



**UNIVERSIDAD NACIONAL DE LA
MATANZA
ESCUELA DE POSGRADO**

Tesis de Maestría en Informática

**ESTRATEGIA DE REQUISITOS ADAPTABLE
SEGÚN FACTORES DE SITUACIÓN**

Autora: Ing. Viviana Alejandra Ledesma

Directora: Dra. Graciela D.S. Hadad

Codirector: Ing. Jorge H. Doorn

Buenos Aires, Marzo de 2019

Índice

Índice	i
Resumen	iv
Lista de Figuras	v
Lista de Tablas	vii
Capítulo 1	1
Introducción	1
1.1 Motivación	1
1.2 Alcances del Trabajo	2
1.3 Hipótesis	2
1.4 Objetivos	3
1.5 Organización del Documento	3
Capítulo 2	5
Marco Teórico	5
2.1 Introducción	5
2.2 Visiones del Proceso de Ingeniería de Requisitos	6
2.3 Estrategia de Ingeniería de Requisitos basada en Modelos de Lenguaje Natural	12
2.3.1 Comprender el Vocabulario	15
2.3.2 Comprender el Universo de Discurso Actual	16
2.3.3 Definir el Contexto del Software	17
2.3.4 Explicitar Requisitos	20
2.4 Variabilidad de Procesos	21
2.5 Ingeniería de Métodos Situacional	27
2.5.1 Factores Situacionales	31
2.6 Adaptabilidad en la Ingeniería de Requisitos	32
Capítulo 3	36
Factores Situacionales Comprometidos en la Estrategia de Requisitos	36
3.1 Determinación del Conjunto de Factores Situacionales	36
3.1.1 Análisis de los Factores	38
3.1.2 Definición de los Factores Situacionales Seleccionados	42
3.2 Dependencias entre Factores Situacionales	46
Capítulo 4	51

Mecanismo de Adaptación Situacional de la Estrategia de Requisitos	51
4.1 Introducción.....	51
4.2 El Proceso Base y sus Puntos de Variación.....	52
4.3 Variantes de Adaptación	56
4.3.1 Variantes del Proceso Base	58
4.3.2 Variantes al Crear el Léxico Extendido del Lenguaje	67
4.3.3 Variantes al Construir Escenarios Actuales.....	72
4.3.4 Variantes al Construir Escenarios Futuros	77
4.3.5 Variantes del Subproceso Describir Escenarios Futuros.....	82
4.3.5.1 Describir Escenarios Futuros con Enfoque Procedural	83
4.3.5.2 Describir Escenarios Futuros con Enfoque Dirigido por Objetivos	85
4.3.5.3 Describir Escenarios Futuros con Enfoque Híbrido	88
4.3.5.4 Describir Escenarios Futuros con Enfoque Top-Down.....	90
4.3.5.5 Describir Escenarios Futuros con Enfoque Basado en el Léxico	93
4.3.6 Variantes al Crear Léxico Extendido del Lenguaje del Sistema	94
4.3.7 Variantes al Especificar Requisitos	96
Capítulo 5	100
Adaptación Dinámica del Proceso de Requisitos	100
5.1 Estimación y Evolución de los Factores Situacionales.....	100
5.2 Estudio de la Evolución de los Factores Situacionales	102
5.3 Proceso de Adaptación Dinámica	108
Capítulo 6	116
Validación de la Propuesta de Adaptación	116
6.1 Método de Validación.....	116
6.2 Descripción del Estudio de Caso.....	116
6.3 Análisis de los Resultados.....	120
6.4 Conclusiones de la Validación.....	125
Capítulo 7	127
Conclusiones y Futuras Líneas de Investigación.....	127
7.1 Conclusiones.....	127
7.2 Resultados Publicados de la Investigación	129
7.3 Líneas Futuras de Investigación.....	130
Anexo A.....	132

Análisis de Factores Situacionales en la Literatura	132
Anexo B.....	146
Formulario para Evaluar Factores Situacionales.....	146
Anexo C.....	150
Información Adicional del Estudio de Caso sobre Evolución de Factores	150
Anexo D.....	152
Información Adicional del Estudio de Caso sobre Variantes de Adaptación	152
Referencias	156

Resumen

Esta tesis presenta una propuesta para mejorar la aplicación de un proceso de Ingeniería de Requisitos basado en modelos de lenguaje natural. La mejora tiene que ver con proveer mecanismos claros que sirvan de guía a los ingenieros de requisitos para ajustar dicho proceso de acuerdo con el contexto particular del proyecto de software en el cual deba ser aplicado. Entonces, dependiendo de las circunstancias imperantes en el proyecto, es posible que algunas actividades se puedan eliminar, simplificar o reemplazar, incluso puede ser necesario seguir caminos diferentes agregando otras actividades. Para ello se define un conjunto de indicadores basados en factores situacionales que permiten describir dicho contexto.

Adicionalmente se proponen, como una solución mejorada, instrumentos para realizar dicho ajuste. Se intenta abordar la adaptación del proceso de Ingeniería de Requisitos desde un punto de vista dinámico; este enfoque es necesario debido a que algunos de los factores situacionales no se conocen con exactitud desde el comienzo mismo del proyecto, incluso puede suceder que otros factores cambien en la medida que se avanza con el proyecto. Por lo tanto, el proceso de adaptación establece en qué momento se deben reevaluar los factores, revisando y reajustando el proceso de construcción de requisitos en base a reglas predefinidas.

Lista de Figuras

Figura 2-1. Jerarquía del dominio de la IR. Fuente: Wiegers & Beatty, 2013.....	7
Figura 2-2. Proceso de IR. Adaptada de Hadad, 2008.....	8
Figura 2-3. Estrategia de IR basada en modelos de lenguaje natural. Adaptada de Hadad, 2008.....	13
Figura 2-4. Proceso de creación de LEL. Adaptada de Hadad, 2008.....	15
Figura 2-5. Proceso de construcción de EA. Adaptada de Hadad, 2008.....	17
Figura 2-6. Proceso de construcción de EF. Adaptada de Hadad, 2008.....	18
Figura 2-7. Proceso de construcción de ERS. Adaptada de Hadad, 2008.....	20
Figura 3-1. Matriz de dependencias entre los factores situacionales.....	48
Figura 4-1. Obtención de un proceso adaptado a una situación.....	51
Figura 4-2. Proceso base de IR.....	53
Figura 4-3. Factores situacionales y puntos de variación.....	55
Figura 4-4. Variantes del proceso base de IR.....	58
Figura 4-5. Variantes al Crear el Léxico Extendido del Lenguaje.....	67
Figura 4-6. Variantes al Construir Escenarios Actuales.....	72
Figura 4-7. Variantes al Construir Escenarios Futuros.....	77
Figura 4-8. Subproceso Describir Escenarios Futuros con Enfoque Procedural....	83
Figura 4-9. Subproceso Describir Escenarios Futuros con Enfoque Dirigido por Objetivos.....	85
Figura 4-10. Subproceso Describir Escenarios Futuros con Enfoque Híbrido.....	88
Figura 4-11. Subproceso Describir Escenarios Futuros con Enfoque Top-Down...	91
Figura 4-12. Subproceso Describir Escenarios Futuros con Enfoque Basado en el Léxico.....	93
Figura 4-13. Variantes al Crear Léxico Extendido del Lenguaje del Sistema.....	94
Figura 4-14. Variantes al Especificar Requisitos.....	97
Figura 5-1. Nivel de confianza en la valoración inicial y final.....	104
Figura 5-2. Análisis de la evolución del nivel de confianza para el factor situacional Complejidad del contexto.....	105
Figura 5-3. Proceso de adaptación del proceso de IR en función de factores situacionales.....	108
Figura 5-4. Ejemplo de adaptación del proceso de IR para la situación inicial de un proyecto.....	113

Figura 5-5. Ejemplo de readaptación del proceso de IR en el Punto de Variación 3 para la nueva situación..... 115

Lista de Tablas

Tabla 2-1. Actividades de la IR. Fuente: Hadad, 2017	8
Tabla 2-2. Enfoques para la construcción de métodos.....	29
Tabla 3-1. Factores situacionales relacionados al contexto de aplicación	37
Tabla 3-2. Factores situacionales relacionados al proyecto de software.....	38
Tabla 3-3. Análisis de factores asociados al contexto de aplicación	41
Tabla 3-4. Análisis de factores asociados al proyecto.....	42
Tabla 3-5. Valoración de los factores situacionales.....	46
Tabla 3-6. Reglas de dependencias entre factores situacionales	49
Tabla 4-1. Elementos básicos para representar la adaptación del proceso de IR..	57
Tabla 5-1. Tipo de evolución de los factores situacionales	101
Tabla 6-1. Detalle de la situación para los casos bajo estudio	117
Tabla 6-2. Asignación de casos y variantes por grupo	118
Tabla 6-3. Métricas para comparar la aplicación de las variantes del proceso de IR	120
Tabla 6-4. Resumen de mediciones obtenidas para el Sistema de Autogestión de Alumnos	120
Tabla 6-5. Resumen de mediciones obtenidas para el Sistema de Gestión de Biblioteca.....	121
Tabla 6-6. Análisis de la recomendación de adaptación del proceso de IR para cada caso.....	122
Tabla 6-7. Resultados del análisis de dos EF del Sistema de Autogestión de Alumnos	123
Tabla 6-8. Resultados del análisis de dos EF del Sistema de Gestión de Biblioteca	123
Tabla C-1. Resultados de Nivel de Confianza por Factor.....	150
Tabla C-2. Resultados de la Evolución de Valores y de Nivel de Confianza.....	151
Tabla D-1. Funcionalidades cubiertas para el Sistema de Autogestión de Alumnos	152
Tabla D-2. Funcionalidades cubiertas para el Sistema de Gestión de Biblioteca.	153
Tabla D-3. Resultados estadísticos para el Sistema de Autogestión de Alumnos	154
Tabla D-4. Resultados estadísticos para el Sistema de Gestión de Biblioteca.....	155

Capítulo 1

Introducción

1.1 Motivación

Para obtener un producto software de buena calidad es vital la realización apropiada de las actividades de la Ingeniería de Requisitos (IR), por tal motivo, con frecuencia la literatura presenta la necesidad de aplicar un proceso específico, bien definido, para la definición de los requisitos del software.

La experiencia adquirida en el desarrollo profesional de software indica que cada proyecto tiene características particulares, por las que se hace necesario efectuar cambios en los procesos definidos, de ahí la necesidad de disponer de un método claro que guíe la decisión en cuanto a cómo el proceso y los modelos a construir se deberían adaptar en cada caso.

Lo anterior aplica también a una estrategia de IR basada en modelos escritos en lenguaje natural (Leite, Doorn, Kaplan, Hadad & Ridaou, 2004), que se utilizará de base para este trabajo de tesis. Esta estrategia ha sido desarrollada a lo largo de varios proyectos de investigación desde 1995, difundida a través de cursos de grado y posgrado, y puesta en práctica en casos reales para varios proyectos de software en la industria. Aun cuando esta estrategia puede considerarse suficientemente madura, todavía requiere mejoras en algunos aspectos relevantes. Una de las mejoras posibles está justamente relacionada con la adaptación de la misma a las características del proyecto de software. Estas características se manifiestan en factores observables que atañen al contexto de aplicación y al proyecto mismo. En base a estos factores se pueden tomar decisiones referidas a qué caminos alternativos seguir, en otras palabras, qué artefactos de requisitos deben construirse, qué actividades del proceso de requisitos son necesarias realizar y qué técnicas específicas son más convenientes aplicar. Es decir, se propone adaptar el proceso de requisitos a cada proyecto de software, en base a factores situacionales imperantes en él.

En la práctica dicha tarea de adaptación a veces se realiza parcialmente y de un modo informal, dependiendo de la experiencia del ingeniero de requisitos que lleve

adelante el proyecto, sin un método o guía definida sobre qué parámetros deben ser considerados. Es muy probable que el adaptar la estrategia de una manera organizada y preplaneada, a cada circunstancia particular, tomando como fundamento diversos factores que puedan ser detectados en forma anticipada, podría mejorar la productividad del proceso de construcción de requisitos.

1.2 Alcances del Trabajo

En este trabajo de tesis se definirán cuáles son los puntos de variación del proceso, qué modelos se deben producir y con qué representación, y qué actividades y técnicas se deben aplicar en cada fase del proceso a fin de adaptar la estrategia de IR de acuerdo con un conjunto de factores situacionales definidos.

El proceso se fragmentará en un conjunto de componentes modulares, donde cada componente describirá una actividad, sus entradas y salidas, y las técnicas que se utilizarán. No se considerarán aquellos factores que sólo atienden la selección de técnicas de elicitación, dado que esto ha sido ampliamente estudiado en la literatura (Maiden & Rug, 1996; Carrizo, Dieste & Juristo, 2008).

Se determinará, para cada factor situacional, el grado de certeza esperable en su estimación “a priori” y la probabilidad de necesitar reevaluarlo en etapas intermedias del proceso de requisitos.

1.3 Hipótesis

Se supone que la adaptación del proceso de requisitos a una situación concreta redundará en una mejora del proceso.

Se está suponiendo además que los valores de gran parte de los factores situacionales pueden estimarse antes de iniciar el proceso de IR, permitiendo que el proceso pueda ser adaptado anticipadamente. Esto no impedirá posibles adaptaciones durante su avance.

Cuando se menciona la posible existencia de caminos alternativos, se está considerando que pueden existir componentes o bloques de proceso comunes a los caminos alternativos, que algunos bloques de proceso podrán ser alternativos y que algunos otros bloques de proceso podrán presentar alguna variabilidad interna

menor. Se asume que la variabilidad en las partes del proceso también influirá en los productos, pudiendo entonces manejar modelos ligeramente diferentes.

1.4 Objetivos

El objetivo general de esta tesis es mejorar la estrategia de IR basada en modelos en lenguaje natural, mediante el establecimiento de un proceso de adaptación de la misma según los diferentes factores situacionales.

Los objetivos específicos son los siguientes:

- a) Establecer los factores situacionales que comprometen la estrategia y los puntos de variación de la estrategia donde los factores impactan.
- b) Definir criterios que permitan estimar tempranamente los factores situacionales y su posible evolución.
- c) Identificar los bloques de proceso y de producto comunes para la estrategia básica y los bloques para variantes alternativas de la estrategia, además, de los criterios para la selección de los bloques adecuados a los puntos de variación.
- d) Desarrollar la heurística para construir el proceso de IR a ser utilizado en cada situación específica.

1.5 Organización del Documento

Esta tesis se organiza en 7 capítulos. En el presente capítulo se describe la motivación que dio impulso a este trabajo, su alcance, las hipótesis planteadas y también se establecen los objetivos.

En el capítulo 2 se presenta el marco teórico, donde se exponen aspectos relacionados a los procesos de IR y se describe la estrategia de requisitos basada en lenguaje natural en la cual se basa esta tesis. Se introducen conceptos provenientes de dos disciplinas que tratan la adaptabilidad: la variabilidad de procesos y la Ingeniería de Métodos Situacional. Finalmente se analizan propuestas de adaptabilidad en procesos de IR.

En el capítulo 3 se determina el conjunto de factores situacionales que influyen para la adaptación de la estrategia de requisitos, con sus valores admisibles. Además, se detallan las dependencias que pueden existir entre estos.

En el capítulo 4 se presentan los mecanismos de adaptación situacional para la estrategia de requisitos. Esto incluye, por una parte, establecer las reglas de adaptación en función de los valores asignados a los factores situacionales, y por otra, describir los bloques de proceso necesarios para componer cada una de las variantes de la estrategia.

En el capítulo 5 se propone un proceso para la construcción del proceso de requisitos, que permite su adaptación en forma dinámica, considerando la estimación y la evolución de los factores situacionales a medida que el proceso de IR avanza.

En el capítulo 6, se expone un estudio de caso realizado, en el cual se aplican las tres variantes de la estrategia de requisitos que se consideran más significativas, y se realiza un análisis de los resultados obtenidos.

Finalmente, en el capítulo 7 se presentan las conclusiones del trabajo realizado, tomando como referencia los objetivos planteados. Además, se proponen posibles futuros trabajos de investigación sobre la base de esta tesis.

Por otro lado, se incluyen 4 anexos, referidos al siguiente contenido:

- Anexo A: presenta un análisis sobre 40 factores situacionales.
- Anexo B: contiene un formulario para evaluar factores situacionales.
- Anexo C: incluye información adicional del estudio de caso presentado en el capítulo 5.
- Anexo D: contiene información adicional del estudio de caso presentado en el capítulo 6.

Capítulo 2

Marco Teórico

2.1 Introducción

La adaptación de cualquier proceso a situaciones particulares se considera una buena práctica en la mayoría de las disciplinas. La literatura muestra que esta práctica es muy común en los procesos de Ingeniería de Software, tales como las metodologías Rational Method Composer (Haumer, 2005) y Open Process Framework (Firesmith & Henderson-Sellers, 2002).

Implementar una gestión adecuada de los procesos permite a las organizaciones no solo reducir esfuerzos, sino además obtener productos de mayor calidad y elevar el nivel de abstracción de la planificación permitiendo proyecciones de más largo plazo y manejo de carteras de proyectos. Esto no resulta sencillo, especialmente cuando se trabaja en ambientes dinámicos, los cuales conllevan un importante nivel de variabilidad.

En algunas circunstancias los gerentes de proyecto deben considerar la posibilidad de acortar el proceso de requisitos mediante la simplificación u omisión de algunas actividades, mientras que en otras pueden surgir diferentes alternativas, tales como la elección de ciertas técnicas de elicitación, de modelado o de validación, inclusive podría ser oportuno agregar algunas otras actividades. Sería deseable que estas decisiones sobre las elecciones, refinamientos o ajustes al proceso puedan tomarse, para cada proyecto particular, en una forma consciente y planificada.

Dado que la mira central de esta tesis está puesta en proporcionar mecanismos para la adaptación de un proceso de IR, de acuerdo con el contexto o situación en el que se desarrolla el proyecto, se hace necesario plantear algunos conceptos que sirvan de ejes sobre los que apoyar la propuesta. Este capítulo presenta, en primer término, las visiones de distintos autores sobre los procesos de IR y el proceso específico basado en modelos escritos en lenguaje natural sobre el que se elabora la propuesta de adaptación. Posteriormente se exponen nociones de la variabilidad de procesos y conceptos de Ingeniería de Métodos Situacional. Por último, se

estudian distintas propuestas de la literatura relacionadas a la adaptación del proceso de IR.

2.2 Visiones del Proceso de Ingeniería de Requisitos

Cuando se trabaja en el desarrollo de un producto es importante contar con un proceso bien definido, lo cual implica disponer de una serie de actividades predefinidas, con técnicas, entradas y salidas, que en su conjunto ayuden a conseguir un resultado de alta calidad. La primera actividad de estos procesos consiste en definir el resultado esperado. Cuando el producto es un sistema de software, la actividad inicial es un proceso de IR, cuyo resultado es la especificación de los requisitos del software (Sommerville, 2010).

Ampliando lo anterior, se puede decir que un proceso de IR es un conjunto estructurado y coherente de tareas, procedimientos, productos de trabajo, políticas, estructuras organizativas y tecnologías necesarias para identificar, analizar, especificar, validar y administrar los requisitos con un alto nivel de calidad (Zowghi, Firesmith & Henderson-Sellers, 2005). Su objetivo es producir el mejor conjunto de requisitos del software que, en la medida de lo posible, sea completo, coherente, y refleje lo que el cliente realmente necesita. Aunque este ideal probablemente parezca inalcanzable, la aplicación de un enfoque sistemático, basado en principios de ingeniería, conduce a mejores requisitos que un enfoque informal, sin usar un proceso predefinido (Sommerville & Sawyer, 1997).

En las últimas tres décadas se han propuesto diversos modelos para el proceso de IR; cada uno de ellos se enfoca en diferentes aspectos, y tienen diferente granularidad, sin embargo, a pesar de estas variaciones, existe una serie de actividades genéricas comunes en todas estas propuestas.

Pressman (2010) resume las actividades del proceso de requisitos según los siguientes pasos, no necesariamente secuenciales:

1. *Concepción*, los participantes establecen un entendimiento básico del problema, las restricciones generales del proyecto, además de las funciones principales y características que debe cumplir el sistema;

2. *Indagación* a fin de aumentar y mejorar la información obtenida en la etapa de concepción, el foco se pone en la obtención de los requisitos de modo organizado¹;
3. *Elaboración* de un modelo de requisitos refinado, que identifique distintos aspectos de la función del sistema software, su comportamiento e información;
4. *Negociación* a fin de solucionar los conflictos detectados en los requisitos, como también para evaluar la prioridad, la disponibilidad y el costo relativo de cada uno de ellos;
5. *Especificación* de los requisitos del sistema; la formalidad y el formato de esta documentación puede variar de acuerdo con el tamaño y la complejidad del sistema a construir;
6. *Validación* de cada requisito con el objetivo de garantizar que se va a construir el sistema correcto;
7. *Administración de los requisitos* para llevar un control sobre estos, darles seguimiento y gestionar los cambios que pudieran surgir en cualquier momento del proceso de desarrollo.

De acuerdo con el enfoque de Wiegers & Beatty (2013), la Figura 2-1 muestra una representación del dominio de la IR en forma jerárquica dividida en dos grandes áreas de trabajo: i) el desarrollo de los requisitos que, incluye actividades esenciales para construir los requisitos del sistema; y ii) la gestión de requisitos que se realiza en el transcurso de todo el ciclo de vida del proyecto.

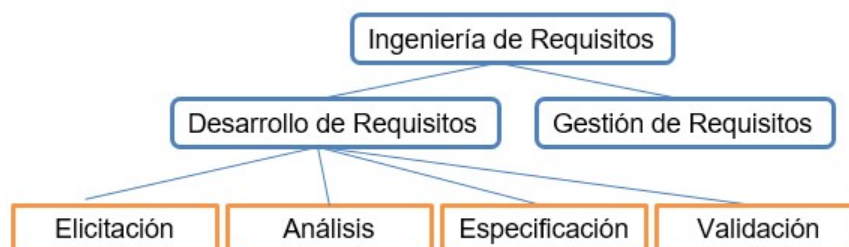


Figura 2-1. Jerarquía del dominio de la IR. Fuente: Wiegers & Beatty, 2013

En la Figura 2-2 se puede ver un proceso de producción de requisitos, el cual es naturalmente iterativo e involucra las actividades de elicitación, modelado, análisis de los requisitos, y su gestión; para cada una de las actividades se definen sus

¹ La visión de la IR es la de construir requisitos a partir de la información elicitada, y no centrarse meramente en elicitar requisitos.

entradas y salidas. Se debe notar que existen interacciones entre dichas actividades y que las mismas pueden realizarse de modo concurrente, así, por ejemplo, a medida que se va alcanzando conocimiento (elicitación), se va modelando y analizando la información obtenida (Hadad, 2008).

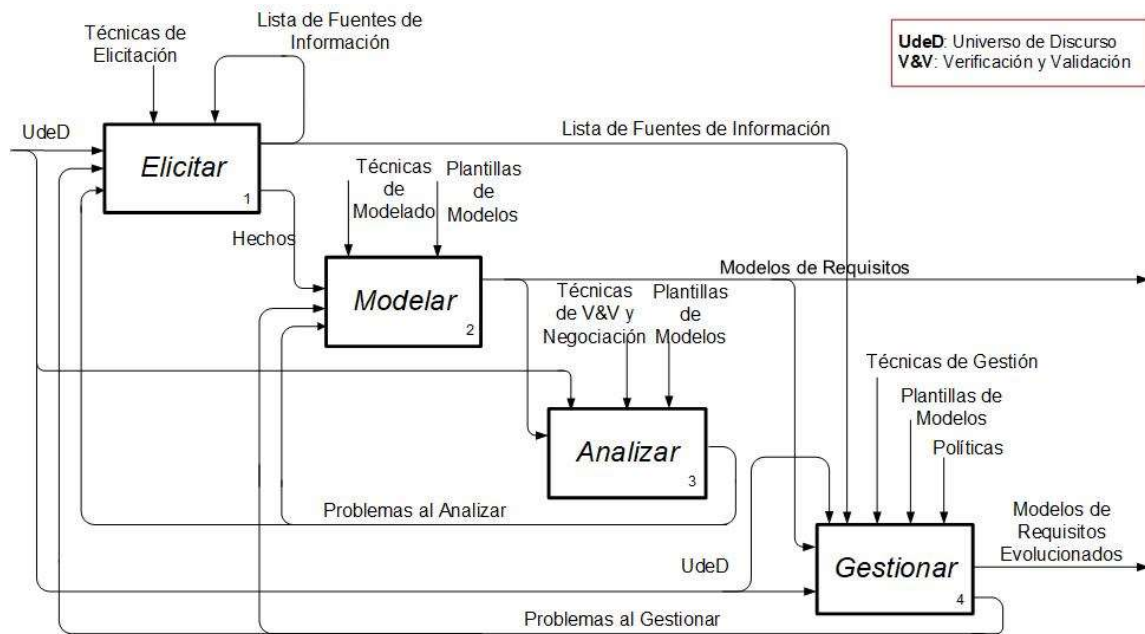


Figura 2-2. Proceso de IR. Adaptada de Hadad, 2008

De las visiones expuestas antes, se consigue deducir que la IR no consiste en un proceso secuencial, sino que se trata de un proceso iterativo, por el cual los requisitos se van construyendo y van evolucionando. Salvando las diferentes terminologías utilizadas por los autores, se puede concluir que la IR incluye las siguientes actividades principales: la elicitación², la modelización y el análisis de los requisitos, más una cuarta transversal a las demás, la gestión de dichos requisitos (Hadad, 2008; Aguilar, Garrigós & Mazón, 2016). La Tabla 2-1 resume las actividades principales del proceso de IR, y un conjunto de subactividades o tareas asociadas a cada una de ellas.

Tabla 2-1. Actividades de la IR. Fuente: Hadad, 2017

Elicitar	Modelar	Analizar	Gestionar
Identificación de fuentes de información	Representación	Verificación	Identificación de cambios
Recolección de hechos	Organización	Validación	Análisis de cambios
Comunicación	Almacenamiento	Negociación	Realización de cambios

² Elicitar: descubrir, explicitar, obtener el máximo de información del objeto bajo estudio.

A continuación, se describen brevemente estas actividades:

- *Elicitar*: es la primera actividad en el proceso de aproximación a una comprensión la necesidad que se quiere atender con el producto software. Va mucho más allá de la recopilación o relevamiento de requerimientos, se trata de descubrir los hechos relevantes del contexto de la aplicación, es decir, es una actividad de adquisición de conocimiento, durante la cual es crucial la comunicación que se establece entre los involucrados.

En primer lugar, se procede a la *identificación de las fuentes de información*, es decir, todo aquello que ayude a comprender tanto el dominio, como el problema en sí mismo. Si el conocimiento proviene de personas, el proceso de adquisición se denomina educación de conocimientos, mientras que, si se obtiene de fuentes no humanas, por ejemplo, en forma escrita, audios, otros sistemas, etc., el proceso se llama extracción de conocimiento (Rodríguez-Lora, Henao-Cálad & Valencia Arias, 2016). Dado que la cantidad de fuentes detectadas puede ser excesiva, una vez identificadas deben ser priorizadas para determinar cuales se eligen para avanzar en el proceso (Zanlorenci & Burnett, 1998; Alexander & Robertson, 2004; Leite , Moraes & Castro, 2007; Ballejos, 2009).

Paso siguiente, se debe seleccionar el conjunto de técnicas que mejor se adecuen para la *recolección de hechos* a partir de las fuentes de información disponibles. Las técnicas que se apliquen serán distintas dependiendo del tipo de fuente de información que se trate y del objetivo de la recolección en sí misma, es decir, del tipo de resultados que se busquen (Kotonya & Sommerville, 1998; Carrizo Moreno, 2009).

Existen diferentes técnicas que se aplican en la elicitación, tales como entrevistas, observación, entre otras; así también, se han desarrollado una cantidad de trabajos asociados a la selección de técnicas de elicitación (Maiden & Rug, 1996; Carrizo, Dieste & Juristo, 2008), por lo cual en esta tesis no se aborda esta temática como parte de la adaptación del proceso de IR.

Resulta fundamental, especialmente en esta etapa, las habilidades de *comunicación* del ingeniero de requisitos, por una parte, con los usuarios y otros involucrados, identificando sus puntos de vista, promoviendo su participación de modo activo, como así también internamente con el equipo de desarrollo,

logrando elaborar un nivel de abstracción adecuado a la información capturada (Hadad, 2008).

- *Modelar*: se debe representar, organizar y almacenar los hechos recolectados en la elicitación, como así también los requisitos que se elaboran. Es decir, se debe documentar de forma comprensible, significativa y con un nivel de abstracción toda la información obtenida y procesada, generando los modelos que representan el problema actual y/o que representarán el sistema a construir.

Se debe seleccionar la técnica de *representación* para cada modelo en particular. Existen numerosas técnicas de modelado estas podrían dividirse en dos grandes categorías: aquellas que se basan en lenguaje formal (Pressman, 2010), por ejemplo, OCL³ y lenguaje Z⁴, entre otras; por otra parte, las basadas en lenguaje natural, tales como casos de uso, escenarios, glosarios e historias de usuario⁵ (Escalona & Koch, 2004; Leite, Doorn, Kaplan, Hadad & Ridaou, 2004; Hadad, 2008).

En resumen, estos modelos tienen su propia representación y *organización* que el equipo de ingenieros de requisitos debe respetar. Además, debe considerarse su *almacenamiento*, de modo tal que resulte sencillo el registro y recuperación de información de los modelos.

- *Analizar*: tiene que ver con verificar y validar los hechos modelados. La *verificación* es una tarea interna que se encarga de controlar la consistencia de un modelo o entre los modelos generados, para asegurar su coherencia, evitar contradicciones y ambigüedades. Es decir, permite determinar si el modelo se ha construido correctamente. Por su parte, la *validación* es una actividad externa, que se ocupa de determinar si el modelo refleja la realidad actual o la esperada

³ OCL (Object Constraint Language): es una notación formal que combina lógica de predicados y teoría de conjuntos, que permite describir expresiones y restricciones en modelos y artefactos orientados a objetos (Pressman, 2010).

⁴ Z es un lenguaje de especificación que se basa en la teoría de conjuntos descrita y en lógica de primer orden; emplea esquemas para la representación del espacio y las operaciones de estado de una especificación (Pressman, 2010).

⁵ Las historias de usuario son descripciones cortas de una funcionalidad que es de valor para el cliente. Su utilización es común para capturar requisitos cuando se aplican marcos de trabajo ágiles (Beck & Fowler, 2000).

que satisfaga las necesidades y expectativas del cliente. Existen diversas técnicas para llevar a cabo la verificación y validación de modelos y documentos realizados (Leite, 1994; Hadad, 2008; Bilal, Ilyas, Tariq & Hummayun, 2016).

Así mismo, se debe incluir en esta actividad la *negociación* de requisitos entre los involucrados. Esto puede ser necesario cuando los ingenieros de requisitos proponen alternativas para la solución del problema en cuestión, o también, por conflictos detectados en las fuentes de información o cuando se deben seleccionar requisitos o cuando se debe establecer el orden de puesta en servicio. En todo caso, implica el intercambio de información, la discusión, la resolución de conflictos, el acuerdo de propuestas con los clientes y usuarios, y la asignación de prioridades a los requisitos.

- *Gestionar*: tal como se mencionó anteriormente, se trata de una actividad transversal a las otras tres, que se realiza durante todo el desarrollo del sistema. Se encarga no sólo de la *identificación de cambios* originados por algún problema en los requisitos, o un pedido de modificación, o de nuevas necesidades, sino también del *análisis de cambios*, lo cual incluye analizar su validez, evaluar y negociar el posible impacto en otros requisitos, en el alcance, la planificación y el presupuesto del proyecto, para finalmente determinar si se prosigue con los cambios. Si hubo aceptación, se procede a la *realización de cambios*, es decir, estos son implementados en los documentos de requisitos y en los modelos, según corresponda, y nuevamente se deben verificar y validar.

Por lo expuesto anteriormente, se hace evidente que el proceso de IR es particularmente diferente de otras partes del proceso de desarrollo de software, ya que es el que más se relaciona con las personas y su entorno, mientras que otras actividades se realizan principalmente dentro del equipo de desarrollo (Carrizo Moreno, 2009). Además, las decisiones sobre un proyecto imponen restricciones, las que muchas veces definen las herramientas y técnicas a aplicar para llevar a cabo las actividades. Es por ello que, si se toman en cuenta las particularidades que rodean al contexto de aplicación y al proyecto en sí mismo, es posible que se consiga un proceso más eficiente, que obtenga como resultado una mejor Especificación de Requisitos de Software (ERS).

Dado que el conocimiento del dominio del problema se expresa sobre todo en lenguaje natural, el uso de un enfoque basado en representaciones de requisitos en lenguaje natural favorece el compromiso de los clientes y usuarios, lo que incrementa la probabilidad de éxito del proyecto (Macaulay, 1993). Una manera de lograr ese compromiso es mantener una apropiada y continua comunicación entre las partes interesadas.

Una buena comunicación se consigue fundamentalmente cuando todos los involucrados utilizan un mismo lenguaje. En la IR, esto se logra por medio del uso de un vocabulario común. Es preferible utilizar el mismo vocabulario que el empleado en el contexto de aplicación (Leite, Doorn, Kaplan, Hadad & Ridaou, 2004). En este sentido, los modelos en lenguaje natural, tales como glosarios, casos de uso y escenarios, favorecen la comunicación con los involucrados, siendo los modelos más difundidos en la IR (Potts, 1995; Kaindl, 2000; Cockburn, 2000; Leite, Hadad, Doorn & Kaplan, 2000; Leffingwell & Widrig, 2003; Antonelli, Rossi, Leite & Oliveros, 2012).

