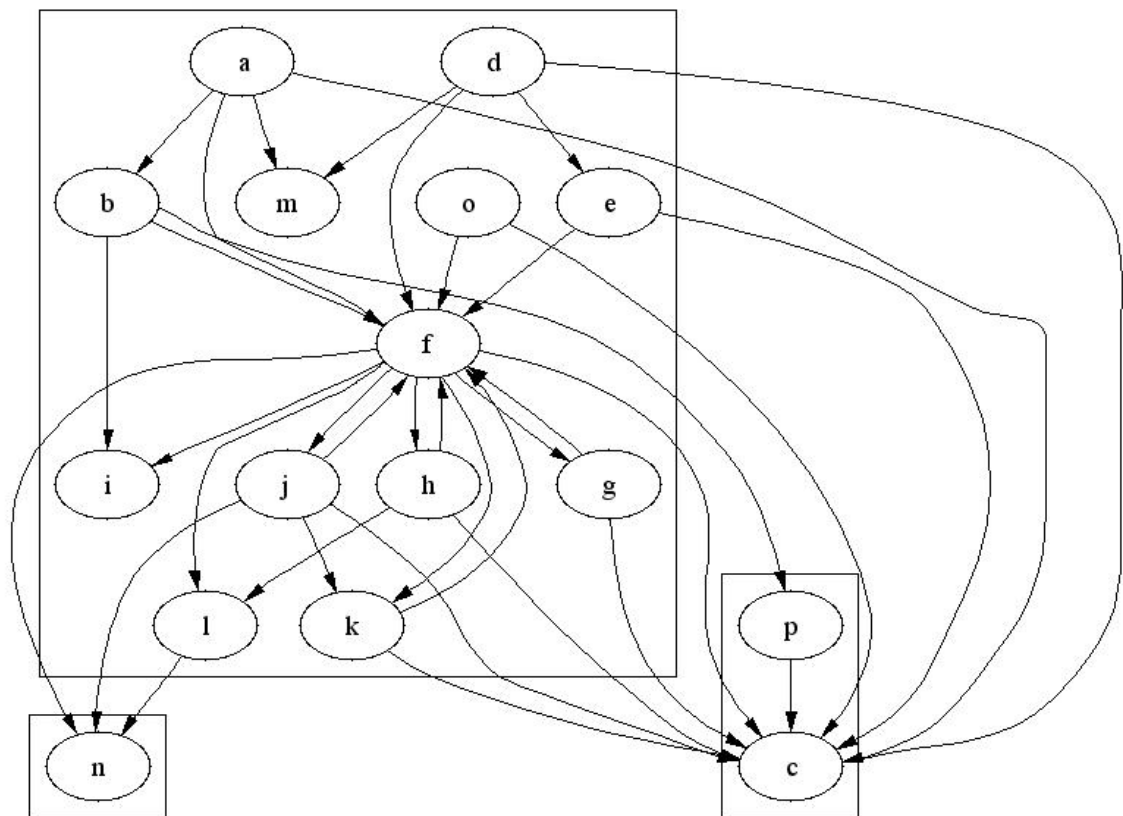


Esta definición de requerimientos es susceptible de ordenación, debido a que no se presentan ciclos dentro de la estructura de la misma. Además

el conjunto de requerimientos disminuye haciendo más manejable la definición de los mismos.

Así mismo, se analizó la relación de acoplamiento entre los módulos de software definidos para este proyecto (...Véase Figura 22...).

Figura 22. Grafo diferenciado de acuerdo a los módulos de software a los cuales pertenece cada requerimiento.



En la figura 22, se puede analizar la dependencia y acoplamiento de los módulos de software, debido a que cada uno de los recuadros que agrupan requerimientos corresponde a un módulo de software.

7.3.3 *Análisis del caso de estudio*

El aplicar los resultados obtenidos a través de la herramienta, permitió el obtener diferentes conclusiones alrededor del análisis de requerimientos realizado para el proyecto MCI.