Esta tesis toma como base una estrategia, que consiste en un refinamiento del proceso de IR, la cual construye y utiliza modelos basados en lenguaje natural, produce un glosario que contiene los términos del dominio y aplica la técnica de escenarios con la finalidad de obtener conocimiento acerca del problema y capturar los requisitos del software. A continuación se describe dicha estrategia con sus principales actividades.

2.3 Estrategia de Ingeniería de Requisitos basada en Modelos de Lenguaje Natural

Esta estrategia es un enfoque dentro de la IR, que se sustenta en dos modelos escritos en lenguaje natural:

- ✓ un glosario especial que captura el vocabulario del dominio de la aplicación, denominado Léxico Extendido del Lenguaje (LEL), y
- ✓ escenarios, que son descripciones de situaciones que ocurren en el universo de discurso⁶ (UdeD), que ayudan a obtener conocimiento acerca del

⁶ Universo de Discurso: representa todo el contexto en el cual el software se desarrolla e incluye todas las fuentes de información y toda las personas relacionadas con el software. Es la realidad acotada por el conjunto

problema y a capturar los requisitos del software, proponiendo alternativas de solución.

En la Figura 2-3, a través de un modelo SADT⁷, se esquematiza la manera en que la estrategia propone un refinamiento del proceso de producción de requisitos, que fundamentalmente involucra cuatro fases principales:

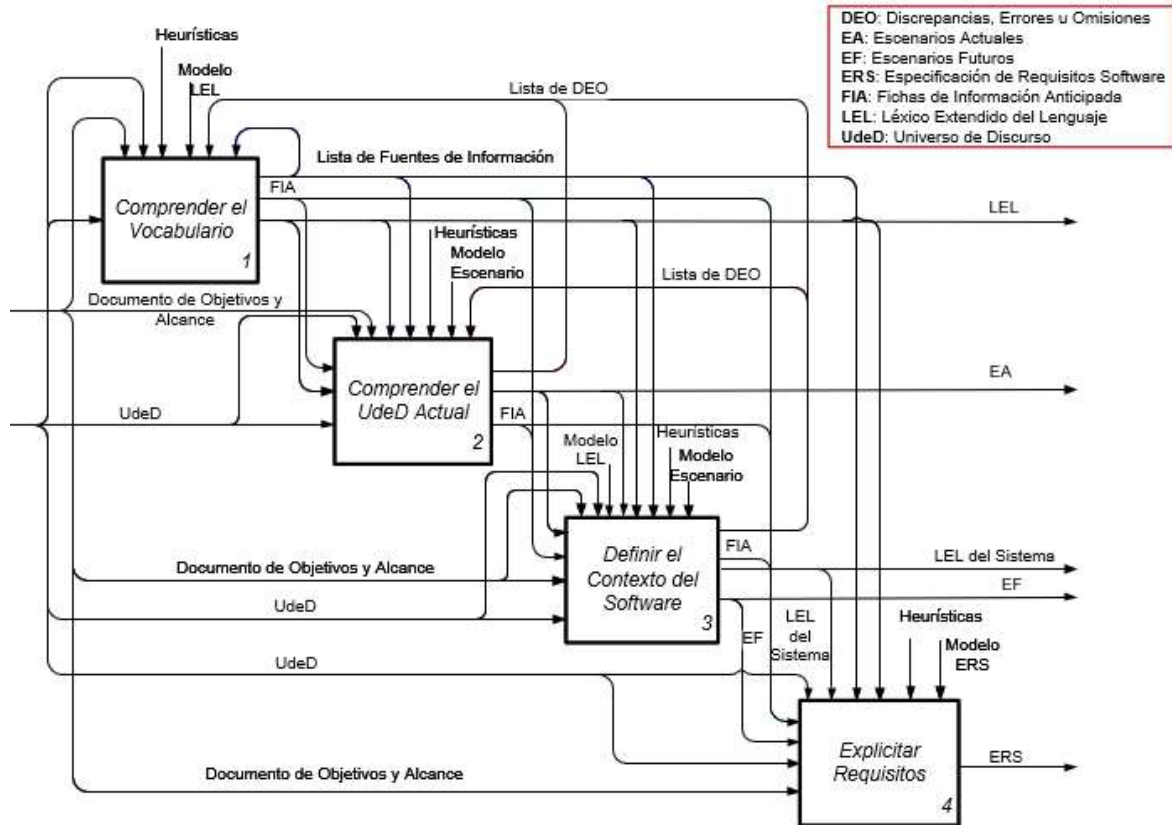


Figura 2-3. Estrategia de IR basada en modelos de lenguaje natural. Adaptada de Hadad, 2008

- 1) *Comprender el vocabulario* del contexto de aplicación, con el apoyo del modelo LEL del UdeD (Hadad, Doorn & Kaplan , 2009).
- 2) *Comprender el UdeD actual*, para ello se construye un conjunto de escenarios que representan el comportamiento observado en el contexto de aplicación; estos se denominan Escenarios Actuales (EA) (Leite, Hadad,

de objetivos establecidos por quienes demandan una solución de software. Las personas involucradas son principalmente: usuarios, clientes, ingenieros de software, expertos del dominio, las cuales son denominadas “stakeholders” en la literatura (Leite, 1994).

⁷ SADT: es una notación en la cual las cajas representan actividades; las flechas a la izquierda representan entradas requeridas por la actividad; las flechas hacia abajo representan controles; y las flechas a la derecha representan salidas desde la actividad (Ross & Schoman, 1977).

Doorn & Kaplan, 2000). Estos EA se describen usando los términos definidos en el LEL del UdeD.

- 3) *Definir el contexto del software*, mediante la modificación de los escenarios previos, se produce un conjunto nuevo de escenarios que representan el comportamiento proyectado en el contexto de aplicación donde funcionará el sistema de software a desarrollar, llamados Escenarios Futuros (EF) (Hadad, 2008). Acompañando a los EF, se construye el LEL del Sistema partiendo del LEL del UdeD (Kaplan, Doorn & Gigante, 2013). Este LEL del Sistema define los términos del UdeD con adaptaciones o agregados que se incorporarán como consecuencia del mismo sistema de software. Es decir, este LEL define los términos utilizados en los EF intentando preservar el vocabulario que se utiliza en el UdeD. El principal objetivo de este LEL es facilitar a los usuarios la comprensión de las descripciones contenidas en los EF.
- 4) *Explicitar requisitos*, produciendo el documento de ERS, después de haber extraído los requisitos del conjunto de EF (Hadad, Doorn & Kaplan, 2009 b). El documento de ERS se escribe respetando el vocabulario del UdeD, con las modificaciones o términos nuevos que se definieron en el LEL del Sistema según la solución propuesta en los EF.

La estrategia se enfoca en alcanzar la máxima comprensión del UdeD, con el objetivo de lograr que los ingenieros de requisitos puedan planificar cambios al macrosistema fácilmente insertables en la cultura de la organización y simultáneamente compatibles con los recursos disponibles.

La Figura 2-3 presenta en un flujo secuencial las etapas de la estrategia, aunque la realidad es que existen reciclos debido a la retroalimentación que se produce por actividades de verificación y validación, y a la mejora continua en la comprensión del problema y por la evolución propia del UdeD.

Así mismo, es frecuente que se elicite cierta información que es importante pero que no corresponda al modelo que se está construyendo en dicha etapa, sino que es información útil para alguna etapa posterior; toda esta información se vuelca en registros auxiliares de las tres primeras etapas del proceso, denominados Fichas de Información Anticipada.

En las siguientes subsecciones, se describen y justifican cada una de las etapas que componen esta estrategia.

2.3.1 Comprender el Vocabulario

Como ya se indicó el LEL es un modelo muy útil durante todo el proceso de desarrollo del software, pues facilita la comunicación escrita y oral entre los involucrados y reduce la ambigüedad en los artefactos producidos durante la IR, como así también en aquellos generados en otras etapas que contengan algún tipo de descripción textual.

El propósito de crear este léxico no sólo es permitir una buena comunicación y acuerdo entre los involucrados, sino también ser el punto de inicio para el proceso de construcción de EA (Leite, Hadad, Doorn & Kaplan, 2000), ayudar en la descripción de los mismos y facilitar el proceso de validación de los EA.

El proceso de creación del léxico, expuesto en la Figura 2-4, comprende cinco actividades independientes:

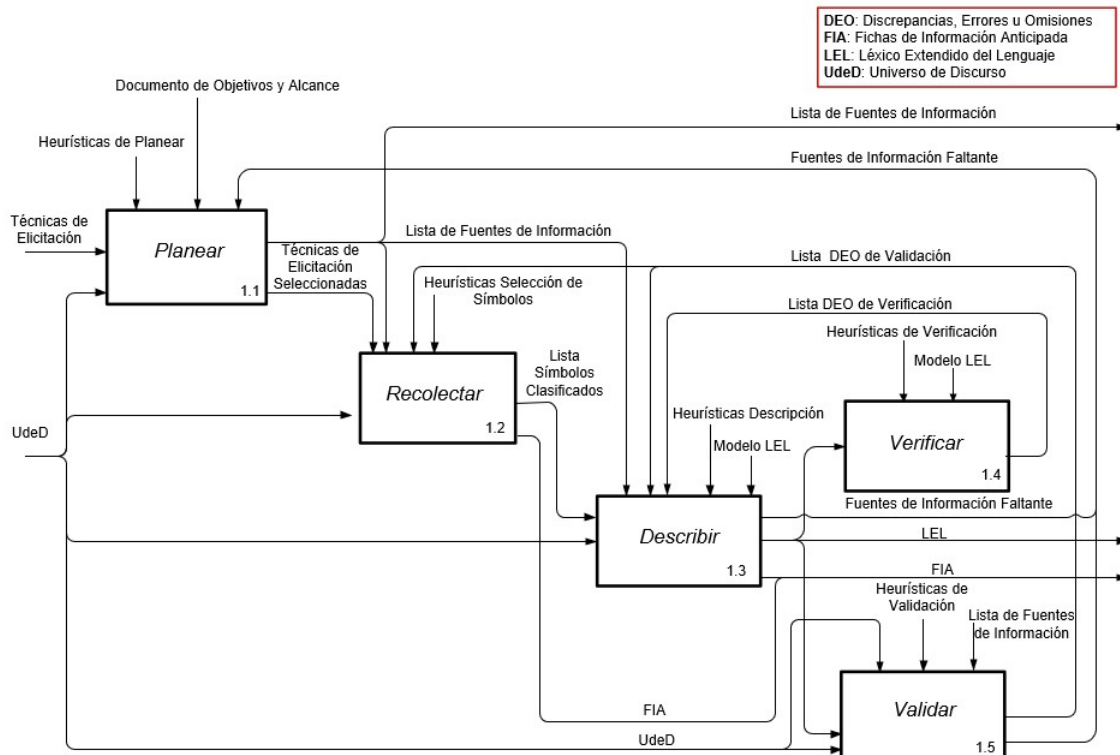


Figura 2-4. Proceso de creación de LEL. Adaptada de Hadad, 2008

- 1) *Planear*, para establecer cómo se obtendrá la información del UdeD. Se obtienen y seleccionan las fuentes de información y se decide sobre las técnicas de elicitación que se utilizarán.

- 2) *Recolectar* el vocabulario, implica identificar los términos o símbolos relevantes en el UdeD, los cuales se organizan en una lista y finalmente se clasifican.
- 3) *Describir* los símbolos, requiriéndose para ello la captura de más información del UdeD; los ingenieros de requisitos deben asegurarse que estos términos efectivamente sean los utilizados por el cliente, los usuarios y otras fuentes disponibles como documentos, software en uso, y otros.
- 4) *Verificar* el LEL, se realiza siguiendo heurísticas que faciliten la detección de defectos, mediante el uso de la técnica de inspección.
- 5) *Validar* el LEL con los usuarios permite que los ingenieros corrijan, corroboren o mejoren el conocimiento sobre el vocabulario del dominio de la aplicación, aclarando dudas, corrigiendo las definiciones y agregando o eliminando símbolos. Esta actividad, al igual que la anterior de verificación, posibilita detectar discrepancias, errores u omisiones.

2.3.2 Comprender el Universo de Discurso Actual

El objetivo de esta etapa es conocer el UdeD tal cual es en la actualidad, para ello se construye un conjunto de escenarios que representan el comportamiento en el contexto presente, es decir, los EA.

Dichos escenarios son narrativas estructuradas de situaciones enmarcadas en el entorno organizacional, que se construyen utilizando lenguaje natural con cierto formalismo y empleando los términos definidos en el LEL.

El proceso para la construcción de EA se describe en la Figura 2-5, donde se pueden observar las siguientes actividades:

- 1) *Derivar*, esta actividad implica identificar los actores del UdeD y generar escenarios candidatos utilizando información contenida en el LEL.
- 2) *Describir* los escenarios, añadiendo información del UdeD para completar los componentes del modelo de escenario.
- 3) *Organizar* el conjunto de EA conseguidos en la actividad previa, implica reorganizarlos aplicando operaciones de composición y descomposición de escenarios e integrarlos. Se generan los escenarios integradores, los cuales brindan una visión global de las relaciones entre varias situaciones.

- 4) *Verificar*, se puede realizar al menos luego de la actividad *Describir* sobre el conjunto de escenarios completamente descritos y después de la actividad *Organizar*, aplicando la técnica de inspección.
- 5) *Validar* cada escenario con los usuarios, para confirmar que la información elicitada se corresponde con las situaciones que realmente suceden en el UdeD.

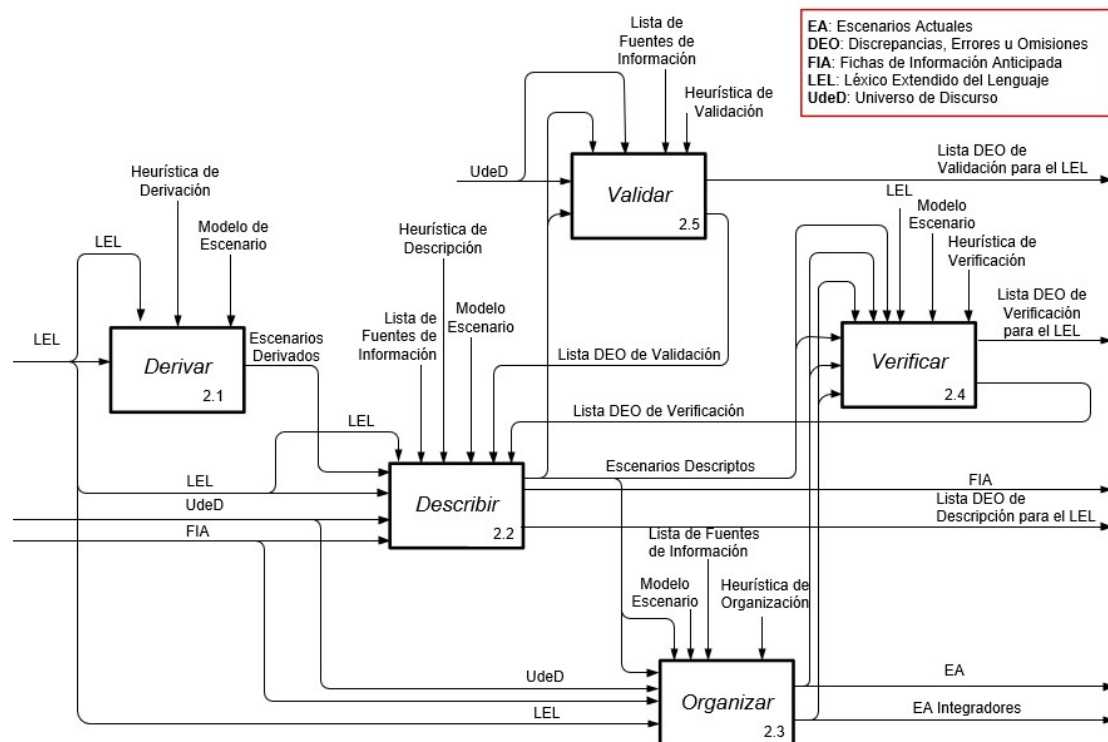


Figura 2-5. Proceso de construcción de EA. Adaptada de Hadad, 2008

Si bien el proceso esquematizado muestra un flujo principal compuesto por las actividades *Derivar*, *Describir* y *Organizar*, éstas no son estrictamente secuenciales, sino que pueden ejecutarse simultáneamente. A su vez, las actividades *Verificar* y *Validar* permiten una retroalimentación, que aporta una segunda instancia de paralelismo.

Como consecuencia de adquirir un conocimiento más preciso sobre el UdeD, se puede mejorar la comprensión del vocabulario, por lo que se genera la lista de defectos a aplicar sobre el propio LEL.

2.3.3 Definir el Contexto del Software

Durante esta fase del proceso se generan los EF, los cuales proponen y modelan situaciones proyectadas para satisfacer las demandas y necesidades planteadas

en el UdeD, y de acuerdo con los objetivos propuestos para el sistema (Doorn, Hadad & Kaplan, 2002). Estos escenarios concentran los requisitos funcionales⁸ del software en sus descripciones, como también los no funcionales⁹.

Durante la construcción de los EF se va generando el LEL del Sistema según los términos relevantes utilizados en las descripciones de las situaciones futuras intentando respetar la terminología utilizada en el UdeD.

La Figura 2-6 esquematiza el proceso de construcción de EF, el que involucra las siguientes actividades:

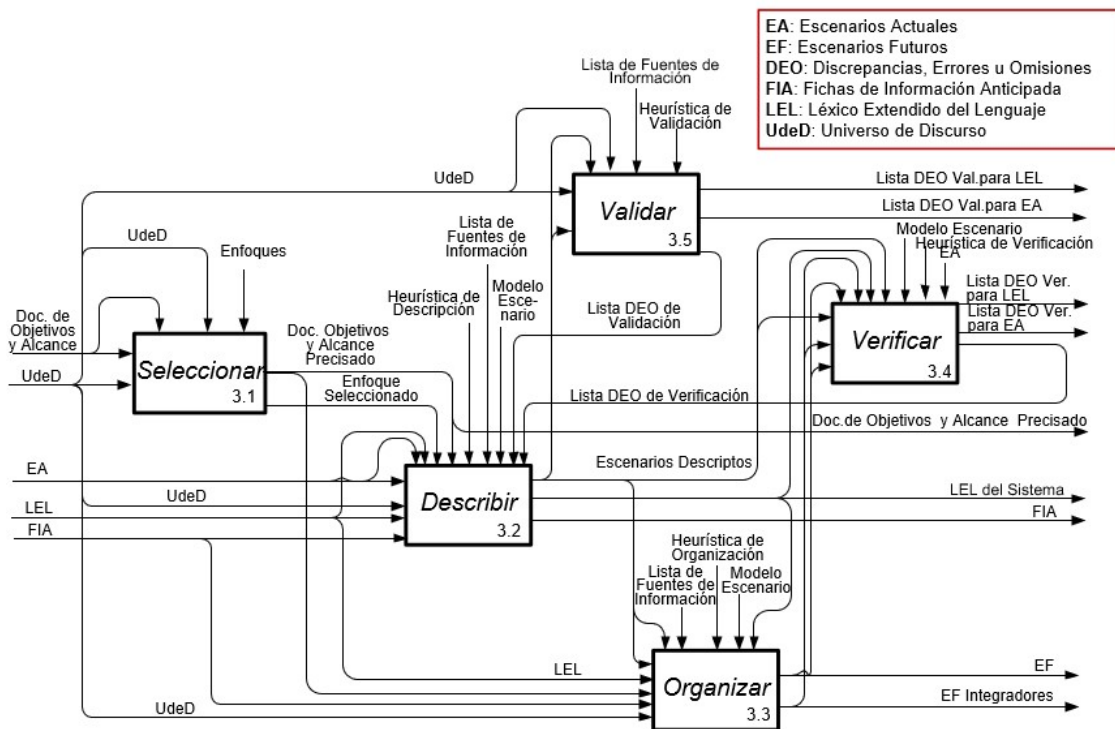


Figura 2-6. Proceso de construcción de EF. Adaptada de Hadad, 2008

- 1) *Seleccionar* el enfoque de construcción de los EF. Para lo cual, se toma en consideración los objetivos preliminares del sistema, estos se refinan y se establece el grado de reingeniería de procesos esperado. Se entiende que

⁸ Requisitos Funcionales: actividades y servicios que debe proveer el sistema. Es decir, expresan el comportamiento esperado del sistema. En algunos casos, pueden expresar lo que el sistema no debe hacer (Sommerville, 2010).

⁹ Requisitos No Funcionales: características y atributos del sistema, así como también cualquier restricción que pueda limitar una solución. Es decir, expresan restricciones y cualidades que el sistema debe cumplir o poseer. En algunos casos, pueden referirse a restricciones al proceso de desarrollo que impactan en restricciones al software mismo y también restricciones derivadas del UdeD (Sommerville, 2010).

un grado de reingeniería bajo ocurre cuando las actividades que se planifica tendrán lugar en el futuro son muy similares a las que tienen lugar actualmente, siendo el sistema de software una mera automatización de tareas. Por otra parte, se entiende que un grado de reingeniería alto tiene lugar cuando las actividades que se planifica tendrán lugar en el futuro, son muy diferentes de las actuales, involucrando cambios en los circuitos de información, en los documentos elaborados y en los procesos involucrados, siendo en este caso el sistema de software un recurso imprescindible para tales cambios. En base a ello, se opta por uno de los siguientes enfoques:

- Orientado a los procedimientos, derivando los EF directamente de los EA,
 - Dirigido por objetivos, construyendo los EF en función de los objetivos establecidos para el software a construir, o
 - Híbrido, construyendo los EF mediante una combinación de los dos enfoques precedentes.
- 2) *Describir*, involucra elicitar, negociar y modelar el contexto con el nuevo sistema de software. Se generan los EF aplicando el enfoque seleccionado en la actividad anterior.
 - 3) *Organizar*, sigue las mismas acciones implicadas en la organización de los EA, donde se aplican operaciones de composición y descomposición, y se construyen los escenarios integradores, para obtener una proyección global de la aplicación.
 - 4) *Verificar* mediante la técnica de inspección el conjunto de EF, para asegurar su consistencia, completitud y la satisfacción de los objetivos del sistema de software.
 - 5) *Validar* los EF aplicando distintas técnicas dependiendo del grado de cambio proyectado respecto del comportamiento actual, facilitando que los usuarios corroboren o rectifiquen las descripciones realizadas.

Como se puede notar, la construcción de los EF constituye una etapa fundamental, dado que permite el acuerdo de los clientes y usuarios respecto de las funcionalidades y características que tendrá el sistema de software y el contexto en el cual este va a operar.

Al describir los EF pueden surgir propuestas alternativas o diferentes a los pedidos de los clientes, por lo que, estas propuestas deben evaluarse y negociarse entre todos los involucrados.

Junto con la construcción de los EF, se crea el LEL del Sistema preservando en la medida de lo posible la terminología del LEL del UdeD, modificando las definiciones de algunos símbolos existentes y agregando nuevos términos utilizados en las descripciones de los EF. Es decir, estos EF contienen vínculos a los símbolos definidos en el LEL del Sistema.

2.3.4 Explicitar Requisitos

Una vez que se ha llegado a un acuerdo sobre el conjunto de EF, que incluirán las funcionalidades y características del sistema de software, se extraen de estos los requisitos del sistema.

Se redacta el documento de ERS, cuya formalidad dependerá de las exigencias organizacionales con relación al uso de estándares o de las políticas establecidas para el proyecto en particular.

Las actividades involucradas en el proceso de explicitar requisitos del software se muestran gráficamente en la Figura 2-7, y se describen a continuación:

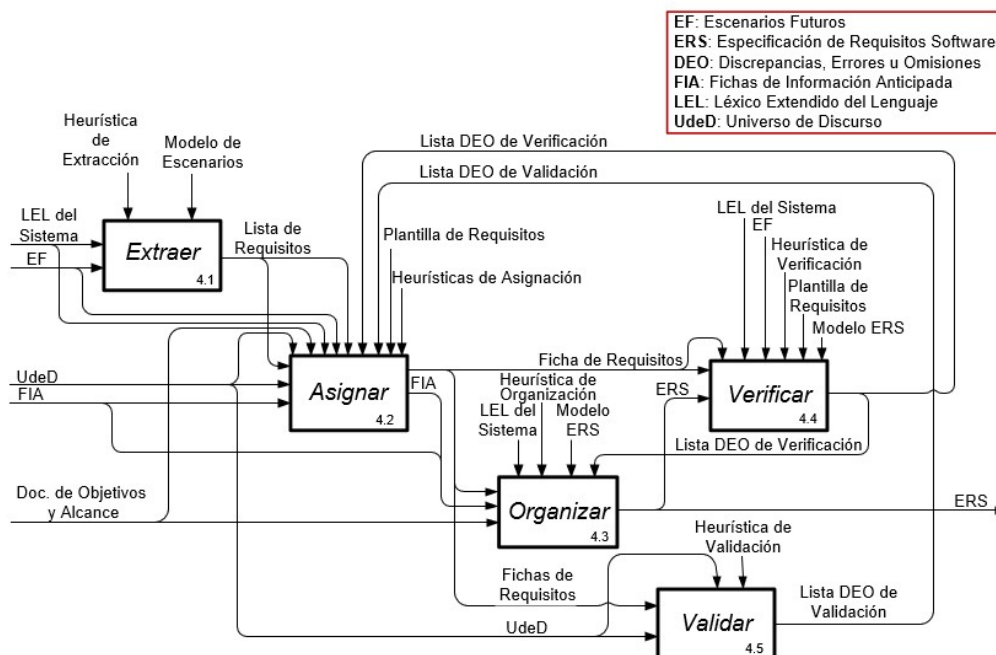


Figura 2-7. Proceso de construcción de ERS. Adaptada de Hadad, 2008

- 1) *Extraer* requisitos de los EF para generar una lista de requisitos únicos del sistema de software, mediante el uso de reglas de extracción de requisitos funcionales y no funcionales según componentes del Modelo de Escenario.
- 2) *Asignar* atributos a cada requisito de la lista generada en la actividad anterior. Esto incluye completar su descripción y además establecer los restantes atributos, tales como la volatilidad, prioridad, criticidad, factibilidad y riesgo. Esta actividad es realizada en forma colaborativa entre los clientes y usuarios y los ingenieros de requisitos.
- 3) *Organizar* los requisitos con el objetivo de lograr la mayor facilidad de comprensión posible. La especificación se realiza de acuerdo con el formato de ERS solicitado por el cliente o adoptado por el equipo de desarrollo.
- 4) *Verificar* para asegurar que se haya extraído la totalidad de requisitos desde los EF, que se haya realizado la asignación de atributos de modo consistente y, que el documento de ERS haya sido construido correctamente, que esté lo más completo posible y que cumpla con el estándar adoptado. Esta actividad se desarrolla separadamente luego de *Asignar* los atributos a los requisitos extraídos de los EF y luego de la actividad *Organizar*.
- 5) *Validar* las especificaciones a fin de que los involucrados puedan confirmar o rectificar los valores asignados a los atributos de los requisitos.

Entre las ventajas que corresponde destacar de la aplicación de esta estrategia de IR es la calidad de los requisitos conseguidos, la razón es que los mismos fueron obtenidos a partir de los EF, los cuales fueron debidamente verificados, validados y acordados y, por otra parte, no presentan conflictos, ya que en caso de haberse detectado se han solucionado por medio de la negociación a través de los propios EF.

2.4 Variabilidad de Procesos

Cuando los procesos se ejecutan en entornos dinámicos, los constantes cambios de contexto que surgen en dichos entornos crean la necesidad de adaptar continuamente estos procesos. Adaptarse y soportar estos cambios resulta clave para el éxito de las organizaciones (Wheelen & Hunger, 2012).

Algunas áreas, por ejemplo, Computación Autónoma (Cetina, Giner, Fons & Pelechano, 2009), Líneas de Producto Software (Pol'la, Buccella, Cechich & Arias, 2014), y Modelado de Procesos de Negocio (Santos, Castro, Sánchez & Pastor, 2010), aplican el análisis de variabilidad para hacer frente los cambios que se producen en el contexto.

Se conoce como variabilidad a la habilidad de un proceso o un producto de ser extendido, personalizado o configurado para ser reutilizado bajo un determinado contexto, cuya detección involucra identificar qué puede variar, porqué y cuáles son las variantes posibles (Schnieders & Puhlmann, 2006; Hallerbach, Bauer & Reichert, 2008).

La variabilidad podría interpretarse como adecuarse a situaciones especiales, en lugar de seguir un proceso rígidamente (o en forma autista) (Galster, Weyns, Tofan, Michalik & Avgeriou, 2014). Es decir, permite planificar situaciones posibles, evitando improvisaciones durante la realización del proceso.

Existe un gran número de trabajos de investigación relacionados a la gestión de la variabilidad, que ofrecen numerosas propuestas para dominios específicos. A continuación, se presenta una breve reseña de algunas de sus aplicaciones, limitándose a aquellas que han sido las más ampliamente estudiadas.

Computación Autónoma

En virtud de la utilización de la Computación Autónoma (Cetina, Giner, Fons & Pelechano, 2009) se puede reducir la complejidad de los procesos mediante mecanismos de autoadaptación en función de las variaciones del contexto de ejecución. La esencia de este paradigma es la autogestión de los sistemas informáticos, mediante esta se libera a los administradores de sistemas de los detalles de su funcionamiento y mantenimiento, proporcionando a los clientes un sistema fiable que se encuentre operativo siempre a máximo rendimiento. Un sistema autónomo debe ser capaz de gestionarse y adaptarse dinámicamente a los cambios de situación conforme a las políticas y objetivos del negocio. Estos ambientes son capaces de auto-configurarse, auto-recuperarse e incluyen la auto-optimización y la auto-protección (IBM, 2005).

Modelado de Procesos de Negocio

El modelado de procesos de negocio, el cual es conocido también como BPM (Business Process Modelling), captura las actividades que realiza una organización para lograr un objetivo de negocio en particular, describe las actividades, restricciones de ejecución, recursos requeridos e información procesada para cada proceso modelado (Hallerbach, Bauer & Reichert, 2008).

Existe una gran cantidad de trabajos que introducen los conceptos relacionados a la representación de la variabilidad en los modelos de procesos de negocio (Schnieders & Puhmann, 2006; Montero, Peña & Ruiz-Cortés, 2008; Groefsema, Bulanov & Aiello, 2011). Bajo este enfoque el análisis de la variabilidad consiste en la definición de caminos alternativos de ejecución en un flujo de trabajo. En otros términos, es posible indicar qué partes del proceso de negocio permanecen abiertos al cambio, o no totalmente definidos, con el fin de soportar diferentes versiones del mismo proceso, personalizado, dependiendo del uso proyectado o el contexto de ejecución (Santos, Castro, Sánchez & Pastor, 2010). Para ello, es necesario identificar, por una parte, las variaciones posibles del proceso y, por otra, qué características propias del contexto en que este será aplicado determinan la elección apropiada de estas variantes. En el trabajo de Santos, Castro, Sanchez & Pastor (2010) se sintetiza esto en 2 etapas:

- 1) La elicitación y representación de la variabilidad en el modelo, lo cual implica descubrir una lista de probables variaciones en el proceso. El resultado podría implicar incluir, cambiar o excluir alguno de sus elementos. Posteriormente, cada uno de estos caminos se debe modelar para reflejar los posibles desvíos para llevar a cabo dicho proceso.
- 2) El análisis y configuración, que involucra asociar las variantes obtenidas en la fase anterior a las causas o factores que le dan origen, y definir mecanismos que permitan llevar a cabo la evaluación y selección de las alternativas.

Líneas de Producto Software

Una característica importante en la Ingeniería de Software es la reutilización, tanto en el diseño, en la implementación de nuevos sistemas, como así también en la producción de requisitos. Existe, en este sentido, un gran número de investigaciones que apuntan a reducir el esfuerzo y tiempo en el desarrollo de

software, mediante las líneas de producto software (McGregor, 2004; Pol'la, Buccella, Cechich & Arias, 2014; Arias, Buccella & Cechich, 2017). Bajo este paradigma se pretende agilizar el proceso de producción a través de la identificación y reutilización de elementos comunes que los productos comparten, conformando así una familia de productos; esto es posible ya que considera a cada sistema individual como una variación o derivación de una base común (Thiel, 2002; Schmid, 2004).

En otras palabras, una línea de producto se puede definir como un grupo de sistemas software que comparten un conjunto de características comunes las cuales satisfacen las necesidades específicas de un dominio o segmento particular de mercado, y que se desarrollan a partir de un sistema común de activos base de una manera preestablecida (Clements & Northrop, 2002).

Existen distintas técnicas para poner en práctica esta idea. Específicamente, la aproximación Base-Variación-Resolución propone la construcción de tres modelos: un modelo base, un modelo de variación y un modelo de resolución, para definir de forma explícita las partes del modelo que son fijas, las que pueden cambiar, y las condiciones que harán que estas cambien (Haugen & Øgård, 2014).

Líneas de Proceso Software

Basándose en la visión de líneas de productos y familias de productos, pero aplicado al dominio de los procesos, se da sustento al concepto de líneas de procesos y familias de procesos software.

Las líneas de procesos software se establecen en base a las semejanzas y diferencias de sus procesos; en este sentido, se denomina familia de procesos al conjunto de procesos, o diferentes variantes de un mismo proceso, que pueden ser gestionados como una única línea de proceso (Schnieders & Puhlmann, 2006; Simidchieva, Clarke & Osterweil, 2007).

Pensar los procesos desde esta perspectiva puede ser una buena estrategia para facilitar un ajuste sistemático del proceso, incluyendo la reutilización y evolución de forma planificada. Lo que se pretende es facilitar la adaptación mediante el intercambio de activos, que podrán ser bloques de proceso o de producto, dependiendo de las circunstancias de aplicación (Grenon, 2009).

Por lo anterior, se hace evidente que la variabilidad es un factor clave que puede aplicarse a la mayoría, sino a todos, los aspectos del desarrollo de software en general (Chen & Babar, 2010). En el dominio de la Ingeniería de Software, se refiere a la posibilidad de cambios en modelos y productos software (Puhlmann, Schnieders, Weiland & Weske, 2005).

Cuando la variabilidad es aplicada en los diferentes procesos del desarrollo de software puede ofrecer los siguientes beneficios (Simidchieva, Clarke & Osterweil, 2007):

- Mejorar los esfuerzos de coordinación, de automatización del proceso y el entrenamiento del equipo que participa en el desarrollo del software.
- Favorecer la reutilización cuando sea necesario definir nuevos procesos.
- Facilitar la adaptación de los procesos mediante el intercambio de sus elementos dependiendo de los contextos de aplicación.

Adaptar sistemáticamente un proceso comprende identificar adecuadamente, por un lado, las posibles alternativas de variación del proceso y, por otro, los aspectos del contexto de aplicación que determinan la selección apropiada de estas variaciones.

Con tal finalidad, se han propuesto en la literatura distintas técnicas para modelar la variabilidad que van desde diagramas de actividad del Lenguaje Unificado de Modelado (Unified Modeling Language -UML) (Razavian & Khosravi, 2008), hasta otros más específicos tales como la notación para el Modelado de Procesos de Negocio (Business Process Modeling and Notation -BPMN) (Delgado & Calegari, 2017), la Especificación de Recursos Reutilizables (Reusable Asset Specification -RAS) (Pacini & Braga, 2015) que define de manera estándar toda la información asociada a un activo o recurso incluyendo las guías necesarias para facilitar la manipulación y reutilización del mismo, o el Metamodelo de Ingeniería de Procesos de Software (Software Process Engineering Metamodel -SPEM) (Martínez-Ruiz, García, Piattini & Munch, 2011) que especifica un lenguaje para modelar familias de procesos de desarrollo de software, entre otros, pudiéndose citar Puhlmann, Schnieders, Weiland & Weske (2005), Hallerbach, Bauer & Reichert (2008) y Simmonds, Bastarrica, Silvestre & Quispe (2011).

En resumen, los distintos autores coinciden en que la gestión de la variabilidad incluye la definición de:

- los puntos de variación a detectar en el proceso, que corresponden a dónde los elementos del proceso pueden cambiar;
- las posibles resoluciones o flujos variantes, que son las partes del proceso que representan la variabilidad e implementan los puntos de variación; y
- el mecanismo de variabilidad a usar, que puede ser de distinto tipo: parametrización, adición, omisión, reemplazo de un elemento simple y variabilidad de tipo de dato.

Por otra parte, tal como se mencionó anteriormente, una contribución interesante que surge de trabajos relacionados a las líneas de producto software es la aproximación Base-Variación-Resolución (Haugen, Moller-Pedersen, Odelvik, Olsen & Svendsen, 2008), la cual puede aplicarse independientemente del lenguaje o notación seleccionada para modelar la variabilidad. Desde esta perspectiva lo que se propone es la construcción de tres modelos:

1. *Modelo base*, es aquel donde se especifican los elementos comunes a todas las variantes del proceso. Adicionalmente, se identifican los puntos precisos del propio modelo que están sujetos a cambios.
2. *Modelo de variación*, aquel que detalla las diferentes alternativas que pueden corresponder a cada punto sujeto a cambios que se han identificado en el modelo base. Así, cada elemento del modelo puede corresponderse con opciones específicas de variación que pueden implicar, por ejemplo, adición, reemplazo, omisión, u otro mecanismo (Puhlmann, Schnieders, Weiland & Weske, 2005).
3. *Modelo de resolución*, es aquel que permite identificar y seleccionar la alternativa que debe aplicarse a cada punto sujeto a cambio del modelo base de acuerdo con las condiciones del contexto que determinan dicha selección.

En adición a lo anterior, Ayora, Alférez, Torres & Pelechano (2011) proponen, además de estos tres modelos, la definición de un modelo de contexto, el cual permite recoger, formalizar y analizar toda la información asociada al contexto en el que será ejecutado el proceso.

2.5 Ingeniería de Métodos Situacional

La Ingeniería de Métodos es una disciplina que permite establecer nuevos métodos¹⁰ a partir de otros ya existentes, con un enfoque ordenado y sistemático. (Maynard, 1991). Se trata de un procedimiento metódico en el cual se analizan todas las operaciones detalladamente para registrarlas, mejorarlas, estandarizarlas y convertir el trabajo en una actividad más sencilla y fácil. Se considera su aplicación como clave a la hora de acrecentar los índices de producción en una organización, y el perfeccionamiento y la estandarización de sus procesos (Freivalds & Niebel, 2013). En otras palabras, consiste en analizar detalladamente cada fase o etapa de un proceso con el objetivo de eliminar toda actividad que no se considere necesaria, buscando alcanzar un proceso estandarizado que permita optimizar la productividad.

Una rama importante que ha surgido dentro de la Ingeniería de Métodos es la Ingeniería de Métodos Situacional (IMS), cuyo objetivo es la construcción de métodos específicos para llevar a cabo un proyecto, que se ajusten a las situaciones particulares de su contexto, de modo flexible, adaptable, ágil y repetible (Kumar & Welke, 1992). Se enfoca en el diseño, construcción y adaptación de métodos, técnicas y herramientas de soporte para el desarrollo de sistemas de software (Brinkkemper, 1996; Rolland, 2008).

Siguiendo los principios de esta disciplina, la adaptación de un proceso de desarrollo de software se basa en indicadores que en su conjunto definen una situación (Khan, bin Mahrin & bt Chuprat, 2014). Se entiende como situación a la combinación de circunstancias en un momento dado y en un contexto específico, que afecta la forma de trabajo y los productos a generar. Una situación puede definirse a partir de diversos factores, tanto internos como externos, que afectan el comportamiento y la eficiencia organizacional. Por lo tanto, se debe considerar que cada situación es una combinación del contexto de aplicación y de las

¹⁰ Dentro de la Ingeniería de Software, es posible definir un método como una serie de instrucciones y reglas a seguir de modo estructurado, para llevar a cabo una determinada actividad o conjunto de actividades, enmarcadas dentro de las etapas del ciclo de vida de un producto software (Brinkkemper, 1996; Pressman, 2010).

características propias del proyecto de software (Bucher, Klesse, Kurpjuweit & Winter, 2007).

Según Brinkkemper (1996), un proceso se compone de fragmentos o bloques más pequeños, que se caracterizan por tener un objetivo, el que puede ser alcanzado por una o más actividades que se realizan aplicando una o más técnicas incluidas en ese fragmento y que pueden generar uno o más artefactos. Naturalmente esto incluye además la existencia de bloques de producto, e incluso bloques que agrupan tanto proceso como producto (Rolland, 2008; Ralyté, 2013). En este mismo orden de ideas, algunos de dichos bloques son comunes a todas las situaciones mientras que otros son variantes de acuerdo a distintos factores de situación. Por lo tanto, el proceso se conforma con la unión de ciertos bloques para una situación particular (Henderson-Sellers & Ralyté, 2010).

En la literatura se utilizan diferentes términos para hacer referencia a aquellos elementos que sirven de base a los enfoques para la creación de métodos en la IMS (Kumar & Welke, 1992; Brinkkemper, 1996). Al hablar de métodos se incluyen dos perspectivas relacionadas entre sí, procesos y productos. Un proceso describe cómo construir un producto, y un producto a su vez permite describir un sistema desde diferentes niveles de abstracción (Henderson-Sellers, Ralyté, Ågerfalk & Rossi, 2014).

Algunos autores definen “fragmento de método” (*method fragment*) de modo análogo a lo que sería la noción de un componente de software. Bajo este enfoque, en general, se hace una diferencia entre dos tipos de fragmentos o bloques: bloques de proceso y bloques de producto (Harmsen, Brinkkemper & Oei, 1994; Brinkkemper, 1996; Rolland, 2008). Otros en cambio, utilizan el término “trozo de método” (*method chunk*), para referirse a la combinación de un bloque de proceso con bloques de producto, es decir, este punto de vista considera que un fragmento de proceso incluye todos los elementos de productos necesarios, tanto de entrada como de salida, para la ejecución del bloque de proceso asociado (Ralyté, Deneckère & Rolland, 2003; Rolland, 2008).

La ventaja que conlleva el primer enfoque (*method fragment*) con relación al segundo (*method chunk*), es que brinda una mayor flexibilidad en el proceso de

adaptación, ya que varios bloques de proceso podrían asociarse a los mismos bloques de producto (Rolland, 2008; Henderson-Sellers & Ralyté, 2010).

En resumen, la IMS no solo considera el desarrollo de componentes de proceso sino también componentes de producto e incluso componentes constituidos por el ensamble de procesos y productos.

La Tabla 2-2 presenta los enfoques, en el marco de IMS, más utilizados para la construcción de métodos presentados en el trabajo de Henderson-Sellers, Ralyté, Ågerfalk & Rossi (2014).

Tabla 2-2. Enfoques para la construcción de métodos

Enfoque	Descripción
Ad-Hoc	Se aplica este enfoque cuando por determinadas razones se debe construir un nuevo método 'desde cero', ante una situación que no se puede resolver con un método existente.
Basado en Ensamblaje	Es un enfoque basado en la reutilización. Un nuevo método se construye tomando fragmentos de métodos ya existentes. Los fragmentos o bloques pueden ser clasificados en bloques de producto y bloques de proceso.
Basado en la Extensión	El método requerido es creado a partir de un método ya existente, el cual se extiende haciendo uso de patrones y completando las falencias del método original.
Basado en Paradigma	Este enfoque se fundamenta en el concepto que un nuevo método, que se obtiene a partir de una abstracción de un método existente. Se le llama también enfoque basado en la evolución.
Diagramas de Actividad de UML	Estos diagramas resultan útiles para representar el flujo de trabajo del modelo de proceso una vez que se han seleccionado los fragmentos o bloques del método. Este enfoque ofrece menos soporte para la selección y el ensamblaje de los fragmentos, se asigna más responsabilidad a la experiencia del ingeniero de métodos.
Basado en Configuración	Este enfoque apunta a la posibilidad de obtener configuraciones de métodos reutilizables que puedan adaptarse a características del desarrollo específicas, introduce conceptos de paquete de configuración y plantilla de configuración, los cuales son clasificaciones de acciones prescritas en el método base.

Independientemente del enfoque a utilizar, la creación o adaptación de un método a partir del ensamblado de bloques en IMS se basa en tres pasos generales (Brinkkemper, 1996; Saeki, 2003; Ralyté, Deneckére & Rolland, 2003):

- Caracterizar el proyecto, es decir, definir el contexto o situación, de modo tal que se puedan identificar los requisitos del método a construir.
- Seleccionar los fragmentos existentes que conformarán el método.
- Ensamblar los fragmentos de modo tal que se obtenga el método para ese proyecto específico.

Desde esta perspectiva, se asume que el método base con los fragmentos de métodos ya existen y si no existieran deben ser creados primero. En el proceso de IMS propuesto en Henderson-Sellers, Ralyté, Agerfalk & Rossi (2014), el ingeniero de métodos, en función de los factores situacionales y las guías de construcción provistas para ello, selecciona los bloques del método base, y construye un proceso específico para la organización, que luego puede ser adaptado por el director de proyecto, para cada proyecto en particular.

En conclusión, se puede notar que la IMS no se orienta a la adquisición de métodos listos para usar, sino que su interés se centra en la construcción de un método específico para llevar a cabo un proyecto. Es decir, abarca todos los aspectos de la creación de un método de desarrollo para una situación determinada a partir de partes de métodos existentes, es decir, ensamblando distintos fragmentos o bloques que conforman dicho método (Kumar & Welke, 1992; Brinkkemper, 1996).

Entre los beneficios que la IMS aporta al proceso de construcción de sistemas software, según Harmsen, Brinkkemper & Oei (1994), se pueden mencionar los siguientes:

- *Flexibilidad.* Se presupone que el método que se utilizará en un determinado proyecto es situacional, es decir, está completamente sintonizado con las características del proyecto en cuestión, lo que da como resultado un enfoque lo más flexible posible.
- *Acumulación de experiencia.* La adaptabilidad completa pero controlada del método permite la adición de la experiencia del proyecto. Esto se puede lograr adaptando los bloques de construcción del método, que se almacenan en un repositorio común. Esta experiencia puede ser reutilizada en nuevos proyectos.
- *Calidad.* El hecho de que la flexibilidad debe controlarse garantiza que el método situacional construido cumple con los mismos requisitos de calidad que los métodos estándar. La estandarización de los bloques de método facilita la medición de características importantes del proyecto.

Contrastando el enfoque de la IMS con el de variabilidad de procesos, que se ha analizado en el apartado 2.4, se detecta que existen ciertas semejanzas, como por

ejemplo, ambos favorecen la adaptabilidad en base a características particulares y la reutilización. Sin embargo, se observa que la IMS presenta una visión más general, dirigida a la construcción de un método para una situación determinada, a partir de fragmentos o bloques que existen o pueden ser creados, en cambio, la variabilidad parte de un proceso base ya existente y se avoca a establecer sus variantes posibles o caminos alternativos en función de ciertas condiciones que pueden darse al aplicar dicho proceso.

2.5.1 Factores Situacionales

Tal como se ha indicado anteriormente, una situación surge de la combinación de determinadas condiciones en un momento dado, e incide en la elección del método de trabajo a seguir y los artefactos que se deben producir. Dichas condiciones se hacen evidentes a partir de diversos factores, que pueden ser tanto internos como externos, por lo cual, se debe considerar que cada situación se compone de características propias del contexto de aplicación y del proyecto en cuestión (Bucher, Klesse, Kurpjuweit & Winter, 2007).

Efectivamente, esto aplica también a la IR; frecuentemente en la literatura, se propone el uso de un método específico para definir los requisitos del software. Sin embargo, existen muchos factores que deberían ser considerados en el momento de definir los mismos, tales como el grado de cambio esperado en los procesos del negocio, el conocimiento previo del dominio de aplicación, la complejidad de dicho dominio, las características de los involucrados y la envergadura del proyecto de software, entre otros muchos.

Con base en la propuesta de la IMS, la cual se avoca a la construcción de métodos ajustados a situaciones específicas para el desarrollo de software (Kumar & Welke, 1992), se han estudiado diversos factores considerados en la literatura para distintas actividades de la IR (Hadad & Doorn, 2013; Hadad, Ledesma & Doorn, 2014), los que en principio, se estima podrían considerarse para determinar la necesidad de adaptar la estrategia.

Algunos de estos factores se enfocan en atributos que definen el contexto en el que el proceso de IR debe ser aplicado. Estos involucran aspectos relacionados a las fuentes humanas de información, considerando por ejemplo su experiencia,

variables de personalidad y disponibilidad (Burton, Shadbolt, Rugg & Hedgecock, 1990; Dhaliwal & Benbasat, 1990; Van de Weerd, Brinkkemper & Versendaal, 2010; Carrizo Moreno, 2009); cuestiones relacionadas al dominio del problema al que se intenta dar solución, tales como su complejidad y volatilidad (Shadbolt & Burton, 1989; Hadad, 2008; Jafarinezhad & Ramsin, 2012); mientras que otros factores tienen que ver con el producto software en sí mismo, considerando su nivel de innovación, grado de criticidad e importancia estratégica, entre otros (Hadad, 2008; Jafarinezhad & Ramsin, 2012).

Por otra parte, hay varios factores concernientes al proyecto de software, que incluyen atributos referidos al equipo que realiza las tareas de IR, considerando cuestiones como por ejemplo su conocimiento del dominio, su experiencia en el proceso de IR y su distribución geográfica respecto a los clientes (Lloyd, Rosson & Arthur, 2002; Ebling, Nicolas Audy & Prikladnicki, 2009); y también los que tienen que ver con particularidades del proyecto como su envergadura, restricciones de tiempo y estándares exigidos (Maiden & Rug, 1996; Jafarinezhad & Ramsin, 2012).

2.6 Adaptabilidad en la Ingeniería de Requisitos

Las estrategias de IR rara vez se adaptan a su contexto (Potts, 1995; Leffingwell & Widrig, 2003; Leite, Doorn, Kaplan, Hadad & Ridaou, 2004; Seyff et al., 2009), en general, presentan un enfoque top-down, bottom-up y, eventualmente middle-out en la construcción de modelos, indicando heurísticas de elicitación y técnicas de verificación y/o validación a utilizar. Aun así, han surgido diversas propuestas para atenuar los problemas detectados en la creación y en la mejora de un proceso de IR, adaptándolo a las características particulares de un proyecto. Los trabajos existentes pueden agruparse de la siguiente manera:

1. *Orientados a definir un proceso de IR utilizando frameworks, ya sea que estos existan en el mercado o son propuestos por los propios autores.* En estos trabajos se pone a disposición de los ingenieros de requisitos una base de conocimiento sobre actividades y técnicas de IR, que sirve de guía y apoyo para obtener un proceso de IR acorde a las necesidades del proyecto. En ciertos casos la conveniencia de los elementos que se deben seleccionar se define en función de atributos propios del proyecto, por ejemplo, Jiang & Eberlein (2008)

brindan algunas condiciones a evaluar, mientras que, otros autores dejan la decisión librada a los ingenieros de requisitos, por lo tanto, el resultado dependerá de la experiencia y la habilidad de estos (Olsson, Doerr, Koenig & Ehresmann, 2005; Zowghi, Firesmith & Henderson-Sellers, 2005). Desde esta misma perspectiva, Firesmith (2004) ofrece pautas para elegir un framework, de los existentes en el mercado, que permita crear y dar soporte al proceso de IR en base a factores que caracterizan al proyecto, tales como la cultura organizacional, el involucramiento de los stakeholders, y la compatibilidad con las herramientas disponibles en la organización, entre otros. A pesar del uso de algunos atributos relacionados con el proyecto, estos autores no dan criterios claros de aplicación de los mismos, o solo presentan pautas de selección muy generales. Aunque estos trabajos no lo mencionan de modo explícito, podrían asociarse con el enfoque basado en ensamblaje de la IMS, ya que todas estas propuestas se fundamentan en la reutilización, y el proceso se define tomando elementos de métodos ya existentes.

2. *Enfoques que contribuyen a seleccionar las técnicas de elicitación de información que mejor se adaptan al contexto situacional.* Los trabajos que se han incluido en este grupo tienen como objetivo aportar guías para la selección de las técnicas que se aplican en la etapa de elicitación de requisitos, es decir, solo se enfocan en dicha actividad; esto marca una diferencia con los demás trabajos que se han analizado los cuales se refieren al proceso de IR completo. Hay numerosos trabajos orientados a este enfoque, probablemente debido a la gran cantidad de diferentes técnicas disponibles para ser seleccionadas (Oliveros & Antonelli, 2015). En este sentido, algunos autores (Maiden & Rug, 1996; Hickey & Davis, 2003; Coulin, 2007; Carrizo, Dieste & Juristo, 2008; Al-Zawahreh & Almakadmeh, 2015), toman en cuenta determinados factores situacionales, o características contextuales, como el número de fuentes de información, la distribución geográfica del usuario, la disponibilidad de tiempo del usuario, y su experiencia en el dominio, entre otras, con el fin de determinar las técnicas de recolección de hechos más apropiadas.
3. *Modelos que permiten adaptar el proceso de IR cuando este se aplica a una clase específica de proyectos.* Se trata de propuestas para la adaptación situacional del proceso a partir de la identificación de los factores que tienen

mayor influencia en las actividades de IR. Se aplica para proyectos de determinada naturaleza o alcance, por ejemplo, un proyecto para el desarrollo global de software, o mediante un método ágil, entre otros (Souer, van de Weerd, Versendaal & Brinkkemper, 2005; Abdullah & Khan, 2015; Bakhat, Sarwar, Motla & Akhta, 2015; Kabaale & Mayoka Kituyi, 2015; Khan, bin Mahrin & Mali, 2016). El propósito que se persigue es optimizar la producción y la gestión de requisitos, así como minimizar los defectos repetitivos detectados en proyectos de estos tipos. Además de proponer una adaptación general del proceso de IR en base a las características propias de la naturaleza del proyecto, también se toman algunos principios de la IMS para flexibilizar el proceso de acuerdo a otros atributos contextuales.

4. *Guías que facilitan la creación de un proceso de IR adaptado al contexto de aplicación.* Se trata de trabajos que aportan mecanismos para la creación de un proceso de IR ajustado a las características situacionales del entorno donde será utilizado y que, a diferencia de los trabajos agrupados en el punto 3, podrían aplicarse en proyectos de diferente naturaleza. En este sentido han surgido algunas propuestas para diseñar un proceso, basándose en la selección de diversas técnicas existentes que pueden ser utilizadas en las distintas etapas del proceso (Lauesen, 2002; Alexander & Beus-Dukic, 2009). Tal es el caso de Lobo & Arthur (2005), que en su trabajo consideran cada actividad del proceso de IR e identifican los métodos que comúnmente se utilizan para llevar adelante dichas actividades, analizan la conveniencia de cada método en base a determinados criterios, tales como el costo, el tiempo y el personal disponible, y de esta manera el ingeniero de requisitos puede tomar una decisión sobre el método más apropiado para una actividad. Hickey & Davis (2004) tienen una visión más dinámica del proceso, dado que proponen un modelo de IR que enfatiza el conocimiento que se va adquiriendo a medida que avanza el proceso, acerca del dominio del problema, del dominio de la solución y del dominio del proyecto; se obtiene un modelo flexible que permite, en cada iteración mediante una formulación matemática, definir las etapas y técnicas más adecuadas según el estado de los elementos situacionales descritos. Asimismo, otros autores (Coulin, Zowghi & Sahraoui, 2006; Jafarinezhad & Ramsin, 2012) han aplicado algunos principios de la IMS para definir procesos de requisitos adaptables

mediante el uso de componentes modulares existentes, incluso en combinación con el paradigma de la variabilidad de procesos, aplicando algunos conceptos de las líneas de producto software.

Como se puede notar, en todos los trabajos considerados, se aplica en mayor o menor medida algunas nociones de la IMS. En general, las propuestas de las cuatro agrupaciones presentan determinados factores situacionales que son evaluados con el objetivo de definir el contexto; particularmente, en los grupos 1 y 4 se han encontrado enfoques basados en la reutilización de componentes para conformar el nuevo proceso de IR. Por su parte, solo en los trabajos del grupo 4, existen algunas propuestas que combinan los conceptos de la IMS con algunos principios de la variabilidad de procesos. Cabe mencionar que en la gran mayoría de los estudios relacionados con la IMS y la variabilidad no se aborda un aspecto esencial relacionado con la probabilidad de evolución o de estimación incorrecta de los factores que definen esa situación, es decir, consideran a la misma de manera estática para formular el proceso.

La propuesta que se presenta en esta tesis podría incluirse en el grupo 4 debido a que, tal como se mencionó en la Introducción (capítulo 1), tiene por finalidad establecer mecanismos claros para crear y adaptar una estrategia de IR de acuerdo con las características particulares del contexto en el que esta deba ser aplicada.

Capítulo 3

Factores Situacionales Comprometidos en la Estrategia de Requisitos

3.1 Determinación del Conjunto de Factores Situacionales

Tal como se mencionó en el planteamiento del problema, la propuesta de adaptación del proceso de IR comprende en primer término, la identificación del conjunto de factores observables que determinan la situación particular en la que dicho proceso será aplicado. Posteriormente, en base a dichos factores se adaptará la estrategia de requisitos, definiendo qué actividades del proceso de requisitos se deben realizar, qué técnicas específicas conviene utilizar, y qué artefactos de requisitos deben construirse.

Como resultado del estudio realizado, como parte de esta tesis, se han identificado en la literatura 40 factores con posible influencia en el proceso de IR, algunos de los cuales emergen de la aplicación de la estrategia de IR. Estos se han plasmado en las Tablas 3-1 y 3-2, en las cuales se listan 23 factores relacionados al contexto de aplicación y 17 factores asociados al proyecto de software, indicándose los autores que los tratan. Como se puede notar, en algunos casos, se presentan distintas denominaciones para factores similares o muy próximos en su definición, por lo cual se los trata como un único ítem, respetándose la denominación dada por los mismos¹¹. Debe notarse que algunos factores ya han sido propuestos por los propios autores de la estrategia de IR objeto de adaptación (Hadad, 2008; Hadad & Doorn, 2013; Mighetti & Hadad, 2015).

Al examinar estos factores, se detecta que algunos de ellos no afectan directamente al proceso de requisitos en general. Otros se han seleccionado como factores influyentes en la estrategia de IR; en algunos casos se incorporaron tal

¹¹ Cabe mencionar una divergencia con el uso del término “informante” en algunos factores, dado que el mismo puede influenciar la selección de la fuente de información. Por otro lado, su uso en singular pareciera suponer una única fuente humana de información, lo cual no es el caso más frecuente ni el mejor esperado en un proceso de IR.

cual han sido propuestas en la literatura o con variaciones menores. Adicionalmente, se ha detectado la necesidad de agregar algunos factores que, aunque no son mencionados en la bibliografía, surgieron durante el desarrollo de la propuesta de adaptación del proceso.

Tabla 3-1. Factores situacionales relacionados al contexto de aplicación

#	Nombre del Factor	Autores
1	Nivel de experiencia del informante	- Burton, Shadbolt, Rugg & Hedgecock (1990) - Dhaliwal & Benbasat (1990) - Van de Weerd, Brinkkemper & Versendaal (2010)
2	Capacidad de comunicación del informante	- Dhaliwal & Benbasat (1990) - Carrizo Moreno (2009)
3	Variables de personalidad del informante	- Dhaliwal & Benbasat (1990)
4	Grado de resistencia del informante	- Jafarinezhad & Ramsin (2012)
5	Problemas cognitivos del informante Habilidades cognitivas del informante	- Byrd, Cossick & Zmud (1992) - Chao & Salvendy (1995)
6	Localización del informante en relación con el equipo de IR Involucrados distribuidos geográficamente	- Carrizo, Dieste & Juristo (2008) - Ebling, Nicolas Audy & Prikladnicki (2009) - Mighetti & Hadad (2015)
7	Tipo de cliente Tipo de producto	- Keil & Carmel (1995) - Hadad (2008) - Gorschek (2011)
8	Nivel de rotación de usuarios	- Hadad (2008)
9	Participación del usuario Disponibilidad de tiempo del informante	- Lloyd (2001) - Carrizo, Dieste & Juristo (2008) - Jafarinezhad & Ramsin (2012)
10	Tipo de fenómeno	- Maiden & Rug (1996)
11	Tipo de información	- Maiden & Rug (1996)
12	Campos de dominio Dominio de la aplicación	- Shadbolt & Burton (1989) - Jafarinezhad & Ramsin (2012)
13	Grado de estructura percibida del dominio Complejidad del contexto	- Kim & Courtney (1988) - Dhaliwal & Benbasat (1990) - Hadad (2008) - Jafarinezhad & Ramsin (2012)
14	Tipo de tareas a elicitar	- Dhaliwal & Benbasat (1990) - Chao & Salvendy (1995) - Wagner, Chung & Najdawi (2003)
15	Volatilidad de los requerimientos Dominio con cambio constante	- Hadad (2008) - Jafarinezhad & Ramsin (2012)
16	Grado de conflicto en el dominio Grado de confusión	- Fazlollahi & Tanniru (1991) - Jafarinezhad & Ramsin (2012)
17	Grado de reingeniería del proceso de negocio	- Hadad (2008)
18	Novedad del negocio	- Hadad (2008)
19	Importancia estratégica	- Jafarinezhad & Ramsin (2012)
20	Número de usuarios	- Maiden & Rug (1996)
21	Compromiso de la dirección de la organización cliente	- Jafarinezhad & Ramsin (2012)
22	Impacto organizacional del proyecto	- Jafarinezhad & Ramsin (2012)
23	Grado de definición del problema	- Carrizo Moreno (2009)

Tabla 3-2. Factores situacionales relacionados al proyecto de software

#	Nombre del Factor	Autores
24	Tipo de proyecto	- Jafarinezhad & Ramsin (2012)
25	Envergadura del proyecto	- Hadad (2008) - Jafarinezhad & Ramsin (2012)
26	Grado de reusabilidad de los artefactos	- Jafarinezhad & Ramsin (2012)
27	Nivel de madurez de los procesos de IR	- Jafarinezhad & Ramsin (2012)
28	Restricciones al proyecto	- Maiden & Rug (1996)
29	Estándares de especificación exigidos	- Hadad (2008)
30	Nivel de rotación del grupo desarrollador	- Hadad (2008)
31	Escasez de personal y recursos	- Jafarinezhad & Ramsin (2012)
32	Momento del proceso	- Carrizo Moreno (2009)
33	Método de desarrollo de software	- Dhaliwal & Benbasat (1990)
34	Experiencia en IR Conocimiento en técnicas de educación Experiencia en métodos de educación	- Agarwal & Tanniru (1990) - Dhaliwal & Benbasat (1990) - Lloyd, Rosson & Arthur (2002) - Jafarinezhad & Ramsin (2012)
35	Familiaridad con el dominio Grado de conocimiento de los requisitos Conocimiento previo del dominio	- Hadad (2008) - Jafarinezhad & Ramsin (2012)
36	Diferencias temporales	- Ebling, Nicolas Audy & Prikladnicki (2009) - Mighetti & Hadad (2015)
37	Diferencias idiomáticas y/o culturales entre los involucrados	- Ebling, Nicolas Audy & Prikladnicki (2009) - Mighetti & Hadad (2015)
38	Nivel de innovación esperada del proyecto y del proceso	- Jafarinezhad & Ramsin (2012)
39	Nivel de información disponible antes de aplicar la técnica de elicitación	- Carrizo Moreno (2009)
40	Nivel de criticidad	- Jafarinezhad & Ramsin (2012)

A continuación, se presenta el análisis realizado para obtener el conjunto de factores situacionales que permita definir la situación de cada proyecto para posteriormente realizar los ajustes necesarios en el proceso de requisitos.

3.1.1 Análisis de los Factores

Se ha examinado el enfoque de los autores respecto a cada uno de estos factores, evaluando en qué etapas de la estrategia de IR se considera que el factor podría provocar algún tipo de adaptación en el proceso (los detalles del análisis realizado se presentan en el Anexo A).

Algunos factores han sido identificados por los autores de la estrategia durante la aplicación práctica de la misma, y se han incorporado con las siguientes denominaciones: *tipo de cliente, rotación de usuarios, rotación del equipo desarrollador, complejidad del contexto, volatilidad del contexto, volatilidad de los requerimientos del cliente, reingeniería en el proceso de negocio y novedad del*

contexto (Hadad, 2008; Hadad & Doorn, 2013). Así también el trabajo presentado por Mighetti & Hadad (2015), relacionado al desarrollo global de software dio origen a la inclusión de los factores: *localización de usuarios, diferencias idiomáticas y/o culturales, y diferencias temporales entre los involucrados*.

Por otra parte, durante el refinamiento de la adaptación de la estrategia se detectó la necesidad de agregar cuatro factores relacionados a aspectos de rastreabilidad y al modelo de proceso de software incremental; estos son:

- **Pre-Rastreabilidad de los requisitos:** Se refiere a la exigencia de mantener la rastreabilidad entre los requisitos y los artefactos generados con anterioridad al establecimiento de los mismos, incluyendo las fuentes de información. Todos los requisitos deben ser rastreables, con el objeto de mantener la consistencia entre los distintos artefactos de un proyecto. Si hay necesidad de rastrear los requisitos hacia atrás, esto impacta en la creación del LEL del UdeD, de los EA y del LEL del Sistema.
- **Post-Rastreabilidad de los requisitos:** Siguiendo los mismos lineamientos aplicados a la pre-rastreabilidad, este factor evalúa la posibilidad de que se requiera la rastreabilidad entre los requisitos y los artefactos generados en las etapas posteriores del desarrollo. Esto puede impactar en la especificación de los requisitos.
- **Granularidad de rastreabilidad de los requisitos:** Se trata de un aspecto estrechamente relacionado a los dos factores anteriores. Se refiere al nivel de detalle en la rastreabilidad de los requisitos. Cuando hay exigencias de pre-rastreabilidad o post-rastreabilidad puede ser que se necesite rastrear individualmente cada requisito, o por conjuntos de requisitos, por ejemplo, el primer caso implicaría la necesidad de explicitar los requisitos, mientras que en el segundo caso sería posible rastrear por EF.
- **Tipo de incremento:** Este factor surge al analizar con mayor detalle el modelo de proceso de software incremental. Cuando un proyecto se guía por tal modelo, puede suceder que se planifique un incremento para desarrollar nuevas funcionalidades, lo cual implica la creación de los artefactos de requisitos desde cero, o puede requerirse mejoras a funcionalidades que ya existen, con lo cual el proceso de requisitos consiste en impactar mejoras en artefactos existentes, y esto afecta a gran parte de las actividades de creación de modelos.

En resumen, a partir del análisis realizado algunos factores han sido desestimados, en algunos casos por no tener impacto en el proceso, y en otros debido a que el proceso ya los trata debidamente sin requerir adaptación alguna al respecto. Por otra parte, todos los factores aceptados se han seleccionado en función de su impacto para las distintas actividades del proceso de IR, dado que podrían generar algún tipo de adaptación en el mismo; y como se mencionó antes, se ha detectado la necesidad de agregar algunos factores que se consideran pertinentes para una mejor adaptación de la estrategia. Las Tablas 3-3 y 3-4 muestran de forma sintética los resultados obtenidos luego de analizar cada factor (ver más detalle en el Anexo A). La decisión que se ha tomado con respecto a cada uno de estos se ha expresado a través de uno de los siguientes estados:

- **Aceptado:** El factor se ha tomado tal cual lo propuso el autor; ha podido tener una variación menor en su denominación.
- **Descartado:** No hay suficiente justificación para mantener el factor, por lo cual no se ha tenido en cuenta para la propuesta.
- **Reorientado:** El factor ha necesitado de alguna adaptación para ser incorporado a la propuesta, ya sea en su enfoque o en su denominación para aportar claridad.
- **Desglosado:** Al analizar el factor en cuestión se ha establecido la conveniencia de separarlo en más de un factor para lograr una evaluación más específica de la situación.
- **Fusionado:** Se han detectado otros atributos que consideran características similares, por lo cual se han reunido bajo un mismo factor.
- **Agregado:** Comprende aquellos factores que han surgido en base al estudio mismo de la adaptación de la estrategia de IR.