- La definición de requerimientos previamente realizada contaba con un elemento fuertemente conectado, que correspondía al alto acoplamiento entre los requerimientos que lo componen.
- La definición de los módulos fue correcta debido a que no se presenta interdependencia entre los mismos.
- El análisis de requerimientos asistido a través de la herramienta, permitió visualizar elementos y características globales que pudieron haber sido de gran importancia en el desarrollo del conjunto de requerimientos, como la dependencia entre requerimientos interconectados.
- Este análisis fue sencillo y más rápido. Se observó que el análisis de requerimientos realizado para el proyecto, tardó 2 días. El análisis realizado a través de la herramienta, tardo tres horas⁹⁹.

7.4 APLICABILIDAD DEL MODELO Y ANÁLISIS DE RESULTADOS

El análisis de los resultados arrojados por el estudio de campo, permitió establecer algunas conclusiones. Este modelo constituye una representación matemática de la definición de requerimientos que se puede dar para un producto de software. Sin embargo, para que este

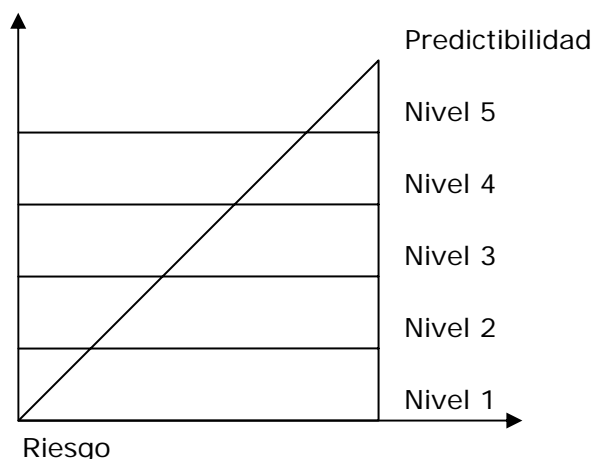
⁹⁹Para más información sobre los estudios véase el anexo F.

modelo no persista sólo como una consigna matemática de lo que se puede realizar con un conjunto de requerimientos, se propone que se utilice para las siguientes operaciones al interior de la pequeña empresa:

7.4.1 El análisis de riesgos asociados a un requerimiento como factor para el incremento de la calidad en desarrollos de software

Analizar los riesgos asociados a una definición de requerimientos, es un elemento muy útil para la mejora de la calidad y el incremento de la predictibilidad en los procesos relacionados con la producción de software.

Figura 23. Relación de las variables riesgo y predictibilidad en el desarrollo de software.