Tabla 3-3. Análisis de factores asociados al contexto de aplicación

#	Nombre del Factor	Decisión	Justificación
1	Nivel de experiencia del informante	Descartado	Impacta no solo en la actividad de elicitación sino en toda actividad del proceso donde se requiera la intervención de los involucrados. Sin embargo, no son útiles para la adaptación del proceso en sí mismo por no provocar adaptaciones en el proceso de IR.
2	Capacidad de comunicación del informante	Descartado	Ídem ítem 1.
3	VARIABLES de personalidad del informante	Descartado	Ídem ítem 1.
4	Grado de resistencia del informante	Fusionado	Se unifica con el factor <i>Conflicto de intereses de usuarios</i> tratado en el ítem 16.
5	Problemas cognitivos del informante Habilidades cognitivas del informante	Descartado	Ídem ítem 1.
6	Localización del informante en relación con el equipo de IR Involucrados distribuidos geográficamente	Reorientado	Afecta principalmente la necesidad de construir o no el LEL del UdeD, los EA y el LEL del Sistema, y la forma de elicitar y de validar los modelos de la estrategia. Para claridad se lo denomina <i>Localización de usuarios</i> .
7	Tipo de cliente Tipo de producto	Aceptado	Afecta la construcción de los EA. Se acepta con la denominación <i>Tipo de cliente</i> .
8	Nivel de rotación de usuarios	Aceptado	Afecta la construcción o no de los EA, y afecta indirectamente la negociación de propuestas mediante EF, que el proceso ya lo maneja apropiadamente sin requerir adaptaciones. Se acepta con la denominación <i>Rotación de usuarios</i> .
9	Participación del usuario Disponibilidad de tiempo del informante	Descartado	Ídem ítem 1.
10	Tipo de fenómeno	Descartado	Afecta al proceso de IR pero este lo maneja apropiadamente sin necesidad de cambios.
11	Tipo de información	Descartado	Afecta al proceso de IR pero este lo maneja apropiadamente sin necesidad de cambios.
12	Campos de dominio Dominio de la aplicación	Descartado	No impacta directamente en la estrategia de IR.
13	Grado de estructura percibida del dominio Complejidad del contexto	Aceptado	Afecta la construcción de los EA. Se acepta con la denominación <i>Complejidad del contexto</i> .
14	Tipo de tareas a elicitar	Descartado	Solo afecta la selección de técnicas de elicitación. El proceso no requiere adaptación al respecto, pues es parte intrínseca de los bloques de proceso que involucran elicitación.
15	Volatilidad de los requerimientos Dominio con cambio constante	Desglosado	Se separa en <i>Volatilidad del contexto</i> y <i>Volatilidad de los requerimientos del cliente</i> . El primero afecta la construcción de los EA y puede afectar la definición del objetivo del sistema, y el segundo limita la posibilidad de aplicar ciertos modelos de proceso.
16	Grado de conflicto en el dominio Grado de confusión	Desglosado	Se separa en <i>Inconsistencias en el contexto</i> y <i>Conflicto de intereses de usuarios</i> . El primero afecta la construcción de los EA, y el segundo la construcción de los EF y la priorización de requisitos.
17	Grado de reingeniería del proceso de negocio	Aceptado	Afecta la construcción de los EF. Se acepta con la denominación <i>Reingeniería en el proceso de negocio</i> .
18	Novedad del negocio	Aceptado	Afecta la construcción de los EA. Se acepta con la denominación <i>Novedad del contexto</i> .
19	Importancia estratégica	Fusionado	Se unifica con el factor <i>Reingeniería en el proceso de negocio</i> tratado en el ítem 17.
20	Número de usuarios	Descartado	Solo afecta la selección de técnicas de elicitación. El proceso no requiere adaptación al respecto, pues es parte intrínseca de los bloques de proceso que involucran elicitación.
21	Compromiso de la dirección de la organización cliente	Descartado	Puede afectar la definición del objetivo del sistema y la definición de la solución en términos de cuáles serán los EF apropiados, y también puede afectar la visión del UdeD desde la definición del vocabulario en el LEL del UdeD. Sin embargo, no impacta directamente en la estrategia de IR como para requerir una adaptación específica del proceso.
22	Impacto organizacional del proyecto	Fusionado	Se unifica con el factor <i>Reingeniería en el proceso de negocio</i> tratado en el ítem 17.
23	Grado de definición del problema	Desglosado	Este factor involucra en parte a <i>Inconsistencia del contexto</i> , <i>Conflicto de Intereses</i> , <i>Volatilidad de Requerimientos</i> y <i>Volatilidad de Contexto</i> , por lo que afecta principalmente la definición del objetivo del sistema.

Tabla 3-4. Análisis de factores asociados al proyecto

#	Nombre del Factor	Decisión	Justificación
24	Tipo de proyecto	- Descartado	No impacta directamente en la estrategia de IR.
25	Envergadura del proyecto	- Aceptado	Afecta la construcción de los EA. Se acepta tal cual fue propuesto.
26	Grado de reusabilidad de los artefactos	- Desglosado	Se separa en <i>Reuso de artefactos de requisitos existentes y Creación de artefactos de requisitos para reuso</i> , Pueden afectar la creación de cualquier modelo de la estrategia de IR.
27	Nivel de madurez de los procesos de IR	- Descartado	No impacta directamente en la estrategia de IR.
28	Restricciones al proyecto	- Fusionado	Se unifica con el factor <i>Restricciones de tiempo y recursos para el proyecto</i> , tratado en el ítem 31.
29	Estándares de especificación exigidos	- Aceptado	Afecta la especificación de los requisitos. Se acepta con la denominación <i>Exigencia de producir el documento ERS</i> .
30	Nivel de rotación del grupo desarrollador	- Aceptado	Afecta la necesidad de disponer o no de los EA. Es apropiadamente manejado en el resto del proceso sin requerir su adaptación. Se acepta con la denominación <i>Rotación del equipo desarrollador</i> .
31	Escasez de personal y recursos	- Reorientado	No provoca adaptaciones en la estrategia, pero puede implicar negociaciones por incompatibilidad con otros factores. Se redefine con la denominación <i>Restricciones de tiempo y recursos para el proyecto</i> .
32	Momento del proceso	- Descartado	Solo afecta la selección de técnicas de elicitación. El proceso no requiere adaptación al respecto, pues ya es tratado como parte de la estrategia de IR.
33	Método de desarrollo de software	- Reorientado	Afecta la forma en que se aplica la estrategia. Se redefine con la denominación <i>Modelo de proceso de software</i> .
34	Experiencia en IR Conocimiento en técnicas de educación Experiencia en métodos de educación	- Aceptado	Afecta la necesidad de disponer o no de los EA para el equipo de desarrollo. También afecta las técnicas a utilizar en las distintas actividades del proceso. Se acepta con la denominación <i>Experiencia en el proceso de requisitos</i> .
35	Familiaridad con el dominio Grado de conocimiento de los requisitos Conocimiento previo del dominio	- Aceptado	Afecta la construcción o no del LEL del UdeD, de los EA y del LEL del Sistema. Se acepta con la denominación <i>Conocimiento previo del dominio</i> .
36	Diferencias temporales	- Aceptado	Afecta la construcción del LEL del UdeD, de los EA y del LEL del Sistema, y también para la forma de validar los modelos de la estrategia. Se acepta con la denominación <i>Diferencias temporales entre los involucrados</i> .
37	Diferencias idiomáticas y/o culturales entre los involucrados	- Aceptado	Afecta al proceso de requisito, principalmente la construcción del LEL del UdeD, de los EA y del LEL del Sistema. Se acepta tal cual fue propuesto.
38	Nivel de innovación esperada del proyecto y del proceso	- Descartado	No impacta directamente en la estrategia de IR.
39	Nivel de información disponible antes de aplicar la técnica de elicitación	- Fusionado	Se unifica con el factor <i>Conocimiento previo del dominio</i> , tratado en el ítem 35.
40	Nivel de criticidad	- Reorientado	Afecta en el formalismo y la rigurosidad con que se aplica la estrategia de IR. Se redefine con la denominación <i>Fiabilidad necesaria del software</i> .
41	Pre-Rastreabilidad de los requisitos	- Agregado	Afecta todo el proceso de requisitos.
42	Post-Rastreabilidad de los requisitos	- Agregado	Afecta la especificación de los requisitos.
43	Granularidad de la rastreabilidad de los requisitos	- Agregado	Afecta la especificación de los requisitos.
44	Tipo de incremento	- Agregado	Implica la construcción de artefactos desde su inicio o la modificación de los que ya existen.

3.1.2 Definición de los Factores Situacionales Seleccionados

Como resultado del análisis realizado se han identificado 26 factores situacionales que pueden provocar algún tipo de adaptación en el proceso de IR. A continuación, se da una descripción formal cada uno de dichos factores, dónde los 10 primeros factores se relacionan al contexto de aplicación, mientras que los 16 restantes tienen que ver con el proyecto en sí mismo:

- F1 - Complejidad del contexto:** Nivel de dificultad en comprender el problema bajo estudio por parte de los ingenieros de requisitos. Puede deberse a la mera complejidad técnica, de naturaleza matemática, social o de otra disciplina, a la cantidad de procesos, a la cantidad de áreas intervinientes en la organización, a interdependencias no evidentes de áreas y de actividades entre áreas, a roles y responsabilidades no precisas, entre otras.
- F2 - Tipo de cliente:** Destinatario del software, si se desarrollará a medida para un cliente específico, o si se trata de un paquete de software para un mercado de potenciales clientes.
- F3 - Novedad del contexto:** Indica si se trata de un contexto de aplicación nuevo o de uno ya existente. En otros términos, si el negocio se va a crear a partir del software a construir o ya existe.
- F4 - Reingeniería en el proceso de negocio:** Nivel de cambios esperados en los procesos del negocio a través del nuevo software, es decir, el impacto en la organización cliente o en el contexto de aplicación del sistema a construir. En algunos proyectos se espera un alto grado de cambios dado que los procesos del negocio pueden ser totalmente redefinidos o ser nuevos en función del software a construir, mientras que en otros casos hay baja reingeniería, por ejemplo, cuando solo se informatizarán tareas manuales o se integrarán tareas informatizadas aisladamente.
- F5 - Volatilidad del contexto:** Nivel de cambios que ocurren en el contexto de aplicación, ya sea por causas externas o internas al contexto. Alta volatilidad implica cambios continuos en la información elicitada y modelada.
- F6 - Volatilidad de los requerimientos del cliente:** Nivel de cambios solicitados por el cliente en cuanto a las necesidades a cubrir con el software. Alta volatilidad indica bajo nivel de madurez del cliente, es decir, cambios continuos de idea respecto a sus necesidades con poca aceptación de propuestas de los ingenieros de requisitos.
- F7 - Inconsistencias en el contexto:** Contradicciones en la información elicitada proveniente de diversas fuentes de información o en los objetivos del sistema.
- F8 - Conflicto de intereses de usuarios:** Nivel de desacuerdo o diferencias de opiniones e intereses entre los involucrados.

- F9 - Rotación de usuarios:** Nivel de cambio en los usuarios que actúan como fuente de información y que luego usarán el sistema. Esto debe ser informado por el cliente.
- F10 - Localización de usuarios:** Distribución geográfica de los usuarios, si están dispersos o en un único lugar.
- F11 - Conocimiento previo del dominio:** Familiaridad del equipo de requisitos con el dominio al cual pertenece el software a construir. Se considera la experiencia promedio del equipo de ingenieros de requisitos en participar en proyectos referidos a dicho dominio o en tener formación en el dominio. El conocimiento sobre el dominio puede ir desde niveles generales abstractos a niveles de detalle.
- F12 - Envergadura del proyecto:** Tamaño del proyecto, puede estimarse por la cantidad de miembros del equipo de ingenieros de requisitos y la duración estimada del proyecto.
- F13 - Rotación del equipo desarrollador:** Nivel de cambio en los miembros del equipo de desarrollo.
- F14 - Fiabilidad necesaria del software:** Nivel de fiabilidad esperada en los artefactos producidos en el proceso de software, en este caso en particular, en el proceso de requisitos. Por ejemplo, se requiere un muy alto nivel de fiabilidad en el caso de un software de tiempo real, de seguridad crítica o que frente a una falla pueda derivar en peligros ambientales, daño o pérdida de vidas humanas, de recursos financieros o de equipamientos.
- F15 - Reuso de artefactos de requisitos existentes:** Posibilidad de reusar en este proyecto modelos, documentos o especificaciones de requisitos, generadas en proyectos anteriores de desarrollo de software de similares características. Si aplica, se debe indicar qué artefactos se pueden reutilizar.
- F16 - Creación de artefactos de requisitos para reuso:** Exigencia de obtener productos reusables del proceso de requisitos para otros proyectos de software. Si aplica, se debe indicar qué artefactos de reuso se exige crear.
- F17 - Pre-Rastreabilidad de los requisitos:** Exigencia de mantener rastreabilidad entre los requisitos y los artefactos previos al establecimiento de los requisitos, incluyendo las fuentes de información. En caso que aplique, puede ser total si los requisitos deben ser rastreables con cada artefacto

propuesto por la estrategia de IR; puede ser parcial si se debe mantener rastreabilidad solo con aquellos creados a partir de la adaptación del proceso.

- F18 - Post-Rastreabilidad de los requisitos:** Exigencia de mantener rastreabilidad entre los requisitos y los artefactos posteriores a especificar los requisitos, es decir, rastros de cómo fue usado el requisito luego de su especificación.
- F19 - Granularidad de rastreabilidad de los requisitos:** Nivel de detalle en la rastreabilidad de los requisitos. Se considera individual cuando se necesita rastrear individualmente cada requisito, y grupal cuando alcanza con rastrear conjuntos de requisitos, por ejemplo, rastrear por escenarios.
- F20 - Exigencia de producir el documento ERS:** Si se exige la construcción de un documento de Especificación de Requisitos del Software, por ejemplo, para disponer del documento como anexo del contrato entre las partes.
- F21 - Restricciones de tiempo y recursos para el proyecto:** limitaciones en los plazos de entrega, así como en la disponibilidad de recursos humanos para el proceso de desarrollo.
- F22 - Modelo de proceso de software:** Define el uso de determinado proceso de software que impacta en la forma en que se debe entregar la documentación de requisitos. Implica, por ejemplo, entregar la totalidad de documentos de requisitos al final, entregar parcialmente o no poder aplicar el proceso de IR.
- F23 - Tipo de incremento:** Se relaciona al modelo de proceso de software incremental. Indica si se trata de una nueva funcionalidad o de una mejora a una funcionalidad existente.
- F24 - Experiencia en el proceso de requisitos:** Familiaridad del equipo de requisitos con el proceso de IR y con las técnicas a aplicar. Se mide en años de experiencia promedio del equipo de ingenieros de requisitos, en función de la experiencia individual de cada integrante.
- F25 - Diferencias idiomáticas y/o culturales:** Grado de divergencia entre los involucrados en el proyecto de software respecto al idioma que utilizan en sus ámbitos de trabajo y/o a sus culturas.
- F26 - Diferencias temporales entre los involucrados:** Grado de diferencias horarias para la comunicación entre los involucrados en el proyecto de software. Se

relaciona directamente con los husos horarios de la ubicación geográfica de los involucrados.

En la Tabla 3-5 se definen los valores admisibles para su evaluación en una situación real (ver detalles en el Anexo B). El valor “No Aplica” incluido para algunos factores se relaciona con un tipo de dependencia del factor que se describe en la siguiente subsección.

Tabla 3-5. Valoración de los factores situacionales

#	Nombre del Factor	Valores Admisibles
F1	Complejidad del contexto	Muy Alta/Alta/Media/Baja/Muy Baja
F2	Tipo de cliente	Mercado Potencial/ Cliente Específico
F3	Novedad del contexto	Sí/No
F4	Reingeniería del proceso de negocio	Muy Alta/Alta/Media/Baja/Muy Baja/No Aplica
F5	Volatilidad del contexto	Muy Alta/Alta/Media/Baja/Muy Baja
F6	Volatilidad de requerimientos del cliente	Muy Alta/Alta/Media/Baja/Muy Baja/No Aplica
F7	Inconsistencias en el contexto	Muy Alto/Alto/Medio/Bajo/Muy Bajo/No Aplica
F8	Conflicto de intereses de usuarios	Muy Alto/Alto/Medio/Bajo/Muy Bajo/No Aplica
F9	Rotación de usuarios	Muy Alta/Alta/Media/Baja/Muy Baja/No Aplica
F10	Localización de usuarios	Distribuidos/Colocalizados/No Aplica
F11	Conocimiento previo del dominio	Muy Alto/Alto/Medio/Bajo/Muy Bajo
F12	Envergadura del proyecto	Muy Grande/Grande/Mediana/Pequeña/Muy Pequeña
F13	Rotación del equipo desarrollador	Muy Alta/Alta/Media/Baja/Muy Baja
F14	Fiabilidad necesaria del software	Muy Alta/Alta/Media/Baja/Muy Baja
F15	Reuso de artefactos de requisitos existentes	LEL/EA/EF/ LEL-S/ Requisitos/No (múltiple selección)
F16	Creación de artefactos de requisitos para reuso	LEL/EA/LEL-S/ Requisitos/No (múltiple selección)
F17	Pre-Rastreabilidad de los requisitos	Total/Parcial/No
F18	Post-Rastreabilidad de los requisitos	Sí/No
F19	Granularidad de rastreabilidad de requisitos	Individual/Grupal/No Aplica
F20	Exigencia de producir el documento ERS	Sí/No
F21	Restricciones de tiempo y recursos para el proyecto	Muy Alto/Alto/Medio/Bajo/Muy Bajo
F22	Modelo de proceso de software	Cascada/Incremental/Iterativo/Transf. Formal/ Otro
F23	Tipo de Incremento	Nueva Funcionalidad/Mejora a Funcionalidad/No Aplica
F24	Experiencia en el proceso de requisitos	Muy Alto/Alto/Medio/Bajo/Muy Bajo
F25	Diferencias idiomáticas y/o culturales	Muy Alto/Alto/Medio/Bajo/Muy Bajo
F26	Diferencias temporales entre los involucrados	Muy Alto/Alto/Medio/Bajo/Muy Bajo

3.2 Dependencias entre Factores Situacionales

Tal como se ha mencionado anteriormente, una situación habitualmente no depende de un factor en forma aislada, sino que surge de la combinación de un

grupo de factores, por lo tanto, es relevante analizar la influencia que pueda existir entre algunos de ellos.

Se debe tener en cuenta la existencia de diferentes tipos de dependencias entre los factores, las que se deben considerar al momento de adaptar el proceso de IR. Cabe aclarar que cada tipo de dependencia solo es válida entre ciertos factores (ver Figura 3-1). Generalizando, si se considera que A y B son distintos factores situacionales puede suceder que exista una de las siguientes dependencias:

- *Dependencia invalidante*: indica que si A toma cierto valor no se debe tener en cuenta el factor B. Es decir, el factor B toma el valor “No Aplica”.
- *Dependencia limitante*: señala que el factor A con determinado valor restringe los valores que puede tomar el factor B.
- *Dependencia incompatible*: implica que A y B no pueden aplicar a la vez con ciertos valores. Por lo tanto, se requiere corregir la incompatibilidad detectada mediante una negociación o acuerdo entre los involucrados para modificar el valor de uno o de todos los factores afectados.

La Figura 3-1 permite visualizar, mediante una matriz, los factores que tienen algún tipo de dependencia entre sí. Como se puede observar, la mayoría de los factores tienen dependencia neutra, aun así, se han detectado 22 puntos con dependencias de los demás tipos, a las cuales se debe prestar especial atención. Éstas pueden establecerse una vez que se ha evaluado cada factor situacional y se le ha asignado el valor correspondiente. En la Tabla 3-6 se define cada uno de estos casos. Debe tenerse en cuenta que los valores de los factores situacionales podrían modificarse durante el avance del proceso (ver sección 5.1) y, como consecuencia, dichas dependencias también podrían cambiar.

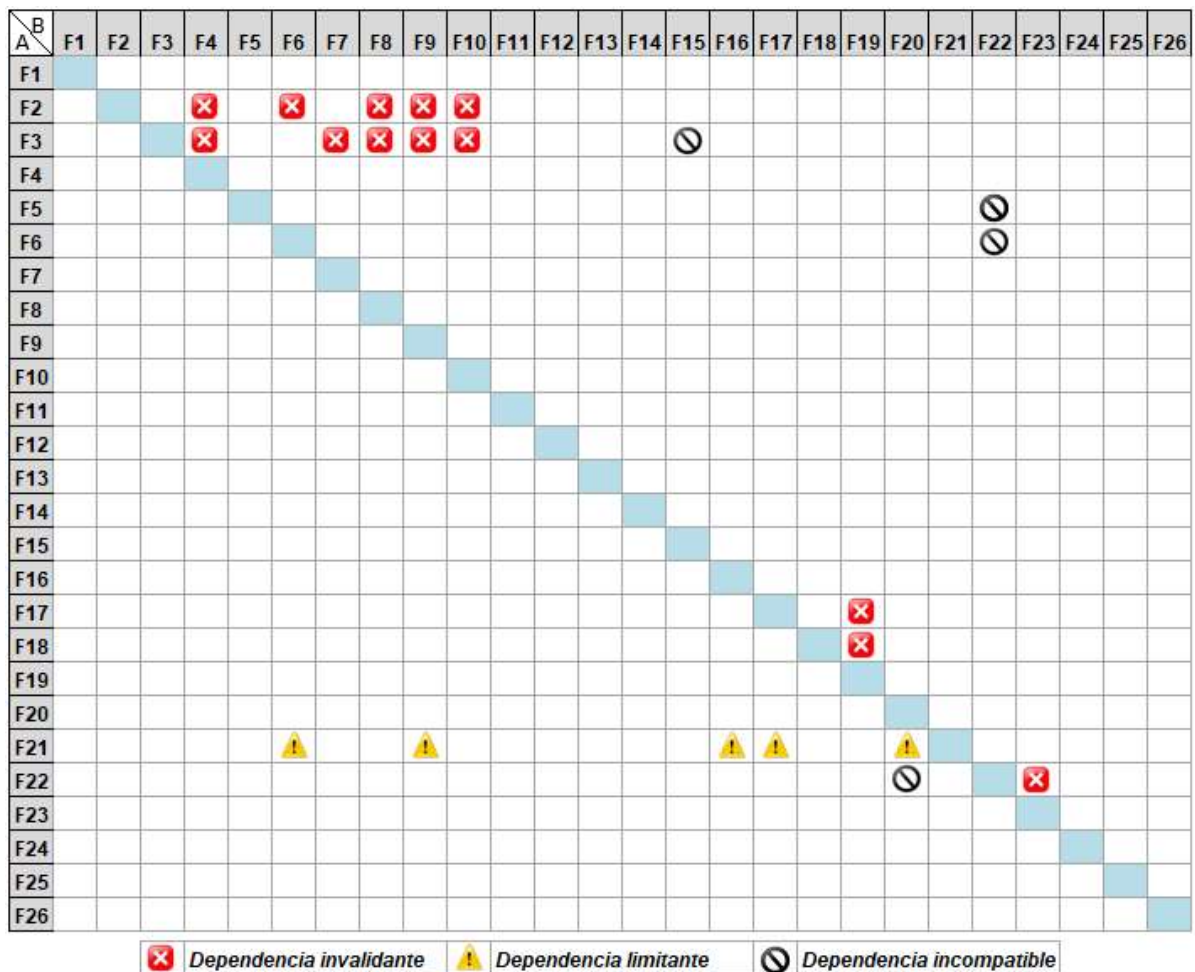


Figura 3-1. Matriz de dependencias entre los factores situacionales¹²

Estas dependencias son unidireccionales, donde un factor influye sobre otro factor, pudiéndose dar la dependencia de un grupo de factores influyendo sobre otro grupo de factores, es decir, son dependencias de N a M factores. Por otro lado, puede haber factores que presenten más de un tipo de dependencia o ninguna (independientes).

Se debe mencionar que al analizar las dependencias del factor *Restricciones de tiempo y recursos para el proyecto*, inicialmente se detectaron conflictos con los factores *Complejidad del contexto*, *Envergadura del proyecto*, *Fiabilidad necesaria del software*, *Conocimiento previo del dominio* y *Experiencia en el proceso de requisitos*. Sin embargo, estos se relacionan a aspectos que deben ser analizados durante el estudio de prefactibilidad del proyecto, por lo tanto, no se deberían negociar durante el proceso de IR. Existe además un análisis de factibilidad

¹² Léase la matriz: “el factor indicado en la fila influye sobre el factor indicado en la columna”.

durante el proceso de IR, es por ello que durante la adaptación del proceso puede haber negociaciones, por ejemplo, debido a las incompatibilidades mencionadas en la Tabla 3-6. Es claro también que si se modifica el alcance del sistema, o incluso los objetivos del mismo, durante este proceso de IR deberá realizarse un nuevo estudio de factibilidad y una replanificación del proyecto.

Tabla 3-6. Reglas de dependencias entre factores situacionales

Tipo de Dependencia	Descripción de Dependencia
Invalidante	<ul style="list-style-type: none"> • Si <i>Tipo de cliente</i> es "Mercado Potencial", no se consideran los factores: <i>Reingeniería en el proceso de negocio</i>, <i>Volatilidad de los requerimientos del cliente</i>, <i>Conflicto de intereses de usuarios</i>, <i>Rotación de usuarios</i> y <i>Localización de usuarios</i>.
	<ul style="list-style-type: none"> • Si <i>Novedad del contexto</i> es "Sí", no se consideran los factores <i>Reingeniería en el proceso de negocio</i>, <i>Inconsistencias en el contexto</i>, <i>Conflicto de intereses de usuarios</i>, <i>Rotación de usuarios</i> y <i>Localización de usuarios</i>.
	<ul style="list-style-type: none"> • Si <i>Pre-Rastreabilidad de los requisitos</i> y <i>Post-Rastreabilidad de los requisitos</i> toman el valor "No", se desestima el factor <i>Granularidad de rastreabilidad de requisitos</i>.
	<ul style="list-style-type: none"> • Si <i>Modelo de proceso de software</i> es distinto de "Incremental", no se considera el factor <i>Tipo de incremento</i>.
Limitante	<ul style="list-style-type: none"> • Si <i>Novedad del contexto</i> es "Si", <i>Reuso de artefactos de requisitos existentes</i> debe ser "No".
	<ul style="list-style-type: none"> • Si <i>Volatilidad del contexto</i> o <i>Volatilidad de los requerimientos del cliente</i> es "Alta" o "Muy Alta", <i>Modelo de proceso de software</i> no puede tomar el valor "Cascada" ni "Transformación Formal".
	<ul style="list-style-type: none"> • Si <i>Modelo de proceso de software</i> es "Incremental" o "Iterativo", <i>Exigencia de producir el documento ERS</i> debe ser "No"
Incompatible	<ol style="list-style-type: none"> 1. Se requerirá algún tipo de negociación si el valor de <i>Restricciones de tiempo y recursos para el proyecto</i> es "Alto" o "Muy Alto" y: <ul style="list-style-type: none"> <i>Volatilidad de los requerimientos del cliente</i> es "Alta" o "Muy Alta", o <i>Rotación de usuarios</i> es "Alta" o "Muy Alta", o <i>Creación de artefactos para reuso</i> es "LEL", "EA", "LEL-S" o "Requisitos", o <i>Pre-Rastreabilidad de los requisitos</i> es "Total", o <i>Exigencia de producir el documento ERS</i> es "Sí".

Con el propósito de facilitar la evaluación del conjunto de factores situacionales establecido, se ha confeccionado el *Formulario de Factores Situacionales* que brinda una guía sobre los valores admitidos para cada factor, como así también permite detectar dependencias de tipo limitante e invalidante entre los factores situacionales en función de los valores asignados a cada uno de estos (ver Anexo B).

Habiendo avanzado hasta este punto, se hace necesario determinar de qué modo dichos factores, por sí solos o agrupados, impactan en las distintas etapas del proceso de IR. En otras palabras, se necesita definir los mecanismos para la

adaptación de la estrategia de requisitos en cuestión, lo cual se analiza en el siguiente capítulo.

Capítulo 4

Mecanismo de Adaptación Situacional de la Estrategia de Requisitos

4.1 Introducción

Tal como se ha expuesto en los capítulos anteriores, la creación de un producto software de calidad depende en gran medida de la correcta especificación de sus requisitos. Dado que cada proyecto tiene características particulares, las que podrían ser definidas a partir de factores situacionales relacionados con el contexto de aplicación y con el proyecto en sí mismo, se hace ineludible un tratamiento adecuado de dichas características, para lo cual es altamente recomendable efectuar adaptaciones en el proceso de IR para cada situación particular, siendo por ende necesario disponer de un método claro que guíe la decisión sobre cómo adaptar el proceso.

La Figura 4-1 resume la idea general de la propuesta de esta tesis, la cual se sustenta en proveer mecanismos que permitan la adaptación del proceso de requisitos para cada proyecto específico, ajustándolo a las situaciones particulares de su contexto, en base a factores predominantes en él, de modo flexible y repetible.

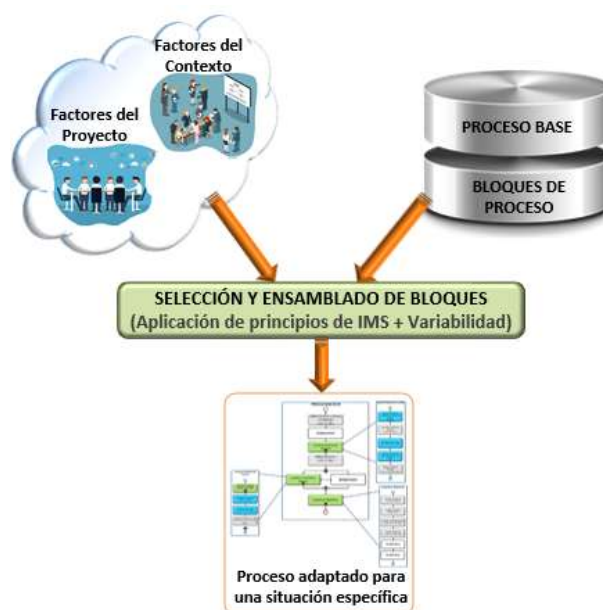


Figura 4-1. Obtención de un proceso adaptado a una situación

Se propone aplicar los principios de la IMS incorporando nociones de análisis de variabilidad de procesos, y utilizando la estrategia de IR basada en modelos en lenguaje natural (definida en sección 2.3) como proceso base para encarar su adaptación dinámica. Esto involucra tomar decisiones referidas a qué artefactos de requisitos construir, qué actividades del proceso realizar y qué técnicas específicas aplicar en cada actividad, de manera tal que se pueda lograr una mayor productividad del proceso, procurando una mejor calidad de los requisitos del software.

4.2 El Proceso Base y sus Puntos de Variación

La estrategia de IR involucra todas las actividades de la IR: elicitación, modelado, análisis y gestión. La Figura 4-2(a) describe dicha estrategia como un proceso base (Leite, Doorn, Kaplan, Hadad, & Ridaou, 2004; Hadad, 2008) con algunas mejoras posteriores, aunque independientes de las características del contexto de aplicación y del proyecto de software. De modo sintetizado el proceso base consiste en:

- i. Identificar el objetivo general y el alcance del sistema de software.
- ii. Comprender el vocabulario utilizado en el contexto de aplicación, con el apoyo del modelo LEL del UdeD.
- iii. Comprender el contexto presente, en base a un conjunto de EA que representan el comportamiento observado en el contexto de aplicación.
- iv. Refinar el objetivo general del sistema, mediante la descomposición en subobjetivos.
- v. Definir el contexto del sistema de software, mediante la producción de un conjunto de EF, que representan el comportamiento proyectado en el contexto de aplicación donde funcionará el sistema de software a desarrollar.
- vi. Crear el modelo LEL del Sistema, paralelamente a la actividad anterior, partiendo del LEL del UdeD, modificando la semántica de términos existentes, eliminando términos obsoletos, y adicionando nuevos términos utilizados en las descripciones de los EF.

- vii. Explicitar los requisitos, extrayéndolos del conjunto de EF, produciendo un conjunto de especificaciones de requisitos con sus atributos (Fichas de Requisitos) y el documento de ERS.

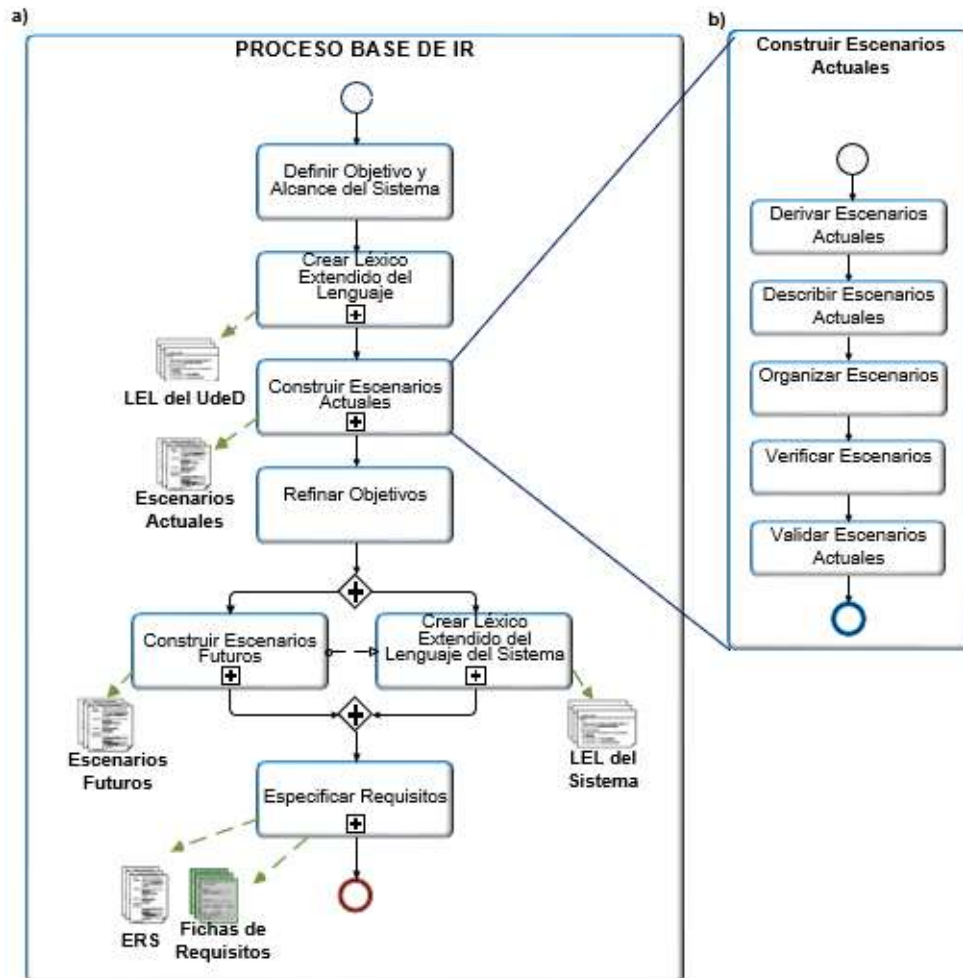


Figura 4-2. Proceso base de IR

Los modelos generados en este proceso, dado que están escritos en lenguaje natural, maximizan el uso de la terminología definida en los glosarios creados.

El modelo LEL del UdeD representa el vocabulario utilizado en el contexto de aplicación, por lo que el modelo de EA se describe utilizando los términos definidos en este glosario. El modelo LEL del Sistema representa un vocabulario lo más cercano posible al utilizado en el contexto de aplicación, aunque considerando algunas modificaciones en este vocabulario e incluyendo algunos términos necesarios para definir el futuro contexto con la incorporación del software a desarrollar (Kaplan, Doorn, & Gigante, 2013). Por ende, el modelo de EF, las Fichas de Requisitos y la ERS se describen utilizando los términos de este segundo glosario. De esta manera, se logra reducir en gran medida la ambigüedad

de las descripciones en lenguaje natural, y se logran modelos de más fácil comprensión por parte de los clientes y usuarios, y también del equipo de desarrollo.

Los principales subprocesos del proceso base, que se muestran en la Figura 4-2(a), se componen de actividades u otro nivel de procesos (utilizando la denominación en términos de IMS), algunos de los cuales pueden ser procesos atómicos en sí mismos o descomponerse en otros procesos (subactividades), como se visualiza en la Figura 4-2(b), la cual muestra de modo expandido el subproceso *Construir Escenarios Actuales*. Aunque en ambas partes de la figura se presenta un flujo secuencial, existen retroalimentaciones debido a las actividades de verificación y validación, y a la mejora continua en la comprensión del problema.

Existen algunas actividades del proceso de requisitos que no son afectadas por los factores situacionales, por lo cual, se deberán realizar siempre, mientras que otras pueden ser modificadas, eliminadas o reemplazadas, e incluso se pueden agregar nuevas actividades. No sólo las actividades están involucradas en estas adaptaciones, sino también los modelos a generar o utilizarse en el proceso pueden ajustarse también a la situación.

De aquí en más se utilizará el término “bloque de proceso” o directamente “bloque” para mencionar las actividades o subactividades involucradas en el proceso de IR.

Como se puede visualizar en la Figura 4-3, se han identificado cinco puntos de variación en el proceso base de IR. Cada uno de estos se relaciona con la capacidad de un subproceso para ajustarse a una situación particular, en función de la influencia que el conjunto de factores situacionales establecidos (ver subsección 3.1.2), por sí solos o agrupados, ejerce sobre cada uno de dichos puntos. Un punto de variación se considera un hito para revisar el proceso de IR posiblemente ya parcialmente adaptado, y reformularlo de ser necesario.

Por lo tanto, el proceso de requisitos puede reajustarse en cada punto de variación, dependiendo de una combinación de factores situacionales determinada. En la parte derecha de la Figura 4-3, se presenta una tabla donde se listan los factores específicos de influencia en cada punto de variación, identificándose: los factores que condicionan cómo componer el proceso de IR y los factores que se usan como parámetros en los bloques internos del subproceso asociado al punto de variación.

En un punto de variación, un mismo factor puede causar adaptaciones en cuanto a qué bloques compondrán el proceso y a la vez ser usado como parámetro de algún bloque del subproceso. Por ejemplo, el factor *Modelo de proceso de software* (F22) influye en la selección de los bloques que se utilizan para componer el subproceso *Crear Léxico Extendido del Lenguaje* en el Punto de Variación 1 y a la vez es parámetro de uno de sus bloques internos (*Identificar Símbolos*) como se visualiza en la Figura 4-5.

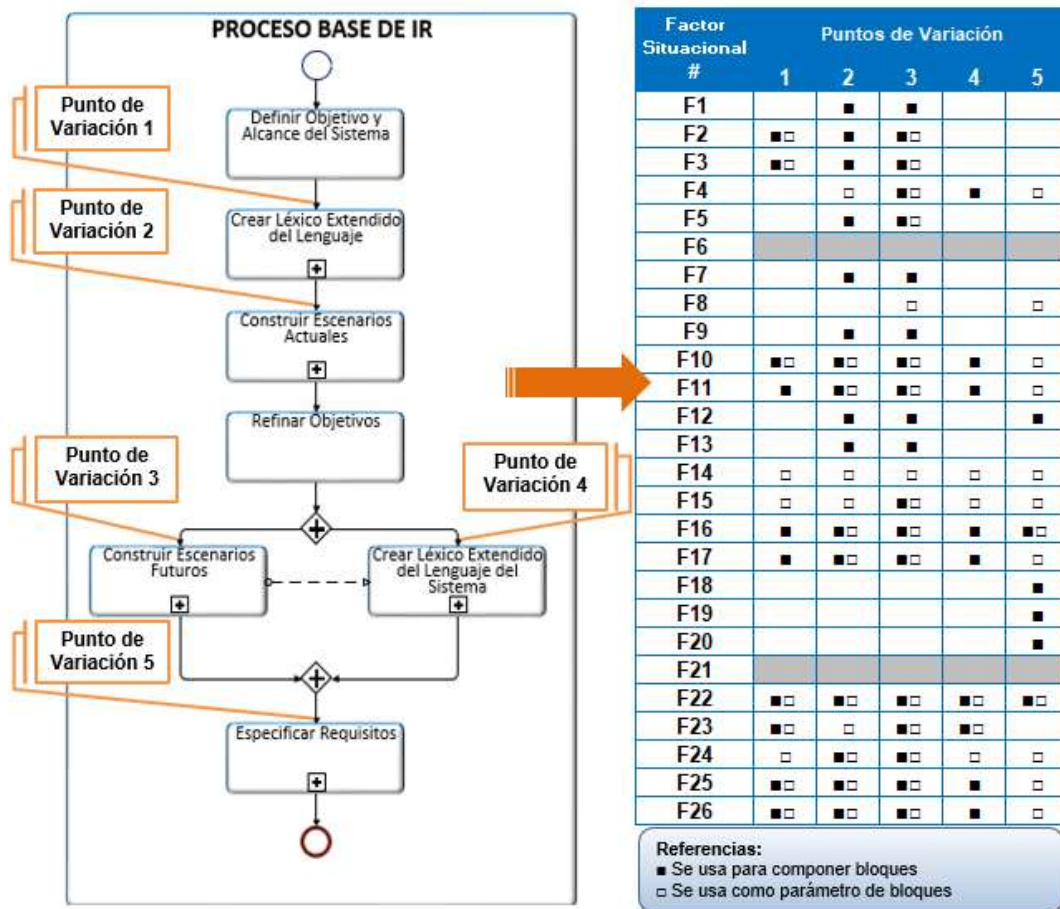


Figura 4-3. Factores situacionales y puntos de variación

En particular, se puede observar en la tabla presentada en la Figura 4-3 el factor *Volatilidad de los Requerimientos del Cliente* (F6) que no tiene influencia directa sobre los puntos de variación, sin embargo, dependiendo de su valoración, puede tener una dependencia de tipo limitante sobre otros factores. Similar es el caso del factor *Restricciones de tiempo y recursos para el proyecto* (F21), el cual no es usado directamente para adaptar el proceso, aun así, puede tener una incompatibilidad con otros factores, entonces es necesario considerarlo para la

negociación ante posibles conflictos. Ambos casos han sido explicados con mayor detalle en la sección 3.2.

4.3 Variantes de Adaptación

Al construir el proceso de IR para una situación específica, cada punto de variación puede admitir las siguientes operaciones:

- **Conservar:** se refiere a mantener el bloque de proceso sin variación;
- **Omitir:** consiste en eliminar un bloque de proceso;
- **Reemplazar:** consiste en sustituir un bloque de proceso en su totalidad por otro bloque de proceso específico;
- **Parametrizar:** consiste en pasar factores situacionales a un bloque de proceso, en el caso que este deba realizarse con alguna variación interna parcial, basada en recomendaciones propuestas según la valoración de dichos factores situacionales;
- **Agregar:** se refiere a incorporar un bloque de proceso específico al flujo del proceso.

La combinación de factores situacionales que intervienen en cada punto de variación determina cuál de estas cinco operaciones se debe llevar a cabo.

Por otra parte, se han definido cuatro tipos de bloques de proceso:

- **Independiente,** aquel que es autónomo, no depende de los factores situacionales. Es decir, pertenece a un subproceso y se incluye siempre en el proceso de IR, y además no es compartido por otros subprocesos. Por ejemplo, el bloque de proceso *Refinar Objetivos* (ver Figura 4-4).
- **Compartido,** aquel que es común a diferentes subprocesos del proceso de IR, puede depender de factores situacionales para ser utilizado. Por ejemplo, el bloque de proceso *Organizar Escenarios* es compartido por dos subprocesos principales: *Construir Escenarios Actuales* y *Construir Escenarios Futuros* (ver Figuras 4-6 y 4-7).
- **Variante,** aquel que es un bloque que pertenece a un único subproceso específico y depende de factores situacionales. Es el caso del bloque de proceso *Planear Elicitación*, el cual como se visualiza en la Figura 4-5,







puede ser reemplazado por el bloque *Planear Estudio de Mercado*, de acuerdo con el valor de los factores situacionales involucrados.

- **Nulo**, es un bloque auxiliar utilizado cuando un subproceso, en cualquier nivel, debe suprimirse o saltarse en el proceso de IR, entonces cada bloque de proceso a omitir se sustituye por el *Bloque Nulo* (ver Figura 4-4).

Cabe mencionar que un bloque parametrizable puede ser un bloque independiente, compartido o variante. Un bloque puede tener como parámetros factores situacionales individuales, o condiciones (combinaciones lógicas de factores), tal es el caso del bloque *Describir Símbolos* en la Figura 4-5, el cual es parametrizable en función de los factores F2, F3, F15, y de la condición C8, la cual se compone de un conjunto de factores con ciertos valores específicos. Esto implica que la descripción de un bloque de proceso con parámetros tiene una descripción general y algunas partes con descripciones de variantes menores según los posibles valores de los parámetros.

Para modelar las adaptaciones del proceso de IR, se utiliza la notación BPMN¹³ con algunas extensiones para favorecer la visualización de la variabilidad en cada parte del proceso. En la Tabla 4-1 se muestran los elementos principales que se utilizarán para tal fin.

Tabla 4-1. Elementos básicos para representar la adaptación del proceso de IR

Elemento	Descripción
	<ul style="list-style-type: none"> • Repositorio común que contiene todos los bloques de proceso
	<ul style="list-style-type: none"> • Bloque independiente
	<ul style="list-style-type: none"> • Bloque compartido
	<ul style="list-style-type: none"> • Bloque variante
	<ul style="list-style-type: none"> • Bloque nulo
	<ul style="list-style-type: none"> • Indica la ubicación donde se debe agregar un bloque de proceso bajo cierta condición
<< p1, p2, pn >>	<ul style="list-style-type: none"> • Parámetros del bloque de proceso

A continuación, en las subsecciones 4.3.1 a 4.3.7, se presentan los mecanismos que sirven de guía para ajustar el proceso. A tal fin, se proveen las reglas de adaptación necesarias en cada uno de los puntos de variación, las cuales indican

¹³ <http://www.bpmn.org/>

las combinaciones de factores situacionales que se emplean en cada parte y determinan las operaciones a aplicar en cada bloque de los subprocessos o en el flujo del propio proceso base.

4.3.1 Variantes del Proceso Base

La Figura 4-4 presenta el proceso base de IR con las variantes de mayor nivel de abstracción. El énfasis está puesto en mostrar, por una parte, el orden de sus distintas etapas y, por otra, cómo bajo determinados contextos, algunos bloques de proceso, correspondientes a los subprocessos de primer nivel, podrían ser omitidos reemplazándose por el *Bloque Nulo*. Las condiciones que rigen estas variantes de nivel superior se describen a continuación en esta subsección. Algunas condiciones pueden ser usadas en más de un punto de variación en distintos niveles del proceso, e incluso como parámetros de bloques de proceso, como se mencionó en la anterior subsección. En la Figura 4-4 (y en subsiguientes), las condiciones se indican con la fórmula:

$C\#=V$ o $C\#=F$, siendo $C\#$: número de condición, V: Verdadero y F: Falso

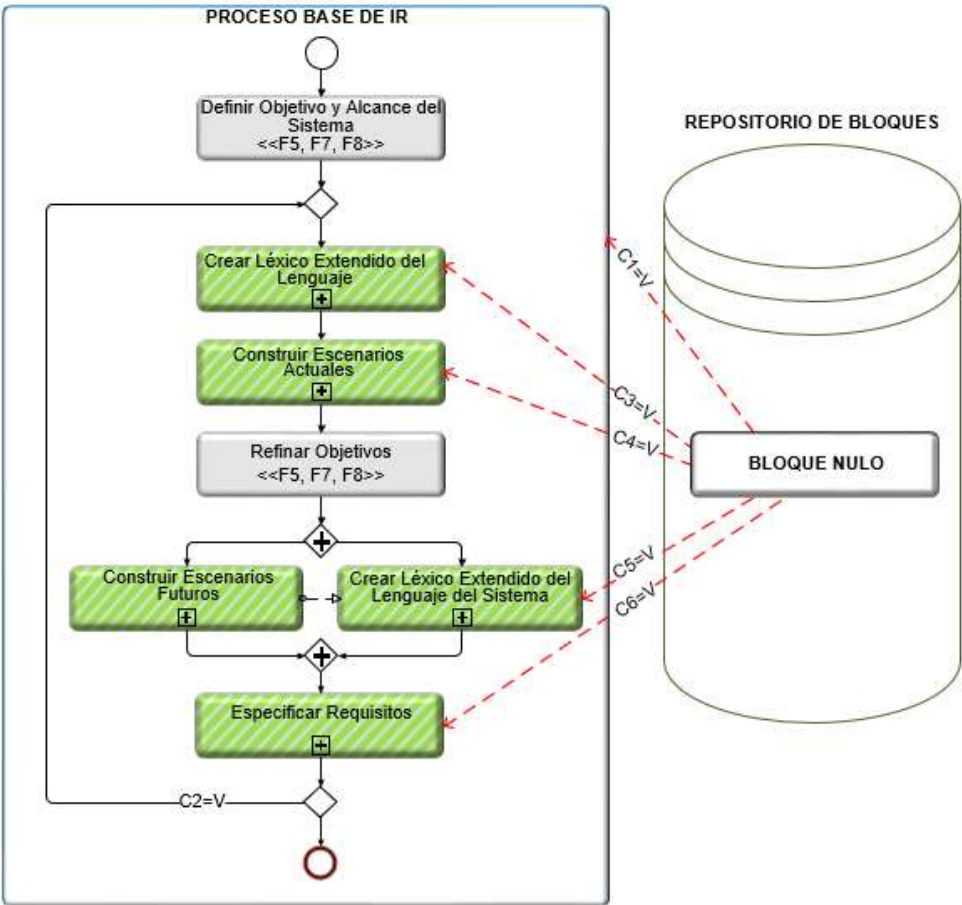


Figura 4-4. Variantes del proceso base de IR

Una de las cuestiones a decidir cuándo se inicia la gestión de cualquier proyecto es el *Modelo de proceso de software* (F22) que va a guiar el desarrollo. En función de este factor se determina la manera en que se estructuran las etapas o actividades a realizar y los artefactos a elaborar para desarrollar el producto software. Es por ello que, este es un factor dominante en la adaptación del proceso de IR, a partir del cual las restantes adaptaciones están fuertemente supeditadas a este. La selección del modelo de proceso trasciende a los ingenieros de requisitos, ya que está vinculada con la gestión del proyecto en sí misma. En otras palabras, el proceso de IR debe ser consecuente con el modelo de proceso elegido, esto es, se deben hacer los ajustes necesarios en el orden y en la forma de realizar las actividades de cada bloque de proceso, siendo incluso determinante para aplicar o no el proceso de requisitos.

Los modelos de proceso que se han considerado influyentes para la adaptación del proceso de IR son: *Cascada*, *Incremental*, *Iterativo*, *Transformación Formal* y, se utiliza la opción *Otros* cuando la estrategia de IR basada en modelos en lenguaje natural no puede ser aplicada. Luego de un análisis detallado de la posible influencia de cada uno de estos modelos, se ha determinado que el equipo de requisitos puede encontrarse con alguna de las siguientes situaciones:

- a) Se ha elegido algún modelo de proceso en el que no es aplicable el proceso de IR, por ejemplo, un modelo prototipado o una metodología ágil, por lo cual la primera condición a evaluar y excluyente es:

Condición C1:

Modelo de proceso de software (F22) = *{Otro}*

Este representa el caso más extremo cuando la condición C1 es verdadera, entonces el *Proceso Base de IR* se omite, reemplazándose el proceso completo por el *Bloque Nulo* (Ver Figura 4-4). Por lo tanto, no tiene sentido evaluar las restantes condiciones.

- b) El proyecto sigue un modelo de proceso “Incremental” o “Iterativo”, esto se evalúa con la siguiente condición:

Condición C2:

Modelo de proceso de software (F22) = *{Incremental, Iterativo}*

Ambos modelos involucran ciclos en el *Proceso Base de IR* (ver Figura 4-4). Si el modelo de proceso es “Incremental” la evaluación de los factores situacionales se repite en cada uno de los puntos de variación para cada ciclo. En cambio, si se trata de un modelo de proceso “Iterativo” los factores situacionales se evalúan la primera vez que se pasa por cada punto de variación y no en cada ciclo. Se debe tener en consideración que los artefactos a producir y las formas de realizar las actividades de cada subproceso están condicionados además por los restantes factores situacionales.

Si la condición C2 es verdadera porque el modelo es “Incremental”, como parte de la gestión del proyecto se realiza un planeamiento de incrementos en función del objetivo y el alcance definido para el sistema. Bajo este modelo de proceso cada incremento puede implicar el desarrollo de nuevas funcionalidades según se asignen prioridades o, puede tratarse de una mejora a funcionalidades creadas previamente. Entonces bajo este modelo, las actividades a realizarse durante el proceso de IR en cada ciclo dependerán del *Tipo de incremento* (F23), es decir, si es una funcionalidad nueva que se agrega o una mejora a lo que ya se tiene. Este factor se determina al inicio de cada ciclo, el cual coincide con el Punto de Variación 1 y tiene validez durante todo el ciclo (sin reevaluarse en los restantes puntos de variación).

En caso de una nueva funcionalidad, implica en el ciclo en curso realizar una parte detallada y completa del LEL del UdeD, de los EA, de los EF, del LEL del Sistema, y de las especificaciones de los requisitos correspondientes a la funcionalidad planificada para el incremento.

En el caso de mejora a una funcionalidad existente, se mejora en el ciclo una parte de los modelos ya construidos, por ejemplo, se mejoran los símbolos de los léxicos, y/o los escenarios afectados a la funcionalidad que se desea mejorar, eventualmente podrían surgir nuevos símbolos o escenarios. En ambos casos, por cada incremento que se realiza, se agrega a cada modelo ya construido una parte detallada y completa que debe acoplarse a lo que existe de los incrementos anteriores.

Por otra parte, la condición C2 puede ser verdadera porque el modelo de proceso adoptado es el "Iterativo". Este modelo implica que las actividades del proceso se deben repetir sucesivamente, donde se comienza con aquello establecido como lo más relevante según el objetivo y el alcance del sistema proyectado, y en cada iteración se irá refinando la definición del sistema hasta lograr su versión final. En cuanto al proceso de requisitos, en cada ciclo se realizan las actividades del proceso y se refina cada modelo ya construido en las iteraciones anteriores, aportando mayor detalle en diversas partes o agregando partes nuevas, que no son funcionalidades completas. En la práctica, esto implica modelos muy incompletos en las primeras iteraciones, por ejemplo, habrá pocos símbolos en el LEL del UdeD o con definiciones poco detalladas, y en el caso de los escenarios, estos contendrán descripciones más abstractas con ausencia de varios episodios o escenarios, los que se irán completando a medida que se va iterando.

- c) El desarrollo se rige por un modelo de proceso "Cascada", donde cada fase del proceso comienza cuando se ha finalizado la anterior, o se aplica un modelo de "Transformación Formal", el cual aplica una serie de transformaciones para convertir una especificación formal en un sistema ejecutable, donde inicialmente se obtiene una especificación de requisitos a partir de la cual se genera la especificación formal. En este contexto la condición C2 es falsa (y la condición C1 también es falsa). Ambos modelos implican la aplicación del bloque *Proceso Base de IR* completo, en una única iteración, por supuesto, con los ajustes necesarios en función de los demás factores situacionales, para finalmente realizar la entrega total de la documentación de requisitos.

Además de lo expuesto anteriormente, en la Figura 4-4 se puede observar que determinadas condiciones implican que algunos bloques de proceso sean omitidos, siendo reemplazados por el *Bloque Nulo*, lo cual indica que no se realiza ese subproceso. Por lo tanto, a continuación, se sintetiza cada uno de los subprocesos del *Proceso Base de IR* con sus posibles adaptaciones, sin adentrarse en las actividades que se realizan internamente en cada uno de ellos, ya que ese detalle se describe posteriormente:

- *Definir Objetivo y Alcance del Sistema:*

Identificar el objetivo general y el alcance del sistema, es una actividad que se debe realizar al inicio de todo proyecto, independientemente de su contexto, por tal motivo es un bloque de proceso independiente. Sin embargo, como se puede observar en la Figura 4-4 recibe como parámetros los factores *Volatilidad del contexto* (F5) e *Inconsistencias en el Contexto* (F7), donde si estos factores tienen valor “Alto” o “Muy Alto” implica cambios continuos y/o contradicciones en la información elicitada lo cual puede dificultar la definición del objetivo del sistema y su alcance. Bajo estas condiciones se debe establecer entre los clientes y el equipo de ingenieros un objetivo de más alto nivel (más abarcativo) y con un alcance que permite ser extendido o eventualmente acotado. También es un parámetro el factor *Conflicto de intereses de usuarios* (F8), se requiere especial atención cuando su valoración es “Alto” o “Muy Alto”, a fin de identificar tempranamente los distintos puntos de vista de los usuarios o intereses en conflicto, de manera tal que el alcance del sistema incluya los puntos comunes a todos los interesados. En general, bajo estas situaciones se requiere una negociación más intensiva para lograr un consenso inicial del objetivo y del alcance del sistema.

- *Crear Léxico Extendido del Lenguaje:*

El producto generado en este bloque es el LEL del UdeD, el cual se ha descrito en la subsección 2.3.1. Con este artefacto se pretende comprender el vocabulario empleado en el dominio de la aplicación, y de este modo mejorar la comunicación con los clientes y usuarios eliminando posibles ambigüedades. Se va a prescindir de crear el léxico en el caso que el equipo de requisitos haya participado en proyectos referidos al mismo dominio al que pertenece el software a desarrollar, lo cual implica que tiene una amplia experiencia en el UdeD (F11). Lo anterior aplica siempre que no existan diferencias idiomáticas (F25) ni temporales entre los involucrados (F26) o que los usuarios se encuentren distribuidos geográficamente (F10). A pesar de ello, hay otros factores que se deberán considerar para evaluar la posibilidad de omitir este bloque: que no haya exigencia de construir un LEL del UdeD para reuso (F16), o la necesidad de cumplir con una pre-rastreabilidad a todos los artefactos del proceso (F17). Lo anterior se define a partir de la siguiente condición:

Condición C3:

Creación de artefactos de requisitos para reúso (F16) = {*No, EA, LEL-S, Requisitos*} Y

Pre-Rastreabilidad de los requisitos (F17) = {*No, Parcial*} Y

Conocimiento previo del dominio (F11) = {*Muy Alto, Alto*} Y

Diferencias idiomáticas y/o culturales (F25) = {*Bajo, Muy Bajo*} Y

Diferencias temporales entre los involucrados (F26) = {*Medio, Bajo, Muy Bajo*} Y

Localización de usuarios (F10) = {*Colocalizados*}

Si la condición C3 se cumple entonces se puede avanzar con el proceso sin construir el LEL del UdeD. Es decir, al planear el proceso de requisitos este subproceso es omitido, mediante su reemplazo por el *Bloque Nulo*. En la subsección 4.3.2 se va a descomponer este subproceso y se describirán las variaciones internas del mismo en función a los factores situacionales que lo afectan.

- *Construir Escenarios Actuales:*

Este subproceso, descrito previamente en el apartado 2.3.2, modela las situaciones del contexto presente, a través de los EA. En determinadas situaciones, esta actividad carecería de sentido, por ejemplo, si se va a desarrollar un software para un mercado potencial (F2), o si se trata de un contexto nuevo (F3). Por otra parte, sería improductivo crear los EA cuando el nivel de cambios en el contexto de aplicación sea extremadamente alto (F5). Además de los casos mencionados, hay dos situaciones en las que se podría prescindir de los EA. Una es cuando el equipo de requisitos además de contar con una sólida experiencia en el dominio conoce en profundidad las actividades del contexto de aplicación, cómo se desenvuelven y cuáles son las responsabilidades de los actores en dicho contexto particular (F11). Si tal fuera el caso, se debe considerar que no deben existir diferencias idiomáticas, culturales (F25) ni temporales (F26) significativas entre los involucrados, y que los usuarios estén en un mismo lugar (F10). De modo similar, podría omitirse la construcción de los EA si el equipo tiene sólida experiencia en el proceso de IR (F24), el tamaño del proyecto no es grande (F12), para un contexto muy simple, de baja complejidad (F1), con mínimas inconsistencias (F7), y hay poca rotación de los usuarios (F9) y también del equipo de desarrollo (F13). Ambos contextos mencionados admiten la omisión de este bloque siempre que no haya la exigencia de construir EA para reúso (F16) o de cumplir con una pre-

rastreabilidad a todos los artefactos (F17). La siguiente condición cubre todas las alternativas mencionadas:

Condición C4:

(Tipo de cliente (F2) = {*Mercado Potencial*}) ○

Novedad del contexto (F3) = {*Si*} ○

Volatilidad del contexto (F5) = {*Muy Alta*} ○

((Creación de artefactos de requisitos para reúso (F16) = {*No, LEL, LEL-S, Requisitos*}) Y

Pre-rastreabilidad de los requisitos (F17) = {*No, Parcial*}) Y

((Conocimiento previo del dominio (F11) = {*Muy Alto*}) Y

Diferencias idiomáticas y/o culturales (F25) = {*Bajo, Muy Bajo*}) Y

Diferencias temporales entre los involucrados (F26) = {*Medio, Bajo, Muy Bajo*}) Y

Localización de usuarios (F10) = {*Colocalizados*}) ○

(Complejidad del contexto (F1) = {*Bajo, Muy Bajo*}) Y

Inconsistencias en el contexto (F7) = {*Bajo, Muy Bajo*}) Y

Rotación de usuarios (F9) = {*Media, Baja, Muy Baja*}) Y

Envergadura del proyecto (F12) = {*Mediana, Pequeña, Muy Pequeña*}) Y

Rotación del equipo desarrollador (F13) = {*Media, Baja, Muy Baja*}) Y

Experiencia en el proceso de requisitos (F24) = {*Muy Alta, Alta, Media*}}))

Por lo tanto, cuando la condición C4 sea verdadera no se construyen los EA, omitiéndose este bloque al reemplazarlo por el *Bloque Nulo*. Las adaptaciones internas de este subproceso se describen en la subsección 4.3.3.

- *Refinar Objetivos:*

Es un bloque de proceso independiente, que siempre ha de realizarse, pues a medida que avanza el proyecto, hay una mejor comprensión de las necesidades, aunque también pueden ocurrir ciertos cambios organizacionales que alteren lo inicialmente solicitado. En consecuencia, se debe refinar aquel objetivo preliminar del sistema, descomponiéndolo en objetivos específicos, y también ajustar su alcance. En esta etapa, se debe considerar los factores *Volatilidad del contexto* (F5), *Inconsistencias en el Contexto* (F7) y *Conflictos de intereses de usuarios* (F8) (parámetros que recibe este bloque de proceso en la Figura 4-4). Cuando la valoración de estos factores sea “Muy Alto” o “Alto”, se requerirán negociaciones con los clientes para acordar sobre los cambios que ocurren en el contexto, las contradicciones en la información elicitada o en los objetivos, y/o intereses en conflicto, incluso pueden plantearse soluciones alternativas a muy alto nivel, presentando las ventajas y desventajas de cada una, a fin de ayudar a

los clientes a tomar decisiones y conseguir un consenso sobre los objetivos específicos a alcanzar.

- *Construir Escenarios Futuros:*

En este bloque de proceso se modelan las situaciones proyectadas de acuerdo con los objetivos específicos identificados. Asimismo, pueden describirse a través de EF propuestas alternativas originadas por los ingenieros de requisitos o provenientes de los clientes, las cuales se deberán negociar y acordar. Una particularidad de este subproceso, descrito previamente en la subsección 2.3.3, es que los EF se van a construir siempre, lo cual marca una diferencia con los otros artefactos que se producen al aplicar esta estrategia de requisitos, pues bajo determinadas situaciones pueden ser omitidos. Sin embargo, dependiendo de la valoración de los factores situacionales, existen diferentes enfoques en cuanto al modo en que estos escenarios se construyen. En la subsección 4.3.4 se describen en detalle las variantes para este subproceso.

- *Crear Léxico Extendido del Lenguaje del Sistema:*

Paralelamente a la construcción de los EF se construye el LEL del Sistema, como se observa en la Figura 4-4. Este bloque de proceso puede omitirse si no hay exigencia de construir un LEL del Sistema para reúso (F16) y no hay exigencia de cumplir con una pre-rastreabilidad a todos los artefactos (F17) y, además, el equipo de requisitos tiene una amplia experiencia en el UdeD (F11), no existen diferencias idiomáticas (F25) ni temporales (F26) entre los involucrados, los usuarios no están distribuidos geográficamente (F10) y no hay alta reingeniería en los procesos del negocio (F4). Dicha situación queda definida con la siguiente condición:

Condición C5:

Creación de artefactos de requisitos para reúso (F16) = {No, LEL, EA, Requisitos} Y

Conocimiento previo del dominio (F11) = {Muy Alto, Alto} Y

Diferencias idiomáticas y/o culturales (F25) = {Bajo, Muy Bajo} Y

Diferencias temporales entre los involucrados (F26) = {Medio, Bajo, Muy Bajo} Y

Localización de usuarios (F10) = {Colocalizados} Y

Reingeniería en el proceso de negocio (F4) = {Media, Baja, Muy Baja} Y

Pre-Rastreabilidad de los requisitos (F17) = {No, Parcial}

Como se puede notar en la condición C5, la mayoría de los factores a evaluar son los mismos que para decidir si se construirá el LEL del UdeD (ver condición C3), con lo cual cuando se omita la construcción del LEL del UdeD también es posible que se omita la del LEL del Sistema. Sin embargo, siempre que haya un alto grado de reingeniería en los procesos de negocio (F4), este último léxico debe ser creado para mitigar la incorporación de ambigüedades en las descripciones de los EF, principalmente respecto a la terminología ya empleada en el UdeD que presente significados ligeramente modificados. Básicamente este bloque podría omitirse cuando los usuarios tengan un buen conocimiento informático, de manera tal que no requieran definiciones explícitas de la terminología empleada en los EF. Los mecanismos para la adaptación interna de este subproceso se describen en la subsección 4.3.6.

- *Especificar Requisitos:*

Las actividades de este subproceso se han explicado en el apartado 2.3.4. En algunos proyectos de cierta envergadura puede existir una exigencia de producir un documento con todos los requisitos del software como parte del contrato entre la organización cliente y la organización desarrolladora (F20), lo que hace mandatorio la utilización de este bloque de proceso. Sin embargo, en otros contextos podría omitirse, por ejemplo, para un proyecto de pequeña envergadura (F12), cuando no hay exigencia de producir requisitos para reuso (F16), ni es necesario individualizar los requisitos (F19) dado que no son imprescindibles actividades tales como mantener la rastreabilidad con los componentes de software u otros artefactos posteriores (F18). La condición que determina si este bloque forma o no parte del proceso es la siguiente:

Condición C6:

Envergadura del proyecto (F12) = *{Pequeña, Muy Pequeña}* Y

Creación de artefactos de requisitos para reuso (F16) = *{No, LEL, EA, LEL-S}* Y

Exigencia de producir el documento ERS (F20) = *{No}* Y

Post-Rastreabilidad de los requisitos (F18) = *{No}* Y

Granularidad de rastreabilidad de los requisitos (F19) = *{Grupal}*

Cuando la condición C6 sea verdadera el bloque *Especificar Requisitos* se anula, es decir, se reemplaza por el *Bloque Nulo*. En la subsección 4.3.7 se describen los ajustes internos a este subproceso.