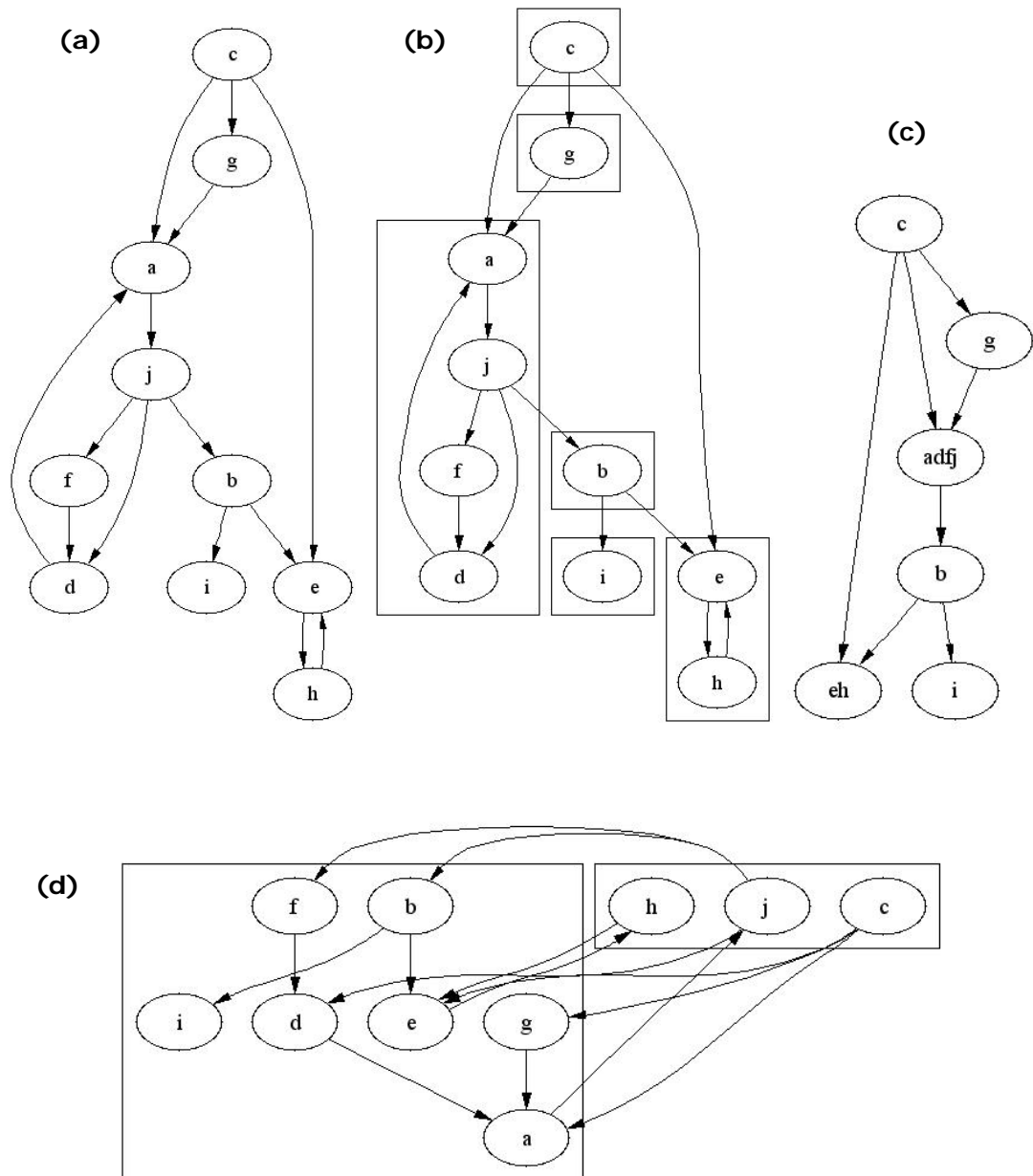


Si se analiza la figura anterior, se puede observar que la magnitud del riesgo y la predictibilidad se afectan de manera inversamente

proporcional. Esto significa que, a medida que se mitiguen los riesgos, se aumenta la predictibilidad. La predictibilidad es una característica del desarrollo de software que se relaciona directamente con la calidad del mismo. Así mismo, se puede observar que la predictibilidad se relaciona con el nivel de madurez (definido en CMM) del desarrollo de software de una empresa. A medida que se escala en la pirámide es necesario ir disminuyendo las probabilidades y factores de riesgo asociados a un desarrollo de software.

7.4.1.1 ¿Cómo se pueden analizar los riesgos asociados a un requerimiento a través de la herramienta? El análisis del riesgo asociado a un requerimiento, se realiza a través del análisis de dos factores de la definición de requerimientos: los atributos asociados a un requerimiento, y la estructura del grafo correspondiente a dicha definición.

Figura 24. Diferentes Estructuras que se conforman de una definición de requerimientos, traducida al modelo y sus correspondientes operaciones.



Cada una de las figuras anteriores muestra un aspecto particular de la definición de requerimientos. La figura 24 –a, representa una definición

de requerimientos traducida al modelo sin operar. En este caso el análisis que se puede derivar de esta gráfica es mínimo. En la figura 24-b, se pueden observar diferentes elementos interconectados. Esta operación se realiza a través de la aplicación del algoritmo (**Strongly Connected Components**). En este análisis, se identifican los componentes fuertemente conectados del grafo, que constituyen elementos de riesgo dentro de una definición de requerimientos.

Un componente fuertemente conectado es claramente un elemento de alto riesgo dentro de una definición de requerimientos de software debido a que representa alguno de los siguientes casos:

- Existen diferentes participantes del análisis de requerimientos, y se presenta un conflicto entre las diferentes perspectivas, de manera que se presenta interdependencia entre requerimientos.
- Ese componente, realmente constituye un conjunto de requerimientos demasiado relacionados como para analizarlos por separado.
- Constituye un error en la definición de los requerimientos de parte del analista de los mismos.

El identificar y focalizar donde se encuentran estos componentes, sirve para tomar decisiones en la definición de requerimientos de software, o en el debido caso, para detectar errores y corregirlos.

La figura 24 -c, representa al conjunto de requerimientos, después de fusionar o compactar la definición de los elementos fuertemente

conectados. Esto significa que se puede facilitar la comprensión de una definición de requerimientos si se aíslan y analizan sus componentes fuertemente conectados como elementos unitarios de la definición. Esto no conlleva a un error, debido a que la fusión no cambia el comportamiento de un conjunto de requerimientos. Esto se debe a que las entradas y las salidas de un elemento fuertemente conectado se conservan como en los elementos individuales, es decir su comportamiento como grupo de requerimientos representa a todos los requerimientos que agrupa.

La figura 24 –d, representa el análisis de una definición de requerimientos a través de la diferenciación de los módulos a los cuales pertenecen. Esto sirve para determinar el acoplamiento entre dos o más módulos de software. Este se puede analizar a través del número de enlaces que intercambian los requerimientos de cada uno de los respectivos módulos. Si la relación de los enlaces de los requerimientos de un módulo es sólo de salida o entrada (de un modulo A sólo salen o entran enlaces de un módulo B), se puede discriminar que la definición de lo módulos tiene un riesgo bajo, debido a que los requerimientos fueron separados de manera que cada módulo sólo precede o le prosigue a otro módulo. Otros casos menos favorables se pueden presentar cuando la relación de entradas y salidas es mutua (de un módulo A pueden salir y entrar enlaces de un módulo B y viceversa). Esto significa, que al igual que en la relación de interdependencia entre requerimientos, los módulos son interdependientes entre sí, y puede acarrear problemas en el subsiguiente desarrollo de los mismos. Sin embargo, el mayor caso de acoplamiento, se da, cuando un componente

fuertemente conectado se encuentra distribuido entre dos o más módulos. Esto constituye un factor de riesgo muy elevado debido al alto acoplamiento en el desarrollo de los módulos que dependen entre sí.

Por otro lado, el análisis de riesgos se puede llevar a cabo a través de los atributos de un requerimiento. Es así que se pueden clasificar los requerimientos por orden de riesgo así: se evalúan los atributos de un requerimiento a través de una función de riesgo, y se determina cuales son los requerimientos más riesgosos.

Tabla 10 Datos correspondientes a la definición de requerimientos de la figura 24.

ID	Nombre	Modulo	Prioridad	Dificultad	Tiempo	Antecedentes
1	a	1	1	3	10	4,3,7
2	b	1	2	2	9	10
3	c	2	3	1	8	0
4	d	1	1	3	7	6,1
5	e	1	2	2	6	3,2,8
6	f	1	3	1	5	10
7	g	1	1	3	4	3
8	h	2	2	2	5	5
9	i	1	3	1	6	2
10	j	2	1	3	7	1

Al organizar los requerimientos en orden de prioridad evaluando los diferentes atributos que poseen, se obtiene lo siguiente:

Tabla 11. Orden por factores de riesgo de los requerimientos de la tabla 10.

ID	Nombre	Modulo	Prioridad	Dificultad	Tiempo	Antecedentes
1	c	2	3	1	8	0
2	f	1	3	1	5	10
3	i	1	3	1	6	2
4	b	1	2	2	9	10
5	e	1	2	2	6	3,2,8
6	h	2	2	2	5	5
7	j	2	1	3	7	1
8	a	1	1	3	10	4,3,7
9	d	1	1	3	7	6,1
10	g	1	1	3	4	3

7.4.2 ¿Cómo aplica a las empresas?

Son varias las utilidades que se le pueden atribuir a este tipo de análisis. En primer lugar, clasificar los requerimientos por módulo y riesgo permite entablar elementos de criterio para determinar como distribuir el trabajo a los desarrolladores del software. Esto aplica a que un director de proyecto puede decidir asignar el requerimiento más riesgoso a su mejor y más experimentado desarrollador.

Así mismo, la definición de módulos de software puede ser orientada a través del análisis de dependencias de los mismos, lo cual traerá beneficios en etapas posteriores, debido a la correcta separación de la funcionalidad de un módulo. Este puede constituir un primer paso para la correcta definición de interfaces de funcionalidad y el desarrollo basado en componentes (**Component Based Software Development**).