La comprensión de las adaptaciones generales del *Proceso Base de IR* sienta las bases para detallar las alternativas internas posibles en cada uno de sus cinco bloques variantes, donde cada uno de estos tiene correspondencia con un punto de variación del proceso (ver Figura 4-3). En las subsecciones siguientes, se describen los mecanismos de adaptación que aplican a cada uno de dichos puntos, en función a las particularidades de cada contexto.

4.3.2 Variantes al Crear el Léxico Extendido del Lenguaje

La Figura 4-5 permite visualizar las adaptaciones generales del subproceso *Crear Léxico Extendido del Lenguaje* (Punto de Variación 1). Tal como se puede observar en la Figura, hay condiciones que implican que algunos bloques de proceso sean reemplazados por otros, estas se describen a continuación:

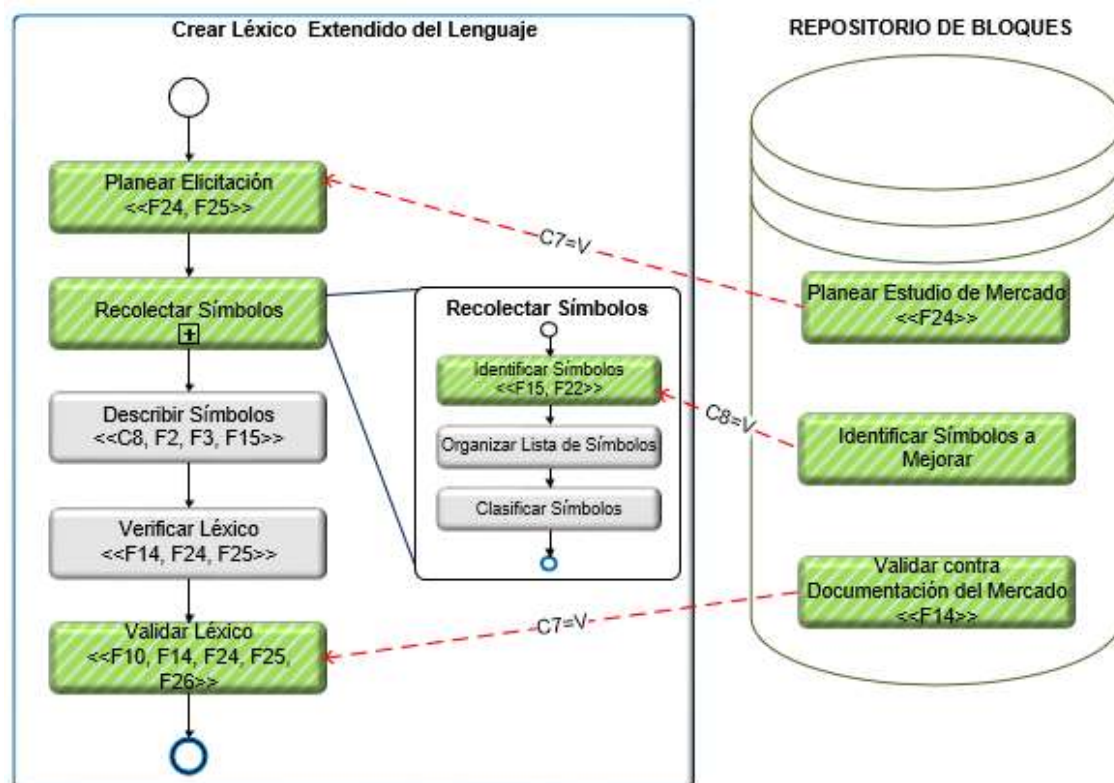


Figura 4-5. Variantes al Crear el Léxico Extendido del Lenguaje

- El bloque *Planear Elicitación* puede ser reemplazado por el bloque *Planear Estudio de Mercado* cuando el software a desarrollar es para un mercado potencial (F2) o es para un contexto nuevo (F3). Dado que habrá distintas fuentes de información, esta situación también implica reemplazar el bloque *Validar Léxico* por el bloque *Validar contra Documentación del Mercado*. Se debe evaluar la siguiente condición:

Condición C7:

Tipo de cliente (F2) = *{Mercado Potencial}* O

Novedad del contexto (F3) = *{S}*

- El subproceso *Recolectar Símbolos* se debe realizar siempre, aunque puede tener modificaciones parciales internas. Como se visualiza en la Figura 4-5, el bloque *Identificar Símbolos* se puede reemplazar por *Identificar Símbolos a Mejorar*. Dicho reemplazo se debe hacer cuando bajo un modelo de proceso de software incremental (F22) se ha planificado una mejora a determinada funcionalidad (F23) para ese ciclo, lo que se determina con la condición C8:

Condición C8:

Modelo de proceso de software (F22) = *{Incremental}* Y

Tipo de incremento (F23) = *{Mejora a Funcionalidad}*

A continuación, se describe cada bloque que conforma el subproceso y el impacto de los factores situacionales en cada uno estos:

- *Planear Elicitación*

Se determinan las fuentes de información y se seleccionan las técnicas de elicitación a utilizar. En la selección de técnicas a utilizar es importante tomar en cuenta la *Experiencia en el proceso de requisitos* (F24), pues la implementación de algunas técnicas puede requerir una experiencia “Alta” o “Muy Alta”. Si el factor *Diferencias idiomáticas y/o culturales* (F25) es valorado “Alto” o “Muy Alto”, parte de esta actividad es la elección del idioma en que se creará el léxico y los siguientes artefactos a construir, dado que serán compartidos entre todos los involucrados. Es práctica habitual realizar la documentación en un único idioma para evitar inconsistencias en la misma dada la posible evolución de la misma.

- *Planear Estudio de Mercado*

Dado que no existe la figura de cliente ni usuario, se identifican otros tipos de fuentes de información y se planean otros métodos para llevar adelante la elicitación, tales como realizar un estudio de mercado, encuestas a potenciales clientes, revisión de software existente en el mercado, entre otros. La elección de las técnicas de elicitación a aplicar será en función de la *Experiencia en el proceso de requisitos* (F24) que tenga el equipo.

- *Recolectar Símbolos*

Es un subproceso que se expande en tres bloques menores (ver Figura 4-5). Los bloques que permiten cubrir las variantes de este subproceso son: *Identificar Símbolos*, *Identificar Símbolos a Mejorar*, *Organizar Lista de Símbolos* y *Clasificar Símbolos*, estos se describen a continuación.

- *Identificar Símbolos*

Se identifican los símbolos relevantes del UdeD. Cuando la valoración del factor *Reuso de artefactos de requisitos existentes* (F15) sea “LEL”, como primera actividad se identifican los términos que pueden reusarse de dicho léxico, luego se pueden identificar otros símbolos directamente del propio UdeD. En todos los casos, el alcance de las tareas descritas en este bloque dependerá del *Modelo de proceso de software* que se esté aplicando (F22). Si se aplica un modelo “Cascada” o uno de “Transformación Formal”, se identifican los términos del UdeD para el alcance definido. Si se trata de un proceso “Iterativo” se pueden identificar nuevos símbolos a partir del UdeD para ampliar el vocabulario definido en ciclos anteriores. En un modelo de proceso “Incremental”, se está en la situación de una nueva funcionalidad (por la condición C8=F), entonces se identifican nuevos símbolos a partir del UdeD para el alcance definido para el incremento.

- *Identificar Símbolos a Mejorar*

En lugar de identificar nuevos símbolos del UdeD, se toman como base el LEL del UdeD creado en los incrementos previos. El propósito es identificar aquellos símbolos, relacionados a la funcionalidad a mejorar, que necesiten ser refinados, para luego incluirles mayor detalle en sus definiciones.

- *Organizar Lista de Símbolos*

Se trata de un bloque de proceso independiente de los factores situacionales, por lo cual siempre es utilizado. La actividad del bloque consiste en definir cómo se crea una lista única y ordenada de los términos recolectados.

- *Clasificar Símbolos*

En este bloque de proceso independiente de los factores situacionales, que se conserva, se describe cómo cada símbolo se clasifica de acuerdo con su tipo:

Sujeto, Verbo, Objeto y Estado, o puede establecerse la necesidad de una clasificación particular para el UdeD en estudio.

- *Describir Símbolos*

Se trata de un bloque de proceso que se utiliza siempre, implica describir cada símbolo a partir del conocimiento previamente elicitado y retornando al UdeD para capturar más información cada vez que sea necesario. Son varios los factores que se consideran parámetros de este bloque. Entre estos, el *Tipo de cliente* (F2) y la *Novedad del contexto* (F3) se deberán considerar para determinar los tiempos verbales al redactar las definiciones. Así es que un LEL del UdeD para un negocio nuevo tendrá sus oraciones en tiempo futuro "será/se hará" o "quisiera ser/hacerse". Si el software es para un mercado potencial puede tener descripciones en presente de lo que "es/se hace" ahora en el mercado, o de lo que "quisiera ser/hacerse". En cambio, en los demás casos se definirá directamente lo que "es/se hace", o también "debiera ser/hacerse" (lo que no se hace, pero debería hacerse) y eventualmente se podría definir algo de lo que "será/se hará". Otra cuestión a tener en cuenta es el factor *Reuso de artefactos de requisitos existentes* (F15=LEL). Si hay posibilidad de reusar un LEL del UdeD existente, se reutilizan todas las descripciones posibles de los símbolos del LEL preexistente, identificados en el bloque previo, y se elicitan descripciones particulares del UdeD. Por otra parte, si se está trabajando en una mejora a determinada funcionalidad bajo un modelo de proceso incremental (C8=V), durante la descripción se mejoran las definiciones de aquellos símbolos existentes, identificados en la actividad anterior.

- *Verificar Léxico*

En este bloque, siempre conservado en el proceso de IR, se describe la selección de las técnicas de verificación a aplicar, las que dependen fundamentalmente del grado de *Fiabilidad necesaria del software* (F14), y su posterior aplicación. Por ejemplo, si hay menor exigencia se pueden aplicar técnicas menos estructuradas como la lectura ad-hoc o análisis con checklist. En cambio, si este factor toma valor "Muy Alta" o "Alta", se deberá aplicar alguna técnica de alta efectividad, que permita realizar la actividad con el mayor rigor posible, por ejemplo, inspecciones, incluso se podrá optar por aumentar la cantidad ingenieros de requisitos que participen en esta actividad, que puedan

aportar distintos puntos de vista para reforzar la efectividad de la técnica. Sin embargo, la implementación de cada una de estas técnicas puede requerir mayor o menor grado de expertise por parte de los ingenieros de requisitos, por tal motivo se considera la valoración del factor *Experiencia en el proceso de requisitos* (F24) al seleccionar las técnicas a aplicar. Otro factor que influye en el tipo de verificación es si hay *Diferencias idiomáticas y/o culturales* (F25) con valores “Alta” o “Muy Alta”, que exigiría un LEL del UdeD de mejor calidad y, por ende, técnicas de verificación más rigurosas.

- *Validar Léxico*

La actividad de este bloque de proceso apunta a encontrar defectos en los símbolos descritos y corroborar la definición de los símbolos en el UdeD. Al igual que en la verificación, la *Fiabilidad necesaria del software* (F14) es un parámetro de este bloque, que se utiliza para seleccionar las técnicas de validación. Por ejemplo, si la exigencia es menor se podrían realizar entrevistas con los usuarios. A mayor efectividad exigida en la detección de defectos, se podrán realizar revisiones más exhaustivas con los usuarios que incluyan distintos puntos de vista. También es indispensable una alta exigencia de calidad en el LEL del UdeD cuando existen *Diferencias idiomáticas y/o culturales entre los involucrados* (F25) con valores “Media”, “Alta” o “Muy Alta”. Además, si la *Localización de usuarios* es distribuida (F10) o hay *Diferencias temporales entre los involucrados* (F26) con valores “Media”, “Alta” o “Muy Alta” deberá establecerse la conveniencia de técnicas de validación que no requieran comunicación sincrónica entre los involucrados, o considerar la posibilidad de traslado de los ingenieros de requisitos a distintas zonas de asentamiento de los usuarios, en función también del factor F14. En todos los casos, las técnicas de validación a aplicar deben ser seleccionadas tomando en consideración la *Experiencia en el proceso de requisitos* (F24) del equipo.

- *Validar contra Documentación del Mercado*

Cuando se utiliza este bloque de proceso no existen clientes ni usuarios concretos con quienes interactuar. Por lo tanto, se valida el léxico contra la documentación del mercado u otras posibles fuentes de información identificadas al *Planear Estudio de Mercado*. El esfuerzo aplicado en esta etapa también depende del grado de *Fiabilidad necesaria del software* (F14).

4.3.3 Variantes al Construir Escenarios Actuales

El Punto de Variación 2 se relaciona con el subproceso *Construir Escenarios Actuales* que se representa en la Figura 4-6. Se puede notar que el bloque *Derivar Escenarios Actuales* puede omitirse o ser reemplazado en las siguientes situaciones:

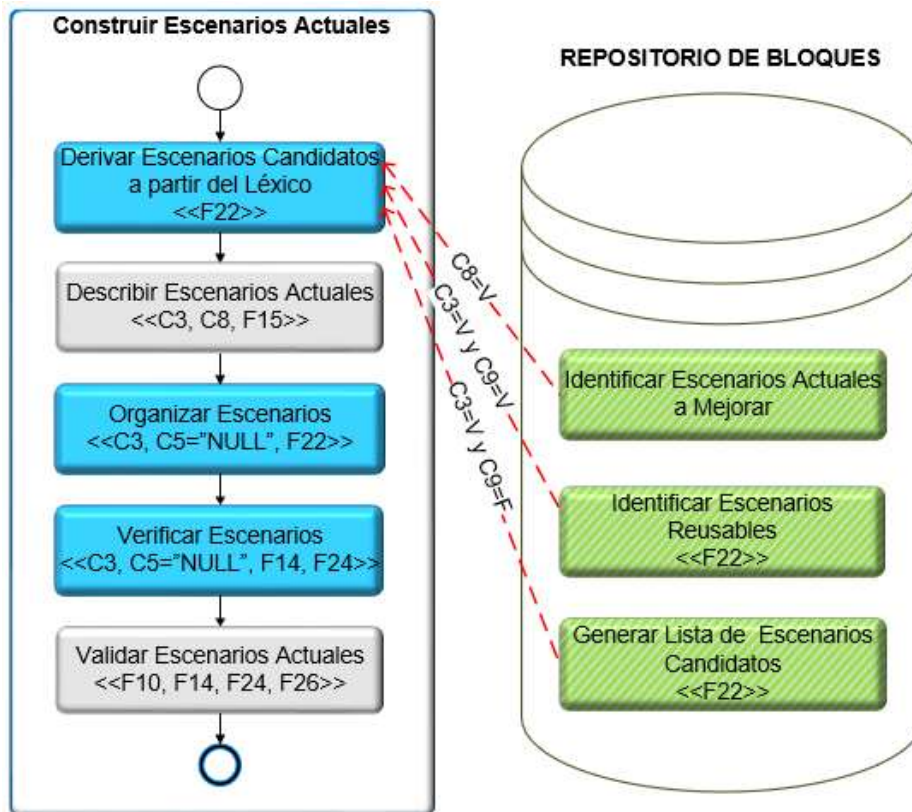


Figura 4-6. Variantes al Construir Escenarios Actuales

- Cuando se está trabajando en una mejora a una funcionalidad en un modelo de proceso incremental (ver condición C8 en subsección 4.3.2) el bloque *Derivar Escenarios Actuales* es reemplazado por el bloque *Identificar Escenarios Actuales a Mejorar*.
- En caso de no disponer del LEL del UdeD (ver condición C3 en la subsección 4.3.1), el reemplazo depende de la existencia o no de EA para reuso. Si hay escenarios de proyectos previos para reutilizar, el bloque de proceso *Derivar Escenarios Actuales* se reemplaza por el bloque *Identificar Escenarios Reusables*. En cambio, si no se dispone de dichos escenarios, se reemplaza por el bloque *Generar Lista de Escenarios Candidatos*. Las condiciones para evaluar son las siguientes:

Condición C3: es de arrastre, si es verdadera indica que no se ha creado el LEL del UdeD

Condición C9:

Reuso de artefactos de requisitos existentes (F15) = {EA}

Cada uno de los bloques que conforma este subproceso es definido a continuación indicando la influencia de los factores situacionales sobre estos:

- *Derivar Escenarios Candidatos a partir del Léxico*

Se trata de un bloque de proceso compartido, que se puede utilizar como un bloque de primer nivel en el subproceso *Construir Escenarios Actuales*, y también como un bloque interno del subproceso *Describir Escenarios Futuros con Enfoque Basado en el Léxico*. Consiste en derivar los escenarios utilizando la información contenida en el LEL del UdeD construido en la etapa anterior. Para ello se identifican los actores del UdeD, se genera una lista de escenarios candidatos y estos se completan tomando la información almacenada en el léxico (ver subsección 2.3.2). El factor *Modelo de proceso de software* (F22) es un parámetro determinante para definir el alcance de esta actividad. En un modelo “Cascada” o “Transformación Formal” se derivan todos los símbolos del léxico. En un modelo “Iterativo” esta actividad se realiza en ciclos de refinamientos sucesivos según lo planificado para la iteración. Por su parte, en un modelo “Incremental”, se derivan solo los símbolos relacionados con la nueva funcionalidad asociada al incremento.

- *Identificar Escenarios Actuales a Mejorar*

Se debe tomar como base los EA construidos en los incrementos previos. El propósito es identificar aquellos escenarios relacionados a la funcionalidad a mejorar, que necesiten ser refinados, para posteriormente incorporar mayor detalle en sus descripciones.

- *Identificar Escenarios Reusables*

Se determina cuáles son los escenarios candidatos a ser reutilizados. Es importante prestar atención al factor *Modelo de proceso de software* (F22) para definir el alcance de esta actividad. Si se aplica un modelo “Cascada” o uno de “Transformación Formal”, se incluyen en el listado aquellos EA asociados al alcance definido para todo el sistema. Si se trata de un proceso “Iterativo” se va

a ampliar el listado definido en ciclos anteriores, considerando el alcance de este ciclo. En un modelo de proceso “Incremental” se identifican los escenarios a reutilizar referidos al alcance definido para el incremento.

- *Generar Lista de Escenarios Candidatos*

Se generan los escenarios candidatos tomando como punto de partida el objetivo y alcance establecido para el sistema software a construir. Cada funcionalidad principal del alcance puede desglosarse y ser un posible EA. Se debe tener presente, que, dado que la condición C3 es verdadera, este bloque se utiliza cuando el equipo de requisitos tiene basta experiencia en el dominio, con lo cual podrán incluirse en el listado otros escenarios candidatos que por experiencia en el dominio se sabe que se desarrollarán dentro del UdeD. Un aspecto a considerar para determinar el alcance de esta actividad es el *Modelo de proceso de software* que se aplica en el proyecto (F22). Si se aplica un modelo “Cascada” o uno de “Transformación formal”, se incluyen en la lista los escenarios para el alcance definido para todo el sistema. Si se trata de un proceso “Iterativo” se va a ampliar el listado definido en ciclos anteriores, considerando el alcance de este ciclo. En un modelo de proceso “Incremental” se identifican los escenarios candidatos para el alcance definido para el incremento.

- *Describir Escenarios Actuales*

Este es un bloque que siempre se utiliza para la creación de los EA. En esta actividad se van completando los escenarios candidatos identificados en el bloque anterior, y se pueden agregar nuevos escenarios en el alcance correspondiente. Si los EA fueron derivados del LEL del UdeD (C3=F, ver condición en la subsección 4.3.1), las descripciones deben contener los vínculos a los símbolos de dicho léxico. En todos los casos en que se disponga de un LEL del UdeD, independientemente de que este se haya creado en el ciclo en curso o en uno previo según el modelo de proceso, las descripciones de los EA incorporan vínculos a los símbolos de dicho modelo. Un aspecto para tener en cuenta es el factor *Reuso de artefactos de requisitos existentes* (F15=EA). Si se dispone de EA de reuso, algunos de estos escenarios pueden requerir algún tipo de actualización. En cambio, si los escenarios candidatos tienen su origen en el documento de objetivos y alcance, las descripciones se hacen bajo un enfoque

top-down. Es decir, se establece primero el objetivo de cada escenario y su contexto, se identifica un episodio de comienzo y uno de fin, para luego ir describiendo los episodios restantes. Por otra parte, en un modelo de proceso incremental, para el que se ha planificado una mejora a cierta funcionalidad (C8=V, ver condición en la subsección 4.3.2), se deben mejorar los escenarios afectados identificados previamente, probablemente incluyendo cambios o incorporando más detalles.

- *Organizar Escenarios*

Se trata de un bloque de proceso compartido que se aplica en los subprocesos *Construir Escenarios Actuales* y *Construir Escenarios Futuros*. Una vez definido el conjunto de escenarios estos deben ser reorganizados e integrados. Para esta actividad se debe considerar la condición C3, que determina si se ha construido o no el LEL del UdeD, y la condición C5, que define si se va a construir o no el LEL del Sistema (ver condiciones C3 y C5 en subsección 4.3.1). Cuando este bloque se utiliza en el subproceso *Construir Escenarios Actuales* la condición C5 no aplica, por tal razón, como se muestra en la Figura 4-6, se le asigna el valor "NULL". Si se dispone del LEL del UdeD los escenarios que se obtienen, luego de aplicar las operaciones de composición y descomposición como también de integrar, contienen los vínculos correspondientes a los símbolos del léxico. El alcance de la actividad a realizar en este bloque depende del *Modelo de proceso software* en que se basa el proyecto (F22). Si se está aplicando un modelo "Cascada" o de "Transformación Formal", el producto resultante será el conjunto completo de todos los escenarios integradores. En un modelo "Incremental", se aplican las operaciones y se realiza la integración para los escenarios afectados por el incremento, lo nuevo se acopla a los escenarios integradores de los incrementos anteriores. Cabe señalar que bajo este modelo de proceso si se trata de una mejora a una funcionalidad, es probable que se obtengan escenarios integradores muy similares a los ya existentes. En cambio, si el modelo de proceso es "Iterativo", se aplican operaciones e integración a todo el conjunto de escenarios que se tenga disponible.

- *Verificar Escenarios*

Se trata de un bloque de proceso compartido, se utiliza tanto en el subproceso *Construir Escenarios Actuales* como también en *Construir Escenarios Futuros*.

Esta actividad consiste en asegurar la consistencia interna y completitud de cada escenario y además la integridad entre el conjunto de escenarios producido. Al verificar se considera el modelo léxico vinculante, por lo cual se tiene presente la condición C3, que determina si se ha construido el LEL del UdeD, y la condición C5, que define si se dispone del LEL del Sistema (ver ambas condiciones en subsección 4.3.1). Asimismo, si se sigue un modelo de proceso con ciclos puede ocurrir que se deba utilizar durante la verificación un modelo léxico vinculante que proviene de un ciclo anterior. Cuando este bloque se utiliza en el subproceso *Construir Escenarios Actuales* la condición C5 no aplica, por tal razón, como se visualiza en la Figura 4-6, se le asigna el valor "NULL". Dependiendo del léxico utilizado en la descripción de los escenarios se verifica, por ejemplo, el uso correcto de la semántica de los símbolos referenciados por el escenario o que todo símbolo del léxico se utilice al menos en un escenario, entre otros controles. En este bloque se considera el nivel de *Fiabilidad necesaria del software* (F14), para seleccionar las técnicas de verificación más convenientes. Si la exigencia es menor se pueden aplicar técnicas como walkthroughs o análisis con checklist. En cambio, si el factor toma valor "Muy Alta" o "Alta" se podría aplicar una técnica o una combinación de técnicas más rigurosas, tal como inspecciones. Ciertas técnicas de verificación pueden requerir más o menos experiencia en su implementación por lo cual para la selección de las mismas se considera el factor *Experiencia en el proceso de requisitos* (F24).

- *Validar Escenarios Actuales*

El propósito de esta actividad es confirmar que el conjunto de EA generados está en conformidad con las situaciones del UdeD. Es importante considerar la *Localización de usuarios* (F10). En especial cuando los usuarios están distribuidos geográficamente, habrá que buscar técnicas que sean aplicables a esta modalidad de trabajo, por ejemplo, la lectura de revisión por parte de los actores del UdeD para posteriormente realizar en reuniones remotas la discusión de los escenarios en cuanto a las diferentes modalidades locales de trabajo. Adicionalmente, se tiene en cuenta la *Fiabilidad necesaria del software* (F14). Si hay baja exigencia se pueden aplicar técnicas menos estructuradas como la lectura ad-hoc. En contrapartida, a mayor exigencia, se aplicará mayor rigor en

el proceso, por ejemplo, podría requerirse más reuniones o aumentar la cantidad de clientes y usuarios que participen en la revisión. Por otra parte, si hay *Diferencias temporales entre los involucrados* (F26) con valores “Media”, “Alta” o “Muy Alta”, deberá establecerse la conveniencia de técnicas de validación que no requieran comunicación sincrónica entre los involucrados. En todos los casos se considera el nivel de *Experiencia en el proceso de requisitos* (F24) del equipo para al determinar qué técnicas de validación pueden ser implementadas.

4.3.4 Variantes al Construir Escenarios Futuros

La Figura 4-7 muestra el subproceso *Construir Escenarios Futuros*, asociado al Punto de Variación 3.

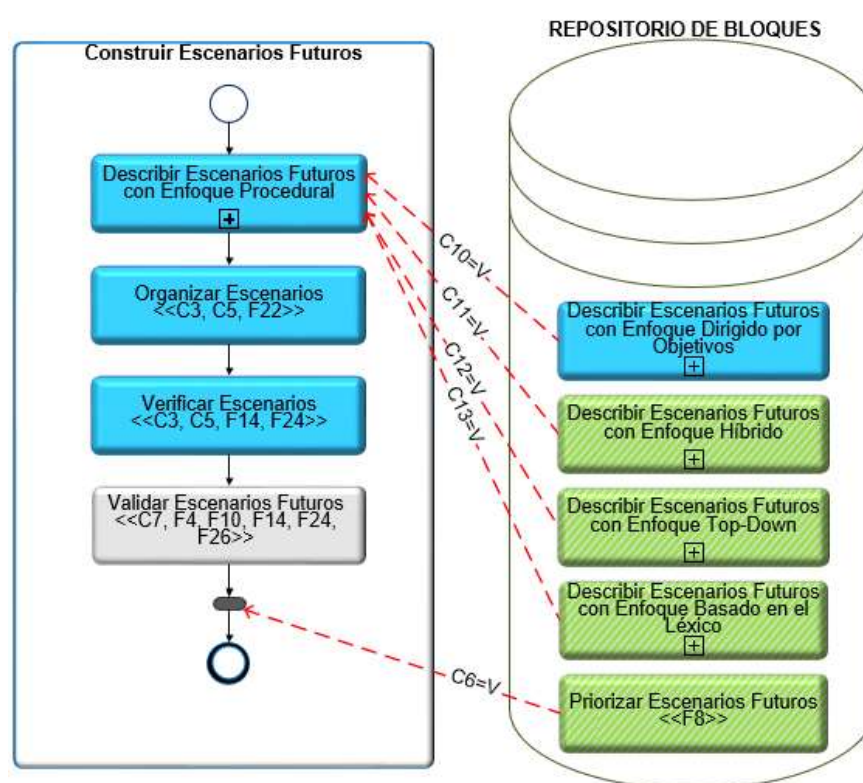


Figura 4-7. Variantes al Construir Escenarios Futuros

Como se puede observar en la Figura 4-7, la principal variante se presenta al momento de determinar el enfoque con el que se van a describir los EF. Para ello se debe prestar atención al nivel de *Reingeniería en el proceso de negocio* (F4). En los enfoques procedural, dirigido por objetivos e híbrido se toma como punto de inicio los EA para la construcción de los EF. La existencia de los EA queda determinada con la condición C4. En los enfoques top-down y basado en léxico, se considera además la condición C3, que define si se ha creado o no el LEL del

UdeD (las condiciones C3 y C4 fueron explicadas en la subsección 4.3.1). A continuación, se detallan las situaciones que determinan la adopción de cada uno de los cinco enfoques:

- Cuando se dispone de EA (C4=F) y el proyecto de software tiene un bajo nivel de cambios esperado (F4 con valoración “Baja” o “Muy Baja”), es decir, los procesos de negocio se mantienen casi inalterados, se conserva el bloque de proceso *Describir Escenarios Futuros con Enfoque Procedural*.
- Si se han generado los EA, pero se espera que los procesos de negocio del UdeD actual cambien significativamente con la implantación del futuro software. Entonces el bloque de proceso *Describir Escenarios Futuros con Enfoque Procedural* se reemplaza por el bloque *Describir Escenarios Futuros con Enfoque Dirigido por Objetivos*. La condición que se debe cumplir es:

Condición C10:

Condición C4=F Y

Reingeniería en el proceso de negocio (F4) = {*Muy Alta, Alta*}

- Hay situaciones en las que se construyeron los EA, y se espera un nivel medio de cambios en los procesos del negocio. Esto implica que algunos procesos se mantendrán iguales o sufrirán cambios mínimos, mientras que otros, tendrán cambios profundos, incluso podrán no existir en el contexto futuro. Entonces el bloque de proceso *Describir Escenarios Futuros con Enfoque Procedural* se reemplaza por el bloque *Describir Escenarios Futuros con Enfoque Híbrido*. Esta situación queda determinada con la siguiente condición:

Condición C11:

Condición C4=F Y

Reingeniería en el proceso de negocio (F4) = {*Media*}

- Puede pasar que se haya omitido la construcción de los EA y también del LEL del UdeD. Entonces, el bloque *Describir Escenarios Futuros con Enfoque Procedural* se debe reemplazar por el bloque *Describir Escenarios Futuros con Enfoque Top-Down*. Otro caso en el que tiene lugar dicho reemplazo es cuando sin haberse construido EA sí se dispone del LEL del UdeD, pero se espera un alto nivel de reingeniería de los procesos

organizacionales. Este reemplazo se determina en base a la condición que sigue:

Condición C12:

(Condición C3=V Y Condición C4=V) O

(Condición C3=F Y Condición C4=V) Y

Reingeniería en el proceso de negocio (F4) = {*Muy Alta, Alta*}

- En determinados casos se ha omitido la creación de los EA, pero se dispone del LEL del UdeD y la reingeniería esperada en los procesos de negocio es baja o no aplica al contexto¹⁴ por tratarse de un sistema software para un mercado potencial o para un negocio nuevo. En tal situación el bloque de proceso *Describir Escenarios Futuros con Enfoque Procedural* se debe reemplazar por el bloque *Describir Escenarios Futuros con Enfoque basado en el Léxico*. Este enfoque se utiliza cuando se cumple la siguiente condición:

Condición C13:

(Condición C3=F Y Condición C4=V) Y

Reingeniería en el proceso de negocio (F4) = {*Medía, Baja, Muy Baja, No Aplica*}

Posteriormente, en la subsección 4.3.5, se detallan las adaptaciones internas asociadas a cada uno de estos cinco enfoques.

Por otra parte, otro ajuste del subproceso *Construir Escenarios Futuros* sucede cuando como parte de la adaptación del *Proceso Base de IR* se omite el subproceso *Especificar Requisitos* (C6=V, ver condición C6 en subsección 4.3.1). En tal caso, dado que no hay requisitos para priorizar, se inserta el bloque *Priorizar Escenarios Futuros*, como se representa en la Figura 4-7.

A continuación, se define cada uno de los bloques de proceso necesarios para componer el subproceso *Construir Escenarios Futuros*, considerando la influencia interna de los factores situacionales sobre dichos bloques:

- *Describir Escenarios Futuros con Enfoque Procedural*

¹⁴ Se debe tener presente que si el factor *Tipo de Cliente* es "Mercado Potencial" (F2) o el valor de *Novedad del Negocio* es "Si" (F3), el factor *Reingeniería en el proceso de negocio* (F4) no se debe completar, se considera con valor "No Aplica" (ver dependencias invalidantes en la Tabla 3.4).

Se derivan los EF directamente de los EA existentes. Este es un subproceso compartido, los detalles de las adaptaciones internas del mismo se definen en la subsección 4.3.5.1.

- *Describir Escenarios Futuros con Enfoque Dirigido por Objetivos*

Implica derivar los EF a partir de los objetivos establecidos para el software a construir, apoyándose en los EA existentes. Este es un subproceso compartido, los detalles de las adaptaciones internas del mismo se definen en la subsección 4.3.5.2.

- *Describir Escenarios Futuros con Enfoque Híbrido*

Se evalúa el grado esperado de reingeniería del negocio para cada escenario. Según corresponda, se describirá cada EF aplicando un enfoque procedural o dirigido por objetivos. Las particularidades y adaptaciones internas de este subproceso se detallan en la subsección 4.3.5.3.

- *Describir Escenarios Futuros con Enfoque Top-Down*

Los EF se generan tomando como punto de partida los objetivos del sistema. Este subproceso y sus adaptaciones internas se describen en la subsección 4.3.5.4.

- *Describir Escenarios Futuros con Enfoque Basado en Léxico*

Se derivan los EF a partir del LEL del UdeD. Para ello se aplica la técnica de derivación de información del léxico utilizada para construir los EA. Los detalles internos de este subproceso se describen en la subsección 4.3.5.5.