Así mismo, se puede verificar que no se constituyan elementos fuertemente conectados en diferentes módulos, es decir, que dado el caso de persistir un elemento fuertemente conectado en una definición de requerimientos, este debe hacer parte de un único módulo para fomentar un bajo acoplamiento en el desarrollo.

En el caso particular de la pequeña empresa desarrolladora de software bogotana, el uso de este tipo de modelos acarrearía beneficios en la calidad del desarrollo de software que realicen. Esto permitiría mejorar a las empresas que se encuentran en niveles bajos de calidad, y no están en capacidad de competir en las condiciones que demanda el mercado internacional. Este tipo de elementos constituye un primer paso para la mejora de los procesos productivos de software al interior de estas organizaciones.

8. RESULTADOS DE LA INVESTIGACIÓN

Los principales resultados que se obtuvieron a través de esta investigación son:

- Se elaboró un prototipo funcional evolutivo de herramienta de software para análisis de requerimientos.
- Se identificó un modelo de proceso para la fase de recolección orientado a la flexibilidad y calidad del desarrollo de software, para facilitar la recolección de los datos a manejar en un proceso de análisis de requerimientos. Este proceso es el apropiado para recolectar la información necesaria para operar el prototipo de la herramienta.
- También se identificó y probó un modelo de proceso para el análisis de requerimientos dentro de proyectos de software, orientado al aumento de la calidad y la disminución de riesgos.
- Se elaboró un modelo matemático y computacional para el análisis de requerimientos, basado en grafos dirigidos. Así mismo se definieron operaciones y algoritmos para llevar a cabo actividades detección de errores, análisis de riesgos, y estimación de recursos. Este modelo se utilizó como base para el desarrollo del prototipo de herramienta.
- Se realizó un análisis de las fases de recolección y análisis de requerimientos al interior de la pequeña empresa desarrolladora de software en Bogotá D.C. Este análisis comprendió la definición de escenarios teóricos de estudio,

estudio de campo en las empresas y la ejecución de casos de estudio.

- Los casos de estudio mostraron que el uso del prototipo funcional es aplicable a los procesos de análisis de requerimientos en proyectos de software. El uso de la herramienta en casos de estudio, representó una disminución en el tiempo estimado para la realización del proceso de análisis de riesgos, facilitándolo a través de representaciones visuales, y mejorando la comprensión de la definición de requerimientos.
- La calidad de los procesos productivos de desarrollo de software se puede aumentar a través de la disminución de riesgos en la etapa de análisis de requerimientos, y esto puede constituir un elemento de apoyo para las pequeñas empresas desarrolladoras de software bogotanas.
- Como investigador en este proyecto de grado, profundicé mis conocimientos en el área de la ingeniería de requerimientos.

9. CONCLUSIONES

Las conclusiones que se pueden obtener alrededor de esta investigación, se pueden diferenciar en dos aspectos principales: el análisis realizado a la pequeña empresa desarrolladora de software, y el desarrollo de un modelo conceptual y posterior elaboración de un prototipo evolutivo para la realización de las actividades relacionadas con el análisis de requerimientos.

Dentro de los aspectos más importantes que se encontraron al estudiar las pequeñas empresas desarrolladoras de software se encuentra que:

- Los procesos de desarrollo de software en estas empresas, y en concreto, los relacionados con la ingeniería de requerimientos están sujetos a muchas mejoras: estas empresas enfrentan el reto de acreditar estos procesos a través de certificaciones como ISO 9001, CMM, CMMI, etc.
- Es necesario que estas empresas mejoren su proceso de ingeniería de requerimientos, si desean ser competitivas con el desarrollo de software a nivel internacional: las empresas del mercado exterior exigen altos estándares de calidad que la mayoría de las pequeñas empresas no puede satisfacer en la actualidad.
- Las actividades de ingeniería de requerimientos dentro de estas empresas se llevan a cabo a través de diferentes herramientas y estrategias: los enfoques que se orienten a solucionar problemáticas de este proceso deben ser abiertos y flexibles de manera que no se restrinjan a herramientas o estrategias particulares.