- *Organizar Escenarios*

Este bloque de proceso compartido ha sido detallado en la subsección 4.3.3, y siempre se conserva. A modo particular, cuando se utiliza en el subproceso *Construir Escenarios Futuros*, tiene sentido la validación de las condiciones C3 y C5, detalladas en la subsección 4.3.1. En función al resultado de estas condiciones, pueden darse tres situaciones: que se disponga del LEL del UdeD y del LEL del Sistema, que sólo se cuente con uno de los dos léxicos o, que se haya omitido la construcción de ambos. Entonces, si se dispone del LEL del Sistema (C5=F), los escenarios integradores deberán utilizar los símbolos de dicho léxico incorporando los vínculos correspondientes. Si solo se dispone del LEL del UdeD (C3=F y C5=V), se toman los términos de éste. Lógicamente, ante

la ausencia de ambos léxicos los escenarios no contendrán vínculos (C3=V y C5=V). El alcance de las actividades de este bloque está definido en función al *Modelo de proceso software* en que se basa el proyecto (F22), donde aplican las mismas consideraciones que para la construcción de los EA.

- *Verificar Escenarios*

Este bloque de proceso compartido se describió previamente en la subsección 4.3.3, y siempre se conserva. En especial, cuando se utiliza el bloque en el subproceso *Construir Escenarios Futuros*, cobra significado la validación de las condiciones C3 y C5, detalladas en la subsección 4.3.1. Esto es debido a que en este subproceso las descripciones de los EF pueden contener vínculos al LEL del Sistema, al LEL del UdeD o no incluir vínculos. Por lo tanto, se realizan las verificaciones correspondientes en base al léxico utilizado para describir los EF. Las técnicas a aplicar deben adaptarse para el modelo de EF. Por ejemplo, además de los controles de consistencia y de completitud, se deberán incluir algunos ítems, como por ejemplo, asegurar que todo objetivo del sistema esté satisfecho por uno o más EF. En cuanto al nivel de *Fiabilidad necesaria del software* (F14) y al grado de *Experiencia en el proceso de requisitos* (F24), las consideraciones son las mismas que para construcción de los EA, para seleccionar las técnicas en función a la exigencia existente y a la experiencia del equipo.

- *Validar Escenarios Futuros*

Este bloque de proceso siempre forma parte del proceso de IR, siendo un bloque particular de este subproceso *Construir Escenarios Futuros*. Es muy importante la correcta selección de las técnicas a aplicar debido a que los EF contienen la funcionalidad y las restricciones bajo las que va a funcionar el sistema software. Entre los aspectos a tener presente para realizar esta actividad, se encuentra la *Reingeniería en el proceso de negocio* (F4) y la *Fiabilidad necesaria del software* (F14). En un contexto en el que se introducen pocos cambios y con baja exigencia de fiabilidad tal vez sea suficiente con reuniones en las que se aplique la técnica de revisión. En cambio, cuando la valoración del factor F4 es “Alta” o “Muy Alta”, lo cual implica que se van a introducir cambios significativos con respecto a la forma actual de desarrollar las tareas, o se espera un alto grado de fiabilidad, será preferible aplicar otras

técnicas, como el prototipado o el uso de storyboards, las cuales proporcionan una visión más precisa de lo que será el contexto futuro. La *Localización de usuarios* (F10) también se considera, pues si se debe validar los escenarios con usuarios distribuidos geográficamente, habrá técnicas que serán imposibles de aplicar, por ejemplo, el role playing. Así también si hay *Diferencias temporales entre los involucrados* (F26) con valores “Media”, “Alta” o “Muy Alta” es conveniente seleccionar técnicas de validación que no requieran comunicación sincrónica entre los involucrados. Por otra parte, si el sistema software es para un mercado potencial o tiene que ver con un negocio nuevo (C7=V, ver condición en el apartado 4.3.2) puede no haber usuarios, con lo cual se deberán aplicar otras técnicas de validación según las fuentes de información utilizadas. En todos los casos, las técnicas de validación se seleccionan en función al nivel de *Experiencia en el proceso de requisitos* (F24), dado que algunas técnicas pueden requerir más o menos experiencia por parte del equipo de ingenieros de requisitos para su implementación.

- *Priorizar Escenarios Futuros*

La actividad de este bloque consiste en evaluar la criticidad y urgencia de las situaciones futuras detalladas en los EF y otorgarles prioridades a estos. Si el factor *Conflicto de intereses de usuarios* (F8) tiene valoración “Alto” o “Muy Alto”, pueden surgir algunos conflictos entre los clientes y usuarios con relación a las prioridades a asignar, se requerirá más negociación para lograr un acuerdo al respecto.

4.3.5 Variantes del Subproceso Describir Escenarios Futuros

En la subsección anterior se ha resaltado la importancia de adoptar el enfoque de descripción de EF más conveniente de acuerdo con el contexto. Esta actividad involucra la identificación de los escenarios candidatos para luego describirlos. El modo de proceder en cada enfoque dependerá principalmente de los objetivos del sistema, de los EA y del nivel de cambios esperado en los procesos.

Como se ha mencionado antes, tanto el enfoque “procedural”, como también los enfoques “dirigido por objetivos” e “híbrido” se pueden utilizar cuando se han construido los EA, con lo cual estos sirven de base para describir los EF. En cambio, los enfoques “basado en el léxico” y “top-down” fueron concebidos para

describir los EF en aquellos casos en los que no se dispone de los EA. A continuación, se definen los subprocesos correspondientes a cada uno de los cinco enfoques para la descripción de los EF.

4.3.5.1 Describir Escenarios Futuros con Enfoque Procedural

La Figura 4-8 representa el subproceso que se debe aplicar para describir los EF bajo un enfoque orientado a los procedimientos. Se trata de un subproceso compartido que se puede utilizar como un bloque de primer nivel en el subproceso *Construir Escenarios Futuros*, y también como un bloque interno del subproceso *Derivar Escenarios Futuros con Enfoque Híbrido*. Como se puede observar, la adaptación principal de este subproceso ocurre cuando se trabaja en una mejora a determinada funcionalidad en el contexto de un modelo de proceso incremental (ver condición C8 en la subsección 4.3.2), en tal caso, el bloque *Derivar Escenarios Futuros Integradores a partir de Escenarios Actuales Integradores* es reemplazado por el *Bloque Nulo*. Seguidamente se describen los bloques de proceso internos involucrados en este subproceso:

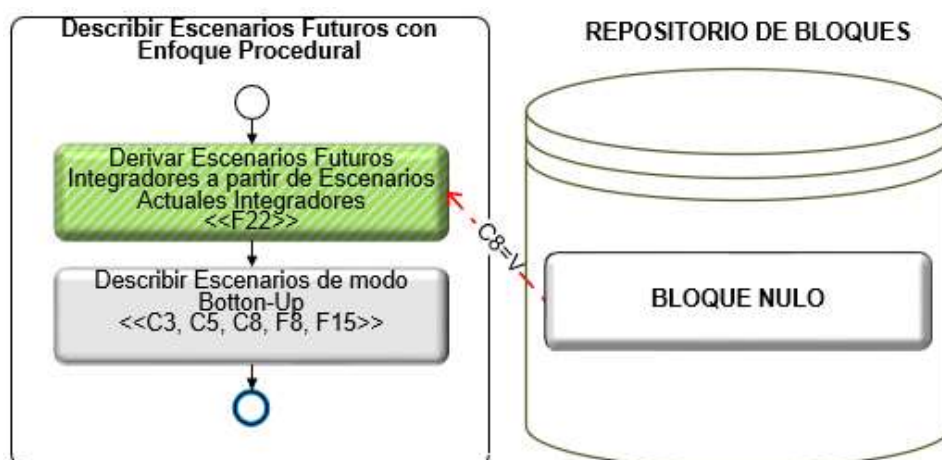


Figura 4-8. Subproceso Describir Escenarios Futuros con Enfoque Procedural

- *Derivar Escenarios Futuros Integradores a partir de Escenarios Actuales Integradores*

En este enfoque los objetivos de los EF básicamente no cambian con respecto de los EA. Por consiguiente, se deben derivar los EF integradores a partir de los EA Integradores, donde prácticamente consiste en realizar una copia de estos. El alcance de esta actividad se determina en base al *Modelo de proceso de software* adoptado (F22). En los modelos “Cascada” y de “Transformación

Formal” se derivan todos los EF integradores para el alcance planificado. En un modelo “Iterativo” en cada ciclo se produce un avance en la derivación, aportando mayor detalle a lo iniciado en iteraciones anteriores o derivando partes nuevas en los escenarios integradores. En cambio, en un modelo “Incremental” solo se derivan los EF Integradores relacionados a la nueva funcionalidad a construir.

- *Describir Escenarios de modo Botton-Up*

Este es un bloque de proceso siempre utilizado en el enfoque procedural. Los EF se describen enfocándose en los episodios de los EA. Cada episodio se examina tomando en cuenta los objetivos del sistema para detectar si es necesaria alguna modificación. Luego se revisan las excepciones y restricciones, los demás componentes del escenario, finalmente, el título y el objetivo por si es necesario algún cambio menor. Si se genera el LEL del Sistema (C5=F), las descripciones contienen los vínculos a los símbolos de dicho léxico. En cambio, si sólo se dispone del LEL del UdeD (C3=F y C5=V), se utilizan los símbolos de este y se los vincula. En otras situaciones, las descripciones no contienen vínculos a léxicos (C3=V y C5=V). Las condiciones C3 y C5 se detallan en la subsección 4.3.1. En todos los casos en que se disponga de un modelo léxico vinculante, independientemente de que dicho modelo se haya creado en el ciclo en curso o en uno previo para modelos de proceso con ciclos, las descripciones de los EF incorporan los vínculos a dicho léxico. Bajo el enfoque procedural raramente el *Conflicto de intereses de usuarios* (F8) puede darse en relación a las funcionalidades esperadas en el sistema. A pesar de ello, puede surgir alguna contrariedad con respecto al modo de realizar alguna tarea en particular. Por consiguiente, si el factor tiene un valor relevante, tal como “Alto” o “Muy Alto”, se deben tomar las medidas pertinentes a fin de negociar las diferencias y lograr un consenso entre los usuarios. Por otra parte, otro factor a considerar es el *Reuso de artefactos de requisitos existentes* (F15=EF). Si se dispone de EF para reutilizar, se tienen que identificar los escenarios candidatos a reusar, y realizar las adaptaciones a los EF o incorporaciones que fuesen necesarias. El alcance de los escenarios a describir queda establecido por los EF Integradores creados previamente. Sin embargo, si se trabaja en una mejora a alguna funcionalidad (C8=V, ver condición en la subsección 4.3.2), solo se mejoran las

descripciones de los EF asociados a dicha funcionalidad siguiendo también el enfoque botton-up dentro de cada EF.

4.3.5.2 Describir Escenarios Futuros con Enfoque Dirigido por Objetivos

El subproceso para describir los EF bajo un enfoque dirigido por objetivos se grafica en la Figura 4-9. Se trata de un subproceso compartido que se puede utilizar como un bloque de primer nivel en el subproceso *Construir Escenarios Futuros*, y también como un bloque interno del subproceso *Derivar Escenarios Futuros con Enfoque Híbrido*. La adaptación principal de este subproceso ocurre cuando se trabaja en una mejora a determinada funcionalidad en el contexto de un modelo de proceso incremental (ver condición C8 en la subsección 4.3.2). En tal caso, el bloque *Generar Escenarios Generales a partir de los Escenarios Actuales Integradores* se reemplaza por el bloque *Regenerar Escenarios Futuros Integradores Existentes*. A continuación, se describen los bloques de proceso internos que componen el subproceso indicando en cada uno la influencia que tienen los factores situacionales:

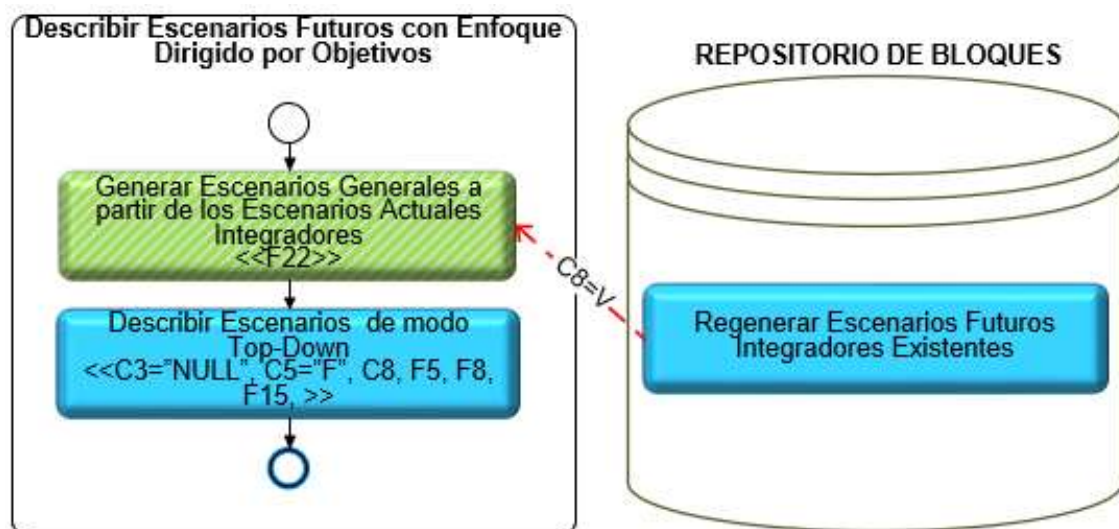


Figura 4-9. Subproceso Describir Escenarios Futuros con Enfoque Dirigido por Objetivos

- *Generar Escenarios Generales*¹⁵ a partir de los Escenarios Actuales Integradores:

¹⁵ Escenarios Generales son aquellos escenarios de alto nivel de detalle, construidos antes de tener escenarios que describen situaciones particulares. Tanto los Escenarios Integradores como los Escenarios Generales contienen descripciones de alto nivel que brindan una visión global del UdeD. Aunque se diferencian en que los Escenarios Integradores se construyen una vez que se han definido los escenarios particulares mediante un mecanismo de integración.

El punto de partida para la actividad de este enfoque es los EA Integradores. Dado que se espera un nivel de cambios alto con respecto a los EA, se revisan cada uno de los EA Integradores para comprobar si es compatible con el objetivo del sistema a construir, es decir, si existirá en el UdeD futuro. En tal caso se crea un EF general que describa un proceso de negocio, donde los episodios serán una versión preliminar de los títulos de los EF particulares por definir. Debido al alto grado de reingeniería, pueden surgir EF generales no asociados al contexto actual, pero necesarios para alcanzar los objetivos específicos esperados. El alcance de las actividades de este bloque se determina según el *Modelo de proceso de software* (F22). En los modelos “Cascada” y “Transformación Formal” se genera el conjunto completo de EF generales según el alcance planificado. En un modelo “Iterativo”, cada iteración trae aparejado un progreso para los EF generales, completando los realizados previamente. En el caso de un modelo “Incremental”, solo se generan los EF generales para la nueva funcionalidad planificada.

- *Regenerar Escenarios Futuros Integradores Existentes*

Este es un bloque de proceso compartido que se puede utilizar en los subprocesos *Describir Escenarios Futuros con Enfoque Dirigido por Objetivos* y *Describir Escenarios Futuros con Enfoque Top-Down*. Se toma el conjunto de EF Integradores que se crearon en incrementos previos para identificar los EF Integradores alcanzados por la mejora planificada para este incremento. Cada escenario que se identifica es revisado, bajo la perspectiva de la mejora a realizar, con el fin de aplicar las modificaciones de alto nivel, de ser necesarias.

- *Describir Escenarios de modo Top-Down*

Este es un bloque de proceso compartido que se usa en los subprocesos *Describir Escenarios Futuros con Enfoque Dirigido por Objetivos*, *Describir Escenarios Futuros con Enfoque Top-Down* y *Describir Escenarios Futuros con Enfoque Basado en el Léxico*. Para describir los escenarios se revisa el título (mencionado en el EF general o integrador según el bloque previo) y se escribe el objetivo de cada EF a fin de asegurar su conformidad con los objetivos específicos que satisfará el sistema. Se describe el contexto, luego se elicitán o proponen los episodios, las excepciones y restricciones, y se completan los demás componentes del escenario. En este bloque se consideran la condición

C3, que determina si se ha construido o no el LEL del UdeD, y la condición C5, que define si se construye o no el LEL del Sistema (ver definiciones de las condiciones C3 y C5 en subsección 4.3.1). En función al resultado de tales condiciones las descripciones pueden contener vínculos al LEL del Sistema (C5=F), al LEL del UdeD si sólo se dispone de tal léxico (C3=F y C5=V), o pueden no tener vínculos si se omite la construcción de ambos léxicos (C3=V y C5=V). Puede ocurrir que se disponga de un modelo léxico vinculante creado en un ciclo previo, en tal caso las descripciones de los EF incorporan los vínculos a dicho modelo. A modo particular, cuando el bloque se utiliza en el subproceso *Describir Escenarios Futuros con Enfoque Dirigido por Objetivos*, se espera un alto grado de cambios en los procesos de negocio. Por tal motivo, siempre se debe generar el LEL del Sistema, con lo cual las descripciones de los EF deben contener los vínculos a los términos de dicho léxico. Por lo anterior, se puede notar en la Figura 4-9 que la condición C3 tiene asignado el valor “NULL”, dado que no interesa evaluar la existencia del LEL del UdeD, y que la condición C5 tiene el valor falso (“F”). Por otra parte, otro aspecto que se tiene que considerar es el *Reuso de artefactos de requisitos existentes* (F15=EF). Si hay posibilidad de reusar EF preexistentes se identifican los escenarios o partes de estos que se puedan reutilizar, y se realizan las actualizaciones que se consideren pertinentes. El alcance de los escenarios a describir está definido por los EF Integradores o Generales generados anteriormente. Aunque, si se está trabajando en una mejora a cierta funcionalidad, se mejoran las descripciones de los EF ligados a tal funcionalidad (C8=V, ver condición en la subsección 4.3.2). Por otro lado, se considera la *Volatilidad del contexto* (F5) para evaluar la conveniencia de motivar la generación o no de soluciones alternativas. Si la valoración es “Alta” o “Muy Alta”, al construir los EF no sería recomendable motivar la generación de soluciones alternativas, dado que cualquier solución propuesta puede llegar a modificarse en cualquier momento. En particular, cuando este bloque de proceso se utiliza en el subproceso *Describir Escenarios Futuros con Enfoque Dirigido por Objetivos*, existen EA generados previamente, lo cual indica que no es posible que la volatilidad en el contexto sea “Alta” o “Muy Alta”, es decir, este factor es considerado cuando se usa en otros subprocesos. Adicionalmente, si el factor *Conflicto de intereses de usuarios* (F8)

tiene valor “Alto” o “Muy Alto”, implica una alta probabilidad de existencia de diferencias en cómo los usuarios quieren que funcione el sistema. Esta circunstancia requerirá mayor negociación, siendo conveniente la generación de propuestas alternativas, es decir, elaborar varios EF con distintas soluciones sobre los que se pueda negociar, a fin de consensuar cómo se trabajará en el futuro contexto con el sistema de software.

4.3.5.3 Describir Escenarios Futuros con Enfoque Híbrido

La Figura 4-10 representa gráficamente la conformación interna del subproceso *Describir Escenarios Futuros con Enfoque Híbrido*. El punto de partida para la descripción de los EF son los EA generados previamente. Este subproceso se aplica cuando se ha evaluado una *Reingeniería en el proceso de negocio* (F4) “Media”, lo que implica que habrá partes del proceso de negocio con un nivel alto de cambios y otras partes con un nivel bajo de cambios. Por lo tanto, partiendo de los EA Integradores, se evalúa el grado de reingeniería esperado para cada escenario. Es decir, la condición a utilizar depende de un factor interno y es la siguiente:

Condición 14

Reingeniería en la parte del proceso de negocio en evaluación = {Muy Alta, Alta, Media}

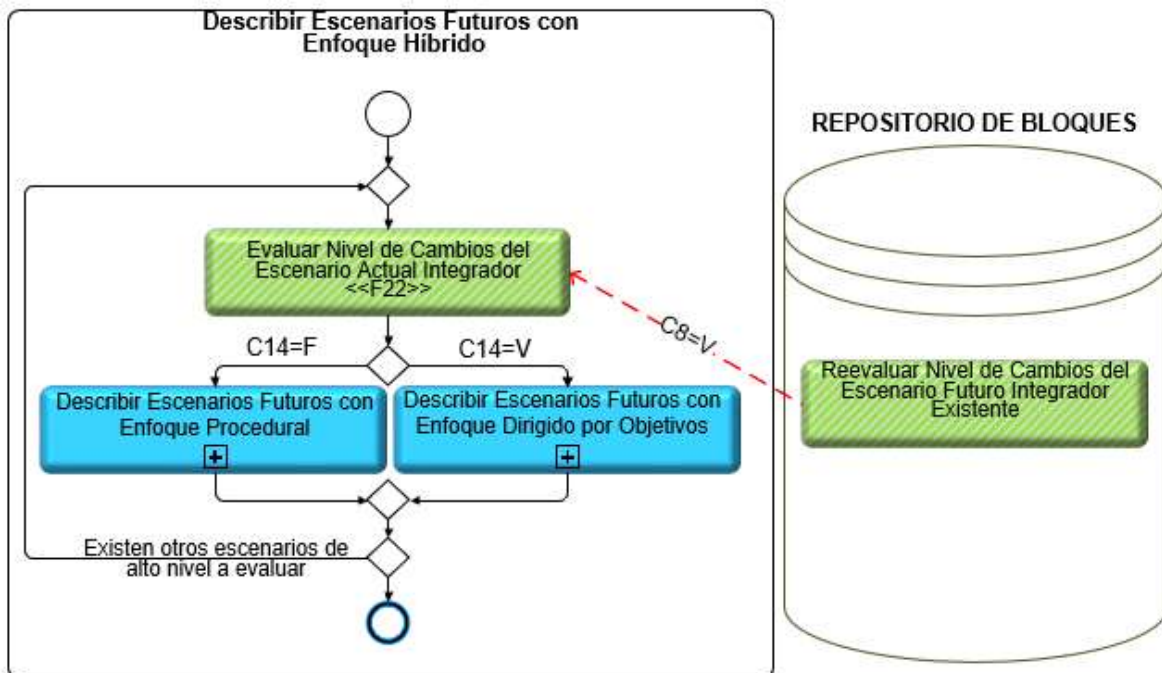


Figura 4-10. Subproceso Describir Escenarios Futuros con Enfoque Híbrido

En el caso de que la condición C14 sea falsa, se utiliza el bloque de proceso *Describir Escenarios Futuros con Enfoque Procedural* para describir los EF asociados al escenario que fue evaluado. En contrapartida, si la condición se cumple, la descripción de los EF se realiza utilizando el bloque de proceso *Describir Escenarios Futuros con Enfoque Dirigido por Objetivos*. Lo anterior se repite para cada EA Integrador.

En la Figura 4-10 también se puede observar que el bloque de proceso *Evaluar Nivel de Cambios del Escenario Actual Integrador* puede ser reemplazado por el bloque *Reevaluar Nivel de Cambios del Escenario Futuro Integrador Existente*. Este reemplazo se realiza cuando se trabaja en una mejora a determinada funcionalidad en un modelo de proceso incremental (ver condición C8 en la subsección 4.3.2).

A continuación, se describen los bloques de proceso necesarios para realizar la descripción de los EF aplicando el enfoque híbrido:

- *Evaluar Nivel de Cambios del Escenario Actual Integrador*

Esta actividad implica evaluar la reingeniería en una parte del proceso de negocio que se espera para cada EA Integrador (*Reingeniería en la parte del proceso de negocio en evaluación*). Se revisa el objetivo y el título de cada EA Integrador en función de los objetivos del sistema para determinar si el proceso del negocio que representa debe ser modificado. Dependiendo del resultado de la evaluación se describirán en el siguiente bloque los EF aplicando el enfoque que corresponda, procedural o dirigido por objetivos, en base a la condición C14 (como se muestra en la Figura 4-10). El alcance de lo que se debe evaluar en este bloque queda determinado por el *Modelo de proceso de software* (F22). En los modelos “Cascada” y “Transformación Formal” se evalúa cada escenario del conjunto de EA Integradores construidos. En un modelo “Iterativo”, se evalúan los escenarios que permiten ir completando lo realizado en el ciclo previo. En el caso de un modelo “Incremental”, solo se evalúan los EA Integradores asociados a la nueva funcionalidad planificada.

- *Reevaluar Nivel de Cambios del Escenario Futuro Integrador Existente*

Dado que los EF integradores ya existen, es necesario identificar cada escenario afectado al incremento para reevaluar el grado de reingeniería en la parte del

proceso de negocio bajo consideración proyectado a partir de la mejora planificada (*Reingeniería en la parte del proceso de negocio en evaluación*). Dependiendo del resultado de la evaluación se continúa con el proceso aplicando el enfoque que corresponda procedural o dirigido por objetivos, en función de la condición C14 (como se muestra en la Figura 4-10).

- *Describir Escenarios Futuros con Enfoque Procedural*

Las actividades asociadas a este subproceso compartido han sido detalladas en la subsección 4.3.5.1.

- *Describir Escenarios Futuros con Enfoque Dirigido por Objetivos*

Las actividades asociadas a este subproceso compartido han sido detalladas en la subsección 4.3.5.2. A modo particular bajo este enfoque la *Reingeniería en el proceso de negocio* es “Media” (F4), con lo cual se puede disponer o no del LEL del Sistema. Entonces, tiene sentido evaluar las condiciones C3 y C5 (detalladas en la subsección 4.3.1). Si se dispone del LEL del Sistema (C5=F), las descripciones de los EF contienen los vínculos a los símbolos de dicho léxico. En cambio, si sólo se dispone del LEL del UdeD (C3=F y C5=V), se utilizan los símbolos de este. Por el contrario, si no se construye ningún léxico (C3=V y C5=V), las descripciones de los EF no contienen vínculos. En este último caso si se sigue un modelo de proceso con ciclos, puede ocurrir que se disponga de un modelo léxico creado en ciclos previos, en tal caso los escenarios incorporan los vínculos a los términos de dicho modelo. Por lo anterior, se debe tener presente que cuando este bloque de proceso se utiliza bajo el enfoque híbrido, no corresponde asignar valores a los parámetros del bloque de proceso interno *Describir Escenarios de modo Top-Down* (tal como se visualizan en la Figura 4-9).

4.3.5.4 Describir Escenarios Futuros con Enfoque Top-Down

La Figura 4-11 grafica el subproceso *Describir Escenarios Futuros con Enfoque Top-Down*. Como se puede observar, hay dos condiciones asociadas a la adaptación general de dicho subproceso:

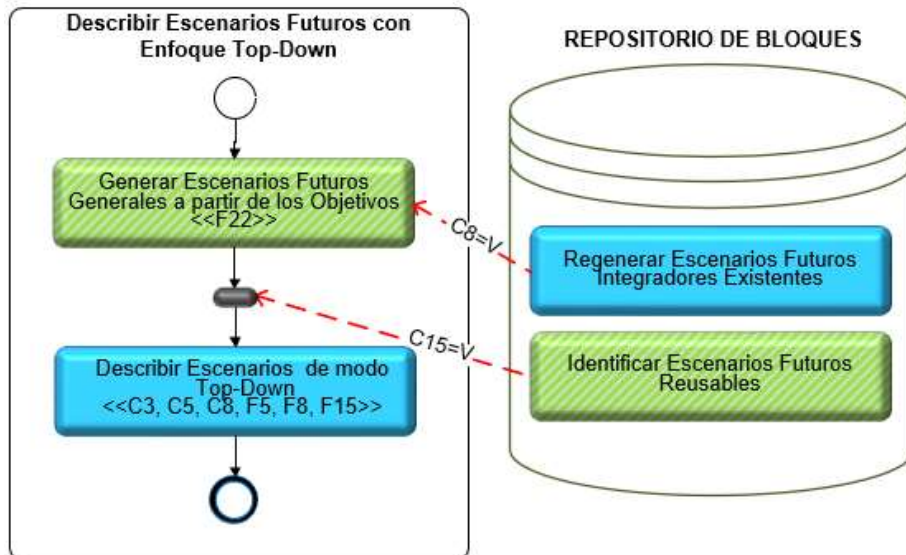


Figura 4-11. Subproceso Describir Escenarios Futuros con Enfoque Top-Down

- Cuando se ha planificado una mejora a determinada funcionalidad en el marco de un modelo de proceso incremental, se reemplaza el bloque de proceso *Generar Escenarios Futuros Generales a partir de los Objetivos* por el bloque compartido *Regenerar Escenarios Futuros Integradores Existentes*. El reemplazo se produce cuando se cumple la condición C8 detallada en la subsección 4.3.2.
- Puede darse el caso de que el sistema software a construir sea para un contexto ya existente (F3) y que se disponga de EF de proyectos previos para reutilizar (F15). En tal caso, como se representa en la Figura 4-11 se adiciona el bloque de proceso *Identificar Escenarios Futuros Reusables*. Esto se debe a que como se indicó anteriormente en este enfoque inicialmente solo se dispone de los objetivos del sistema, dado que no existe un modelo previo. Por ello, los EF para reutilizar sólo se pueden identificar después de disponer de los EF generales o de los EF Integradores existentes regenerados. La inserción del bloque se realiza cuando el incremento es para una nueva funcionalidad (F23) bajo el modelo de proceso incremental o, cuando este factor no aplica por estar siguiendo un modelo de proceso de software distinto al incremental. La condición que define esta situación es la siguiente:

Condición C15:

Novedad del contexto (F3) = {No}Y

Reuso de artefactos de requisitos existentes (F15) = {EF}Y

Tipo de Incremento (F23) = {Nueva Funcionalidad, No Aplica}

Seguidamente se describe cada uno de los bloques que componen este subproceso:

- *Generar Escenarios Futuros Generales a partir de los Objetivos*

En esta actividad se identifican los procesos de negocio bajo la perspectiva de los objetivos planteados para el sistema. Luego, por cada uno de los procesos mencionados se produce un EF general, capturando el conocimiento general del problema. En un *Modelo de proceso de software* (F22) “Cascada” o “Transformación Formal” se genera el conjunto completo de EF generales para el alcance definido. En un modelo “Iterativo” se van perfeccionando con más detalle o con partes nuevas los escenarios generales de iteraciones previas. En cambio, en un modelo “Incremental”, se generan los EF generales que cubran la nueva funcionalidad que se ha planificado.

- *Regenerar Escenarios Futuros Integradores Existentes*

Las actividades asociadas a este bloque de proceso compartido han sido detalladas en la subsección 4.3.5.2.

- *Identificar Escenarios Futuros Reusables*

En esta actividad se toma como punto de partida los escenarios generales o los escenarios integradores que se obtuvieron en el bloque que se utilizó previamente. En función a dichos escenarios se identifican los EF candidatos a ser reutilizados.

- *Describir Escenarios de modo Top-Down*

Las actividades asociadas a este bloque de proceso compartido han sido detalladas en la subsección 4.3.5.2. En particular, cuando este bloque se utiliza en el subproceso *Describir Escenarios Futuros con Enfoque Top-Down*, se puede disponer del LEL del UdeD, del LEL del Sistema, de ambos, o de ninguno. Por tal motivo, cobra sentido la evaluación de ambas condiciones C3 y C5 (detalladas en la subsección 4.3.1).

4.3.5.5 Describir Escenarios Futuros con Enfoque Basado en el Léxico

En la Figura 4-12 se representa el subproceso a utilizar para describir los EF partiendo del LEL del UdeD. Como se puede observar, la adaptación principal de este subproceso sucede cuando se ha planificado en una mejora a determinada funcionalidad siguiendo un modelo de proceso incremental (ver condición C8 en la subsección 4.3.2). Esto implica que en lugar de basarse en el LEL del UdeD hay que mejorar algunos EF que se han creado previamente. Por tal motivo, el bloque *Derivar Escenarios Candidatos a partir del Léxico* debe ser reemplazado por el *Bloque Nulo*.

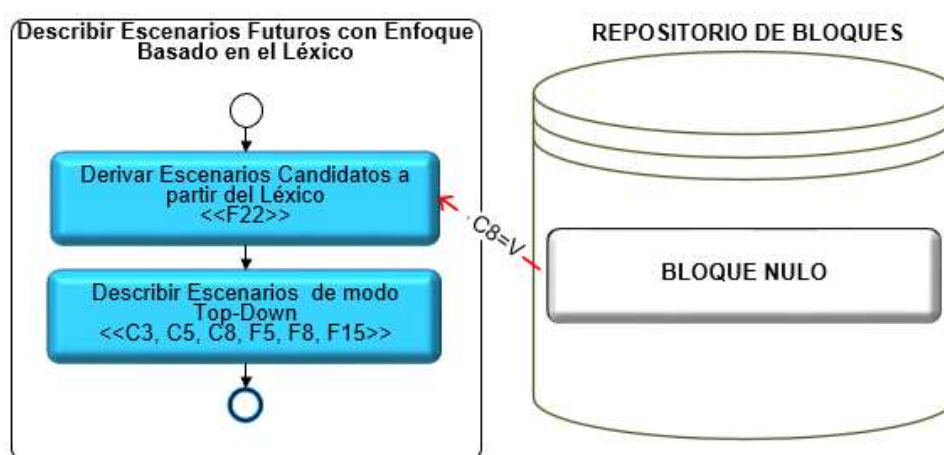


Figura 4-12. Subproceso Describir Escenarios Futuros con Enfoque Basado en el Léxico

A continuación, se describen los bloques de proceso que permiten componer este subproceso:

- *Derivar Escenarios Candidatos a partir del Léxico*
Se aplica el mismo proceso de derivación aplicado para los EA. Las actividades de este bloque de proceso compartido se han descrito detalladamente en la subsección 4.3.3.
- *Describir Escenarios de modo Top-Down*
Las actividades asociadas a este bloque de proceso compartido han sido detalladas en la subsección 4.3.5.2. A modo particular, dado que este enfoque se basa en el LEL del UdeD, no tiene sentido evaluar la existencia de tal artefacto, pues siempre la condición C3 es falsa. Si se construye el LEL del Sistema (C5=F), los EF se vinculan a tal léxico, sino al LEL del UdeD (ver condiciones C3 y C5 en apartado 4.3.1).