- Cualquier elemento que facilite la mejora de la calidad de los procesos de desarrollo de software al interior de estas empresas permitirá a las mismas tener mayores posibilidades de éxito en el mercado nacional como internacional: la herramienta de software desarrollada en esta investigación ofrece una ventaja competitiva para afrontar las necesidades de calidad de estos mercados.

Así mismo, se pueden obtener diferentes conclusiones relacionadas con el desarrollo del modelo computacional y la herramienta. Entre éstas se encuentran:

- El modelo computacional se constituyó a través de Un Grafo de RF basado en precedencias: esta estrategia brinda formalidad, rigor y facilidades computacionales al análisis de requerimientos.
- La utilidad de este grafo es extensible a otros procesos: puede ser utilizado en actividades de otras fases como la planeación y administración de requerimientos.
- Este modelo propone el uso de algoritmos para análisis de grafos, con el fin de facilitar el uso de actividades propias del análisis de requerimientos: detección de incongruencias, análisis de riesgos y estimación de recursos.
- Una herramienta de software ofrece las ventajas asociadas a su implementación computacional: facilita el manejo de grandes volúmenes de datos, el control de los elementos, la velocidad y tiempo de análisis y procesamiento de la información.
- El modelo permite gestionar las necesidades de análisis de riesgos asociados a los requerimientos en etapas tempranas, de los proyectos de software en forma estructurada: cada actividad del

análisis de requerimientos consiste de una serie de pasos organizados y bien definidos.

- El modelo es abierto, adaptable y modular: la estructura del modelo no se restringe a ninguna herramienta, o estrategia para desarrollo de actividades del análisis de requerimientos.
- Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: La IR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
- Mejora la calidad del software: La calidad en el software tiene que ver con cumplir un conjunto de requerimientos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.), y el modelo se orienta mejorar estos aspectos.
- Mejora la comunicación entre equipos: la representación gráfica de los requerimientos a través de un grafo permite que las características de los mismos sean comprendidas por diferentes actores del proceso de desarrollo de software.

TRABAJOS FUTUROS

La investigación se realizó de manera aislada para las fases de recolección y análisis de requerimientos. Por esto, existen interrogantes alrededor de su aplicabilidad y extensibilidad a las demás fases de la ingeniería de requerimientos. Así mismo, el enfoque de desarrollo sobre el cual se orienta el modelo, es un enfoque tradicional, de manera que se plantea el reto de extrapolar sus resultados a otros ámbitos de desarrollo de software, menos tradicionales. Algunas de las posibles extensiones de esta investigación las constituyen:

- Definir la integración de este paradigma de análisis de requerimientos con las demás fases de la ingeniería de requerimientos.
- Construir una herramienta que realice más fases de la ingeniería de requerimientos, como la administración.
- Definir una extensión de este modelo de análisis de requerimientos en proyectos de software no convencionales, como el desarrollo de agentes.
- Definir una extensión del modelo al ámbito de los requerimientos no funcionales.
- Elaborar una representación o lenguaje para la integración con otras herramientas, como la extensión o definición de una representación XML bajo el estándar de la OMT para compartir información.
- Realizar Plug-Ins para la utilización del modelo en herramientas especializadas en otras fases, existentes en el mercado.
- Cómo llevar a cabo las actividades del análisis de requerimientos que la herramienta no satisface.

BIBLIOGRAFÍA

ANTÓN, A. Goal Based Requirements Analysis. College of Computing, Georgia Institute of technology. Atlanta, Georgia; U.S.A. 1996.

BARRERA FUENTES, William Eduardo. Ingeniería de requerimientos desde una perspectiva social. Bogotá, 2002. Trabajo de Grado (Ingeniero de Sistemas). Universidad de los Andes. Facultad de Ingeniería. Área de Ingeniería de Software.

BOEHM, Barry. Software Engineering Economics. New Jersey: Prentice Hall, 1981.

CAMARÁ DE COMERCIO DE BOGOTÁ. Estadísticas generales sobre las empresas colombianas.

CARVALHO, Marcia y **ABDELOUAHAB**, Zair. Un metodo para elicitação e moderação de requisitos baseado em objetivos [online].

COCKBURN, Alistair. Basic Use Case Template [online]. Disponible en Internet:

< <http://members.aol.com/acockburn> >

CORMEN, T., y **LEISERNON**, C. Introduction to Algorithms. 2da edición. Mc Graw Hill. MIT Press 2002.

CORTÉS B., Gloria C. Los retos actuales para nuestra industria de software En: Sistemas: El entorno colombiano en procesos modernos de desarrollo de software, N° 86, agosto-octubre de 2003 [consultada mayo de 2005]. Disponible en Internet:
<<http://sitio.acis.org.co/Paginas/publicaciones/editorial86.html> >

CYSNEIROS, L. M. Requisitos Não Funcionais: Da Elicitação ao Modelo Conceitual. Tesis Ph. D., Departamento de Informática, Pontificia Universidad Católica de Río de Janeiro, Río de Janeiro, Brasil, 2001.

DÁVILA, Nicolás Davyt. Ingeniería de requerimientos: Una guía para extraer, analizar, especificar y validar los requerimientos de un proyecto. Artículo Técnico (Licenciado en Análisis de Sistemas). Universidad ORT, Uruguay. Facultad de Ingeniería. Área de Sistemas.

DURÁN TORO, A. *et al.* A requirements elicitation approach based in templates and patterns [online]. Disponible en Internet:
<http://wer.inf.puc-rio.br/WERpapers/artigos/artigos_WER99/toro.pdf>

FAULK, Stuart R. Software requirements: A tutorial. En: **THAYER**, Richard y **DORFAM**, Merlin. Software Requirements Engineering. 2 *ed.* Los Alamitos, California: IEEE Computer Science Press, 2000. p. 158 - 179.

FEDERACIÓN COLOMBIANA DE LA INDUSTRIA DEL SOFTWARE Y TECNOLOGÍAS INFORMÁTICAS RELACIONADAS, Fedesoft. Página

oficial [consultada enero de 2004]. Disponible en Internet:
<<http://www.fedesoft.com>>

GOGUEN, Joseph A. y **LINDE**, Charlotte. Techniques for requirements elicitation. En: **THAYER**, Richard y **DORFAM**, Merlin. Software Requirements Engineering. 2 ed. Los Alamitos, California: IEEE Computer Science Press, 2000. p. 140-152.

HAY, David C. Requirements analysis: from business views to architecture. Upper Saddle River, New Jersey: Prentice Hall, 2003

INSTITUTE FOR ELECTRONICS AND ELECTRICAL ENGINEERS.
Glosario estándar de la terminología de la ingeniería de software estándar
610.12-1990. s.l.: La institución, 1997.

KONTOYA, Gerald y **SOMMERVILLE**, Ian. Requirements engineering: Processes and techniques. Chinchester, Inglaterra: John Wiley & Sons Ltd, 1998.

KUDIKYALA, U. K. Reducing misunderstanding of software requirements by conceptualization of mental models using pathfinder networks, Disertación de Ph. D., Departamento de Ciencias de la Computación, Mississippi State University, Mississippi, U.S.A., Agosto de 2004.

LU, X. Using pathfinder networks to analyze and categorize software requirements. Reporte Técnico MSU – 030922, Requisito para la obtención del título de Maestría en Ciencias de la Computación, Departamento de Ciencias de la Computación, Mississippi State University, Mississippi, U.S.A., Agosto de 2002.

MCEWEN, Scott. Requirements: An Introduction. The Rational Edge [online]. Abril 2, 2004 [consultada mayo de 2005]. Disponible en Internet:

<<http://www-106.ibm.com/developerworks/rational/library/4166.html>>

MEAD, Nancy R. Requirements management and requirements engineering: You can't have one without the other. En: Cutter IT Journal. Vol. 13. No. 5 (may 2000).

MICROSOFT DEVELOPERS NETWORK (MSDN). Sitio Web para estudiantes Latinoamérica. Disponible en Internet:

<<http://www.microsoft.com/spanish/MSDN/estudiantes/ingsoft/ingenieria/especif.asp>>

OBERG, Roger; **PROBASCO**, Leslee y **ERICSSON**, Maria. **RATIONAL SOFTWARE**. Applying requirements management with use cases [online]. Rational Software Corporation, 2003 [consultada mayo de 2005]. Disponible en Internet:

<<http://www.uml.org.cn/RequirementProject/pdf/apprmuc.pdf>>

PAETSCH, Fachhochschule; **EBERLEIN**, Armin and **MAURER**, Frank. Requirements engineering and agile software development [online]. Disponible en Internet: <<http://citeseer.ist.psu.edu>>

PUYANA SILVA, David Guillermo. La Problemática De Las PYMES en Colombia: Internacionalizarse o Morir [online]. Centro de investigación de de la Escuela de finanzas y Comercio Exterior, Universidad Sergio Arboleda. Agosto 2002 [consultada mayo de 2005]. Disponible en Internet: <<http://www.usergioarboleda.edu.co/pymes/noticia1.htm>>

RATIONAL CORPORATION IBM. Página de Rational Unified Process [consultada enero de 2004] Disponible en Internet: <<http://www-306.ibm.com/software/awdtools/rup/>>

REAL ACADEMIA DE LA LENGUA ESPAÑOLA (RAE). Definición de la Palabra "modelo". Disponible en Internet: <<http://www.rae.es>>

ROJAS MARIN, Obdulio. Herramienta para la administración de los requerimientos en los proyectos de ingeniería de software y procesos productivos. Bogotá, 2000. Trabajo de Grado (Ingeniero de Sistemas). Pontificia Universidad Javeriana. Facultad de Ingeniería. Área de Ingeniería de Software

ROBERTSON, Suzanne y **ROBERTSON**, James. Mastering the requirements process. Londres: Addison - Wesley, 1999.

ROBERTSON, Suzanne. Requirements Driven Software Planning.

RODRÍGUEZ, Astrid Genoveva. La realidad de la PyME Colombiana: Desafío para Colombia. Bogotá: Fundes, 2003. p. 1.

RUMBAUGH, James. Getting started: Using use cases to capture requirements. En: **THAYER**, Richard y **DORFAM**, Merlin. Software Requirements Engineering. 2 ed. Los Alamitos, California: IEEE Computer Science Press, 2000. p. 153 - 157.

SAWYER, Peter y **KONTOYA**, Gerald. Software requirements. En: Software Engineering Book Of Knowledge (SWEBOK), capítulo 2. p.18. Disponible en Internet:
<<http://www.swebok.org>>

SOFTWARE ENGINEERING INSTITUTE (SEI). Capability Maturity Model [online]. Disponible en Internet:
<<http://www.sei.cmu.edu/cmm>>

SOMMERVILLE, Ian y **SAWYER**, Peter. Requirements engineering: A good practice guide. 3 ed. Chinchester, Inglaterra: John Wiley & Sons Ltd., 2000.

SOMMERVILLE, I. *An Integrated Approach to Dependability Requirements Engineering*. Proc. 11th Safety-Critical Systems Symposium, Bristol. 3-15, Springer. 2003.

SOMERVILLE, Ian. Ingeniería de software. 7 ed. México: Addison – Wesley, 2004.

THAYER, Richard y **DORFAM**, Merlin. Software Requirements Engineering. 2 ed. Los Alamitos, California: IEEE Computer Science Press, 2000. p. 1.

THE ZACHMAN INSTITUTE FOR FRAMEWORK ADVANCEMENT, ZIFA. Página oficial [consultada enero de 2004]. Disponible en Internet: <<http://www.zifa.com>>

TORRES, M. An automated tool for software requirement refinement and pathfinder networks generation. Reporte Técnico MSU – 030922. Requisito para la obtención del título de Maestría en Ciencias de la Computación, Departamento de Ciencias de la Computación, Mississippi State University, Mississippi, U.S.A., Agosto de 2003.

VALDES CÁRDENAS, Luis Eduardo. Situación actual de la informática en Colombia. Bogotá: Centro de Apoyo a las Tecnologías Informáticas (CATI), abril de 2004.

WIEGERS, Karl. Software Requirements. 2 ed. Washington: Microsoft Press, 2003

WIEGERS, Karl. When telepathy won't do: Requirements engineering key practices. En: Cutter IT Journal. Vol. 13. No. 5 (may 2000). Disponible en Internet:

<<http://www.processimpact.com/articles/telepathy.pdf> >

WIEGERS, Karl. First things first: Prioritizing requirements. En:
Software Development Magazine [online], September 1999. Disponible
en Internet: <www.processimpact.com/articles/prioritizing.pdf>