4.3.6 Variantes al Crear Léxico Extendido del Lenguaje del Sistema

La Figura 4-13, grafica el subproceso de *Crear Léxico Extendido del Lenguaje del Sistema*, asociado al Punto de Variación 4.

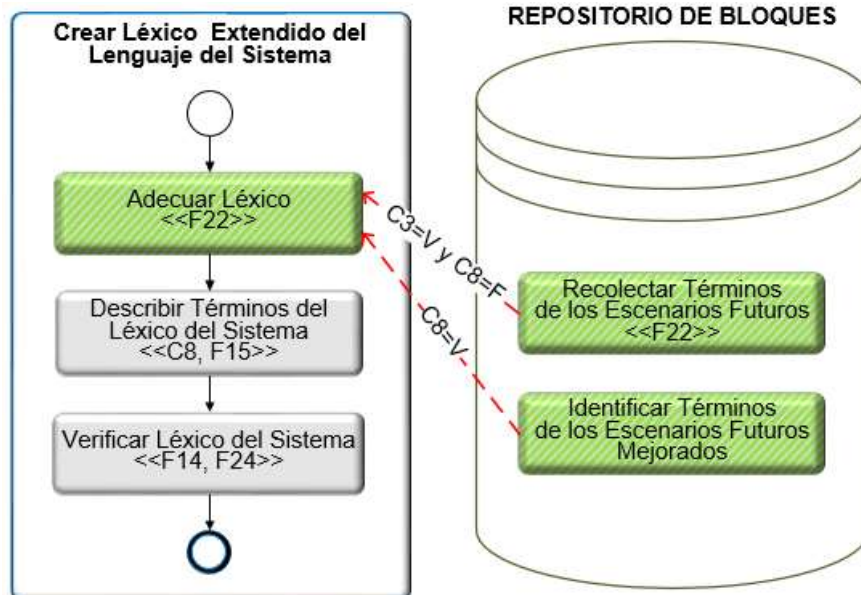


Figura 4-13. Variantes al Crear Léxico Extendido del Lenguaje del Sistema

Como se puede observar en la Figura, una adaptación de este subproceso ocurre cuando no se ha creado el LEL del UdeD ($C3=V$, ver condición en la subsección 4.3.1) siempre que no se trate de una mejora a una funcionalidad en el modelo Incremental ($C8=F$, ver condición en la subsección 4.3.2). En tal caso el bloque *Adecuar Léxico* es reemplazado por el bloque *Recolectar Términos de los Escenarios Futuros*. La otra adaptación ocurre cuando se trata de una mejora a una funcionalidad usando el modelo Incremental ($C8=V$, ver condición en la subsección 4.3.2) donde en incrementos anteriores ya se describieron símbolos que pueden ser actualizados o incorporar nuevos. En tal caso el bloque *Adecuar Léxico* es reemplazado por el bloque *Identificar Términos de los Escenarios Futuros Mejorados*.

A continuación, se definen los bloques asociados a este subproceso, indicando la manera en que los factores situacionales influyen sobre estos:

- *Adecuar Léxico*

En este bloque se toma como punto de partida el LEL del UdeD. Los términos que compondrán el LEL del Sistema se identifican a medida que se van construyendo los EF, partiendo de aquellos existentes en el LEL del UdeD.

Pueden identificarse nuevos términos necesarios para describir los cambios en los procesos de negocio, puede detectarse algún cambio en el significado de símbolos del UdeD, e incluso algunos términos pueden quedar en desuso en el contexto del sistema. Se debe tener presente que el alcance de esta actividad está dado por el LEL del UdeD creado previamente, en forma total o parcial, y dependiendo además de los EF construidos o refinados en el ciclo, dado por el modelo de proceso que se aplica (F22). En el caso de identificarse nuevos símbolos, estos deben clasificarse.

- *Recolectar Términos de los Escenarios Futuros*

Dado que este bloque se usa cuando no se ha construido un LEL del UdeD, los símbolos que compondrán el LEL del Sistema se identifican y clasifican a partir del vocabulario usado en las descripciones de los EF generados en el alcance dado por el modelo de proceso utilizado (F22), es decir, todos los EF en el modelo “Cascada” o “Transformación Formal” o los EF afectados por el ciclo en el modelo “Iterativo” o en el “Incremental” cuando involucra nuevas funcionalidades.

- *Identificar Términos de Escenarios Futuros Mejorados*

La actividad de este bloque se realiza en el marco de un modelo incremental para una mejora a determinada funcionalidad, donde a medida que se van mejorando los EF se revisan los términos afectados, ya descritos en ciclos anteriores, para identificar aquellos que han cambiado y si aparecen nuevos símbolos en los EF, estos se clasifican.

- *Describir Términos del Léxico del Sistema*

Este bloque, siempre utilizado para la creación del LEL del Sistema, comprende completar o actualizar las definiciones de los símbolos identificados en la actividad anterior. Si el símbolo se corresponde con uno del LEL del UdeD se revisa su definición original, si no hay cambios se transcribe tal cual, si es necesario algún cambio se realizan los cambios. Todo símbolo nuevo se completa con su definición correspondiente según el tipo asignado. Se debe prestar atención al factor *Reuso de artefactos de requisitos existentes* (F15=LEL-S). En caso de disponer de un LEL del Sistema creado para un proyecto previo, se pueden reusar sus definiciones, pudiendo realizarse las actualizaciones que

se consideren pertinentes. En caso de un término proveniente del LEL del UdeD, se usa primeramente esta definición y se revisa en función del símbolo que podría reusarse. Si se trata de una mejora a cierta funcionalidad, bajo un modelo incremental (C8=V, ver condición en la subsección 4.3.1), durante la descripción se mejorarán las definiciones de aquellos símbolos existentes, identificados en la actividad anterior.

- *Verificar Léxico del Sistema*

Este bloque de proceso independiente involucra, por una parte, comprobar la consistencia interna, y además verificar que todos los términos del léxico existan en el conjunto de EF. En especial, se debe prestar atención a la *Fiabilidad necesaria del software* (F14). Si hay poca exigencia se puede aplicar técnicas menos estructuradas como la lectura ad-hoc. En cambio, si la exigencia es mayor, con valores “Media”, “Alta” o “Muy Alta”, la verificación se debe llevar adelante de un modo más exhaustivo, por ejemplo, aplicando la técnica de inspección de LEL, o incluso aumentando la cantidad de participantes para reforzar la actividad. Así también se debe tener en cuenta en este bloque la *Experiencia en el proceso de requisitos* (F24) para seleccionar técnicas de verificación acordes al nivel de experiencia del equipo.

4.3.7 Variantes al Especificar Requisitos

El Punto de Variación 5 se relaciona con el subproceso *Especificar Requisitos* que se muestra en la Figura 4-14. Como se puede observar, la adaptación principal de este subproceso tiene que ver con la omisión de los bloques *Organizar Requisitos* y *Verificar Documento de Requisitos* reemplazándolos por el *Bloque Nulo*. Estos bloques pueden omitirse cuando no existe alguna exigencia contractual para la confección de un documento formal de requisitos según algún formato de ERS, lo cual se debe evaluar con la siguiente condición:

Condición C16:

Exigencia de producir el documento ERS (F20) = {*No*}

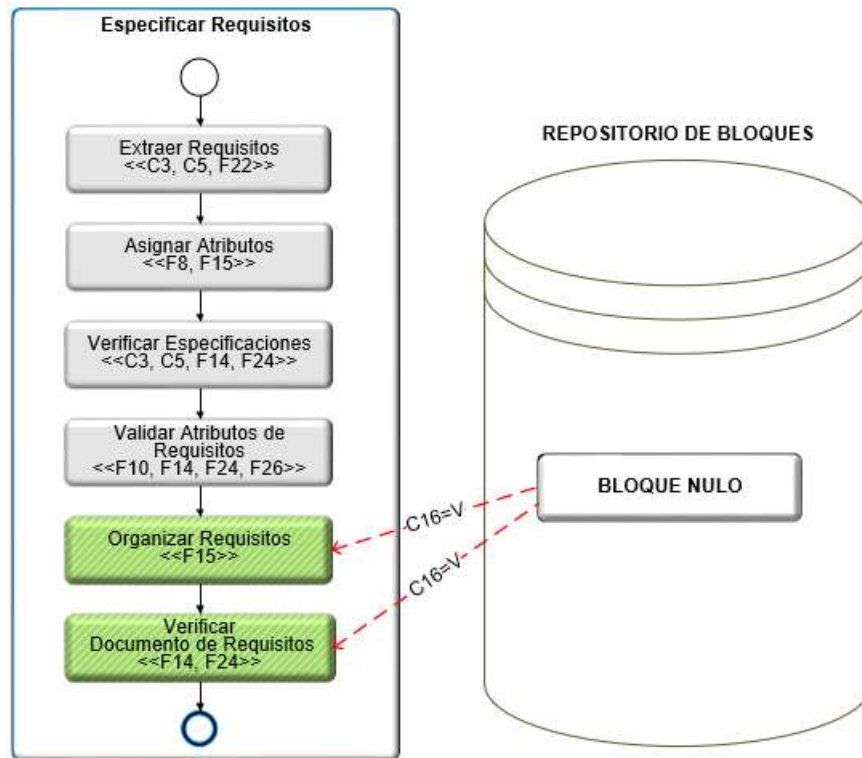


Figura 4-14. Variantes al Especificar Requisitos

A continuación, se describen los bloques del subproceso indicando la manera en que los factores situacionales influyen en cada uno de estos:

- *Extraer Requisitos*

Se trata de un bloque de proceso independiente en el cual se extraen los requisitos del conjunto total de EF en caso de utilizarse el modelo de proceso “Cascada”, “Transformación Formal” o “Iterativo”, o del conjunto de EF afectados por el incremento en caso del modelo “Incremental” (F22). Las condiciones C3 y C5 (ver en subsección 4.3.1), que determinan la existencia del LEL del UdeD y del LEL del Sistema respectivamente, son parámetros para considerar en este bloque. Si se ha creado el LEL del Sistema (C5=F) los requisitos extraídos de los EF utilizan los términos de dicho léxico con sus vínculos correspondientes. Caso contrario, si sólo se dispone del LEL del UdeD (C3=F y C5=V), los requisitos utilizan los símbolos de este. Puede darse la situación en que ambas condiciones sean verdaderas, cuando se ha omitido la creación de ambos léxicos, en tal caso los requisitos no tienen vínculos asociados. En este último caso si se sigue un modelo de proceso con ciclos, puede ocurrir que se disponga de un modelo léxico proveniente de ciclos previos, en cuyo caso los

requisitos que se extraen se describen incorporando los vínculos correspondientes a dicho modelo.

- *Asignar Atributos*

Este bloque de proceso, siempre utilizado al especificar los requisitos, implica completar las definiciones de los requisitos y asignarles sus atributos. Inicialmente se considera el factor *Reuso de artefactos de requisitos existentes* (F15=Requisitos). Si hay posibilidad de reusar requisitos creados anteriormente para otro proyecto, se los identifica para permitir precisar la descripción de los extraídos en la actividad anterior y colaborar en completar sus atributos. En este bloque se considera también el factor *Conflicto de intereses de usuarios* (F8), si su valoración es “Alto” o “Muy Alto”, pueden surgir algunas contradicciones con respecto a la asignación de prioridades a los requisitos, se requerirá más negociación entre los usuarios.

- *Verificar Especificaciones*

La actividad de este bloque de proceso independiente comprende comprobar que cada requisito sea sintácticamente correcto, que todos se hayan extraído desde los EF, que la asignación de atributos sea la que corresponde y que se use correctamente las referencias al léxico correspondiente (según condiciones C3, C5 o el léxico proveniente de ciclos previos). Se debe prestar atención a la *Fiabilidad necesaria del software* (F14) a fin de seleccionar apropiadamente las técnicas de verificación a aplicar. Si no existe demasiada exigencia tal vez sea suficiente realizar revisiones informales internas. En contraposición, si hay alta exigencia de fiabilidad, con valores “Alta” o “Muy Alta”, el proceso se debe realizar de modo más exhaustivo, por ejemplo, con inspecciones con una lista de control, o incluso es posible reforzar la actividad aumentando la cantidad de participantes en el proceso. En adición a lo anterior, para la selección de las técnicas de verificación, se debe considerar el nivel de *Experiencia en el proceso de requisitos* (F24) a fin de seleccionar las técnicas más convenientes conforme a la experiencia que tiene el equipo de ingenieros de requisitos.

- *Validar Atributos de Requisitos*

La finalidad de la actividad de este bloque de proceso independiente es confirmar o rectificar los atributos asignados a los requisitos por parte de los

involucrados. Un parámetro a tener en cuenta es el nivel de *Fiabilidad necesaria del software* (F14). Si no hay mucha exigencia, la validación puede realizarse informalmente, por ejemplo, mediante entrevistas. En cambio, ante mayor exigencia, con valores “Alta” o “Muy Alta”, el proceso debe realizarse de un modo más exhaustivo, por ejemplo, con reuniones formales con los clientes y usuarios, donde se lean detalladamente las especificaciones enfocándose en controlar los valores asignados a cada atributo. Asimismo, si la *Localización de usuarios* es distribuida (F10) o hay *Diferencias temporales entre los involucrados* (F26) con valores “Media”, “Alta” o “Muy Alta” deberá establecerse la conveniencia de otras técnicas de validación que puedan implementarse individualmente con los distintos usuarios o que no requieran comunicación sincrónica entre los involucrados. Además, la selección de las técnicas de validación se realiza en función a la *Experiencia en el proceso de requisitos* (F24) del equipo, ya que determinadas técnicas pueden requerir mayor o menor nivel de experiencia para su aplicación.

- *Organizar Requisitos*

En este bloque de proceso se organizan los requisitos de modo tal de obtener el máximo de legibilidad, utilizando algún formato o estándar adoptado para el documento. El parámetro para considerar es el *Reuso de artefactos de requisitos existentes* (F15=Requisitos). Si hay algún documento de ERS para reutilizar se define qué partes del documento existente pueden ser reutilizadas.

- *Verificar Documento de Requisitos*

Se trata de un segundo bloque de verificación en el subproceso que tiene como objetivo asegurar la correctitud del documento de ERS con respecto a la inclusión de cada requisito identificado y el cumplimiento del estándar adoptado. La selección de las técnicas a aplicar tiene las mismas consideraciones que para el bloque *Verificar Especificaciones* detallado antes según la *Fiabilidad necesaria del software* (F14) y la *Experiencia en el proceso de requisitos* (F24).

Capítulo 5

Adaptación Dinámica del Proceso de Requisitos

5.1 Estimación y Evolución de los Factores Situacionales

Como se ha destacado a lo largo de este trabajo, cada proyecto de software tiene sus características particulares, algunas de las cuales pueden ser de importancia al momento de llevar adelante el proceso de IR. Sería conveniente que dichas características se puedan identificar desde un principio a fin de planificar la adaptación adecuada del proceso. Sin embargo, las condiciones iniciales del contexto y del proyecto pueden evolucionar a lo largo del ciclo de desarrollo del software. Por tal motivo, a veces no alcanza con solo adaptar el proceso antes de su puesta en ejecución, sino que este puede requerir readaptaciones durante su desarrollo, en posteriores hitos estratégicos.

Tal como se expone en el capítulo 4, los mecanismos de adaptación situacional del proceso dependen fuertemente de los factores situacionales, pues su adaptación es función directa de los valores que se le asignan a dichos factores. Esa asignación es en realidad una estimación que debe ser lo más aproximada posible a la situación real imperante para el proyecto.

Básicamente, en la adaptación del proceso, la eliminación, reemplazo o adición innecesaria de una actividad puede tener una importancia diferente, sin embargo, es claro que, en condiciones similares, la primera de ellas es mucho más perjudicial que las otras dos.

Al inicio del proceso de requisitos no es posible conocer con precisión los valores de algunos factores dado que no resulta sencillo percibir las particularidades que cada proyecto presenta. Aun cuando no sea este el caso, algunos factores pueden requerir la modificación de sus valores iniciales debido a cambios en el contexto. Por lo tanto, la adaptación del proceso puede verse comprometida cuando los factores situacionales evolucionan a lo largo del proceso y ello no es percibido oportunamente, o cuando los factores no son estimados con precisión desde el inicio del proceso. Es así que, desde la dimensión de la evolución, como se puede apreciar en la Tabla 5-1, los factores pueden clasificarse en dos tipos:

- *Invariante*, aquel factor que, por su propia naturaleza, no sufre cambios durante el proceso de IR.
- *Contingente*, aquel factor que puede evolucionar naturalmente debido a cambios en el contexto de aplicación y/o en el contexto del proyecto.

Tabla 5-1. Tipo de evolución de los factores situacionales

#	Factores Situacionales	Tipo de Evolución
F1	Complejidad del contexto	- Invariante
F2	Tipo de cliente	- Invariante
F3	Novedad del contexto	- Invariante
F4	Reingeniería en el proceso de negocio	- Contingente
F5	Volatilidad del contexto	- Invariante
F6	Volatilidad de los requerimientos del cliente	- Contingente
F7	Inconsistencias en el contexto	- Invariante
F8	Conflicto de intereses de usuarios	- Contingente
F9	Rotación de usuarios	- Invariante
F10	Localización de usuarios	- Invariante
F11	Conocimiento previo del dominio	- Invariante
F12	Envergadura del proyecto	- Invariante
F13	Rotación del equipo desarrollador	- Invariante
F14	Fiabilidad necesaria del software	- Invariante
F15	Reuso de artefactos de requisitos existentes	- Invariante
F16	Creación de artefactos de requisitos para reuso	- Contingente
F17	Pre-Rastreabilidad de los requisitos	- Invariante
F18	Post-Rastreabilidad de los requisitos	- Invariante
F19	Granularidad de rastreabilidad de los requisitos	- Invariante
F20	Exigencia de producir el documento ERS	- Invariante
F21	Restricciones de tiempo y recursos para el proyecto	- Contingente
F22	Modelo de proceso de software	- Invariante
F23	Tipo de incremento	- Contingente
F24	Experiencia en el proceso de requisitos	- Invariante
F25	Diferencias idiomáticas y/o culturales	- Invariante
F26	Diferencias temporales entre los involucrados	- Invariante

Otro aspecto importante a considerar es el nivel de confianza con el que los ingenieros de requisitos estiman cada factor situacional. Así desde esta perspectiva, el valor de cada factor se tipifica como:

- *Seguro*, cuando es posible establecer con precisión el valor del factor. Por ende, se asume que este no cambiará durante el proceso de IR.
- *Razonablemente seguro*, cuando se considera que es baja la probabilidad de haberse cometido un error en su estimación.
- *Dudoso*, cuando no es posible establecer el valor del factor con precisión, sino recién cuando se logra una mejor comprensión de la situación.

A modo particular, los factores *Tipo de cliente* (F2), *Novedad del Contexto* (F3) y *Modelo de proceso de software* (F22) se determinan al inicio del proyecto de software, pues son informados por el líder del proyecto y no sería esperable que cambiaran. Es por ello que estos factores no necesitan ser reevaluados, debiéndose ser su nivel de confianza seguro.

El formulario de captura de datos sobre la situación del proyecto de software (ver Anexo B), permite establecer los valores de cada factor y el nivel de confianza asignado a dicho valor.

5.2 Estudio de la Evolución de los Factores Situacionales

Se realizó un estudio de caso para analizar la percepción de la situación imperante en el proceso de IR por parte de los ingenieros de requisitos, a través de la valoración de los factores que caracterizan dicha situación y la verosimilitud o estabilidad de dichas valoraciones a lo largo del proceso.

Son muy difundidos los peligros del estudio de caso como método para la comprensión de un problema (Kyburz-Graber, 2004) en virtud de sus características de ser intuitivo, primitivo y difícil de gestionar. Aunque parte de estos peligros no son tales, sino que provienen de la usual confusión entre los tipos de evidencias manipuladas (datos cualitativos o cuantitativos), los mecanismos de recolección de información y la estrategia de investigación (estudio de caso). Claramente, el estudio de caso puede combinarse con cualquier forma de evidencia y con cualquier mecanismo de recolección de información (Yin, 1981). La diferencia entre un estudio de un caso y un experimento es que en el primero se examina un fenómeno en su contexto real, especialmente cuando dicho fenómeno no puede separarse fácilmente de su contexto o cuando esta interrelación es de interés, mientras que en el segundo se separa deliberadamente el fenómeno de su contexto.

En particular, la presencia combinada de este fenómeno y su contexto genera la presencia de tantas variables que hace inviable el diseño de experimentos clásicos. Es por ello que en el presente trabajo se llevó a cabo el estudio de caso. La cantidad de factores situacionales, 21, y la cantidad de clases cualitativas posibles (ver Tabla 3-5) hacen que la cantidad de combinaciones, 1680, sea imposible de

cubrir, aun con un ejemplar por combinación. Además, como el proceso de requisitos fue aplicado estrictamente en organizaciones reales, no resulta siquiera posible tener un ejemplar para cada posible valor de cada uno de los factores situacionales.

El fenómeno bajo estudio consistió entonces en un conjunto de factores situacionales y la calidad con que se los estimaba en el transcurso de un proceso de requisitos. La captura de datos para el estudio de caso fue realizada entre 2015 y 2017, tomando como base 21 factores situacionales. Posteriormente, al avanzar con la tesis se detectó la necesidad de incorporar 5 factores que no habían sido considerados en un comienzo. En el estudio de caso se evaluó este fenómeno en 35 proyectos de requisitos en organizaciones reales en su contexto natural, con el propósito de describir y en lo posible explicar esas calidades y evoluciones, para poder luego refinar el proceso de adaptación dinámica del proceso de IR. Se recolectaron las valoraciones asignadas a cada factor y el nivel de confianza asignado, al iniciar y al finalizar cada proyecto. Estos datos se analizaron transversalmente para cada factor, logrando una razonable confiabilidad (ver un resumen de resultados en Anexo C). El detalle del trabajo fue publicado en (Ledesma, Hadad, Doorn & Bedetti, 2018).

Las siguientes son algunas características generales de los 35 proyectos en estudio:

- La *Complejidad del proyecto* era media en 66% de los casos, y baja o muy baja en los 34% restantes.
- La *Envergadura del proyecto* era mediana en 49% de los casos, y en un 49% pequeña.
- En un 76% de los casos el *Conocimiento previo del dominio* era bajo o muy bajo.
- La *Volatilidad del contexto* en un 82% de los proyectos era baja o muy baja.
- El grado de *Inconsistencias en el contexto* del 76% de los casos fue bajo o muy bajo.
- El *Conflicto de intereses* era bajo o muy bajo en el 89% de los casos.
- La *Volatilidad de los requerimientos* del cliente era baja o muy baja en el 76% de los proyectos.

Por otro lado, se realizó una evaluación del nivel de confianza con el que se calificó a los valores asignados a cada factor. La Figura 5-1 presenta a modo muy general el porcentaje promedio de valoraciones seguras, razonablemente seguras y dudosas obtenidas al inicio y al final de los 35 proyectos en su conjunto.

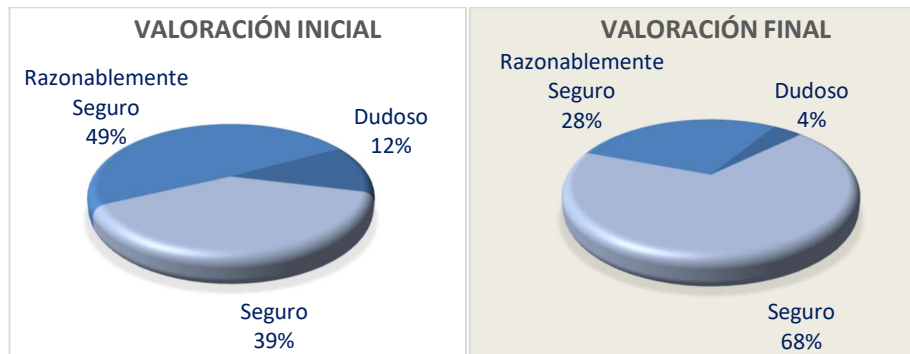


Figura 5-1. Nivel de confianza en la valoración inicial y final

A simple vista resulta evidente que en la medida que avanza el proyecto el nivel de confianza de los ingenieros de requisitos para valorar los factores aumenta, donde obviamente al finalizar el proyecto todos los factores debieran haber sido evaluados con certeza, sin embargo, esto no es lo que muestra la Figura 5-1. Esto está evidenciando una cierta inhabilidad de los ingenieros de requisitos en la forma de realizar la valoración asignada, lo que debiera ser compensado con un refinamiento de la heurística que guíe más eficazmente la estimación de los factores.

A fin de obtener un análisis más preciso se estudió en detalle la evolución de la valoración de cada factor para cada proyecto en particular, considerando el nivel de confianza dada por el equipo de requisitos. En la Figura 5-2 se puede observar a modo de ejemplo el examen de la evolución del nivel de confianza para el factor *Complejidad de contexto*. Los resultados obtenidos se discriminaron en función de los casos que:

- mantuvieron su calificación de confianza inicial en seguro o razonablemente seguro,
- mejoraron su valoración de razonablemente seguro a seguro y de dudoso a razonablemente seguro o a seguro,
- bajaron su nivel de confianza de seguro a razonablemente seguro, o
- finalizaron con nivel de confianza dudoso.

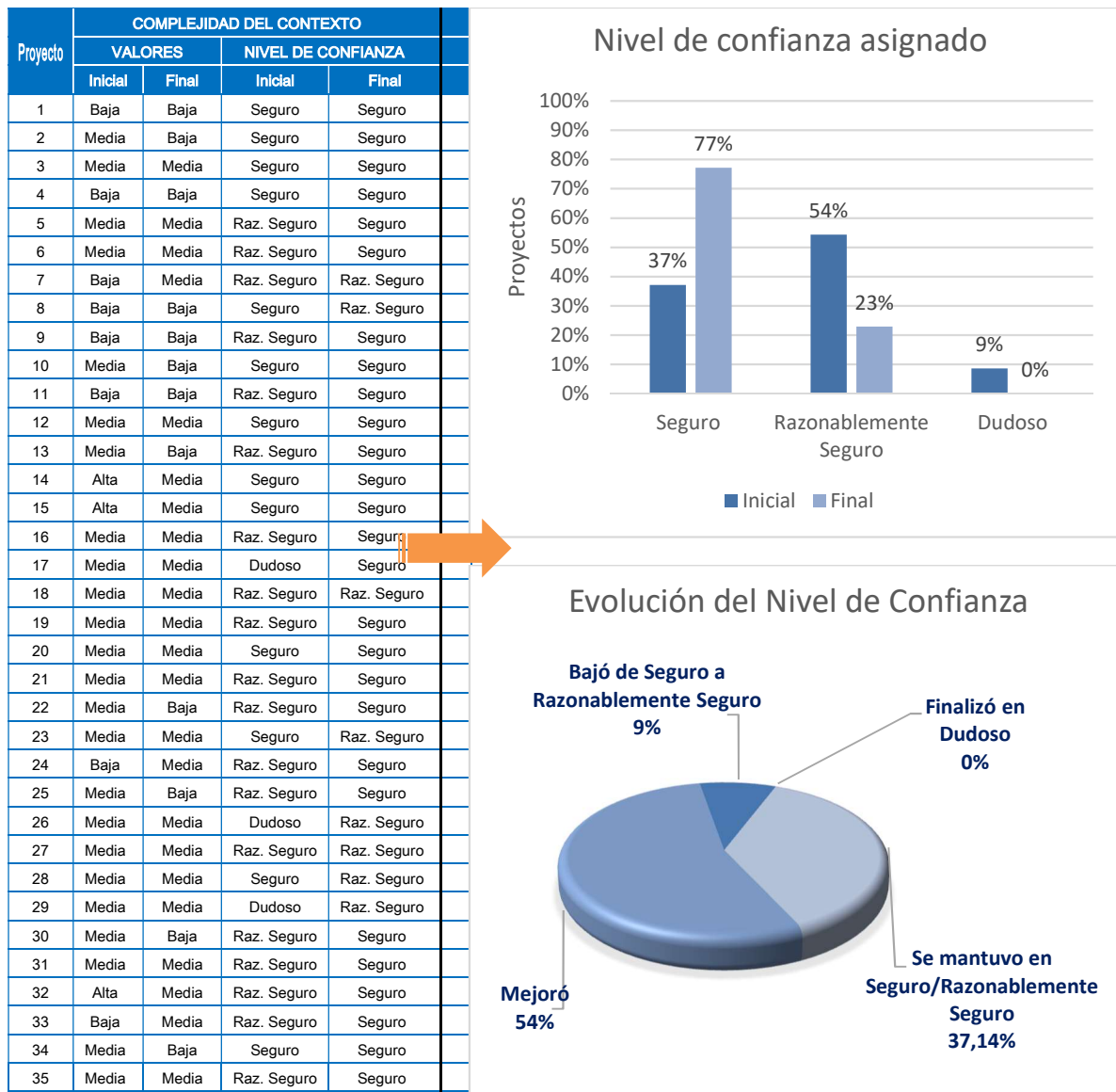


Figura 5-2. Análisis de la evolución del nivel de confianza para el factor situacional Complejidad del contexto

Algo muy llamativo detectado durante este análisis es que 13 factores cambiaron su nivel de confianza en más del 50% de los proyectos. Además, vale mencionar las siguientes observaciones:

- Factores que se destacaron por mantener el nivel de confianza en seguro o razonablemente seguro:
 - *Tipo de cliente* → en 74% de los proyectos
 - *Novedad del negocio* → en 74% de los proyectos
- Factores que mejoraron su nivel de confianza en más de la mitad de los proyectos:
 - *Complejidad del contexto* → en 54% de los proyectos
 - *Creación de artefactos para reuso* → en 50% de los proyectos

- *Pre-Rastreabilidad de los requisitos* → en 50% de los proyectos
- Factores cuya valoración final fue dudosa, correspondientes a una escasa cantidad de proyectos que en su mayoría tenían números altos en la cantidad de cambios en el nivel de confianza:
 - *Inconsistencias en el contexto* → en 10% de los proyectos
 - *Envergadura del proyecto* → en 9% de los proyectos
 - *Post-Rastreabilidad de los requisitos* → en 12% de los proyectos
 - *Rotación del equipo desarrollador* → en 9% de los proyectos

En estos cuatro factores aún al finalizar el proyecto de IR se tuvo dificultad en su valoración, lo cual no debiera ocurrir. El no poder establecer con certeza la *Envergadura del proyecto* al finalizar el mismo, está indicando la inhabilidad del equipo de requisitos para realizar esta tarea, lo que implica la necesidad de mejorar la heurística que guía cómo establecer con precisión el tamaño del proyecto. De la misma manera, debe mejorarse la heurística para identificar el nivel de *Inconsistencias en el contexto*. En cuanto al factor *Post-Rastreabilidad de requisitos*, no hubo pautas precisas en los proyectos sobre este tema. Respecto a la *Rotación del equipo desarrollador*, puede considerarse imprecisa la definición de cómo valorar efectivamente este factor, lo cual obliga a explicitar el significado de cada posible valor.

Todo lo anterior debe ser evaluado en el entendimiento que las heurísticas que se proveen en un proceso de requisitos no deben ser vistas como una mera ayuda memoria o una lista de control sino como un soporte con la fortaleza suficiente para lograr que el desempeño de un novicio se aproxime al desempeño de un experto en la disciplina. No lograr este efecto es una indicación terminante de la debilidad de la correspondiente heurística.

Por otra parte, se han identificado factores, que independientemente de su tipo de evolución, sufrieron cambios relevantes en su estimación entre el valor inicial y el final en más del 30% de los proyectos. Los que tuvieron mayor cantidad de cambios en la valoración fueron los siguientes:

- *Reingeniería en el proceso de negocio* → en 54% de los proyectos
- *Calidad exigida del software* → en 53% de los proyectos
- *Envergadura del proyecto* → en 46% de los proyectos
- *Pre-Rastreabilidad de los requisitos* → en 44% de los proyectos

En cuanto a estos últimos cuatro factores mencionados, lo notable es que la mayoría de los valores fueron asignados inicialmente con un nivel de confianza razonablemente seguro. Es evidente la existencia de cierta dificultad inicial para asignarles un valor con precisión, por consiguiente, son factores a los que hay que prestar especial atención al momento de valorarlos.

Así también, algunos factores cuyas opciones de valoración son muy precisas y no se espera que cambien a lo largo del proyecto tales como: *Pre-Rastreabilidad de los requisitos* o *Modelo de proceso de software*, han cambiado sus valores. Es posible que estas inconsistencias se deban a que el significado del factor no haya resultado lo suficientemente claro, fuese ambiguo o no fuera clara la manera de lograr un conocimiento apropiado sobre el mismo. Es por ello que se han tomado medidas para refinar la heurística de valoración de dichos factores a fin de que el equipo de requisitos comprenda claramente lo que involucra cada uno de ellos y así mejorar su valoración.

Del análisis de los resultados del estudio de caso presentado, queda en evidencia la dificultad en establecer valores confiables al inicio del proceso de IR. Este estudio ha permitido evidenciar la necesidad de disponer de un proceso de requisitos que se ajuste dinámicamente durante su desarrollo. Gran parte de estas adaptaciones a lo largo del proceso se deben principalmente a cambios por apreciaciones imprecisas de la situación. Se ha podido detectar que las valoraciones inciertas o erróneas de estos factores se deben en gran parte a la dificultad de identificar claramente al inicio la situación del contexto y del proyecto y, en menor medida, al desconocimiento de algunas actividades de la IR o la dificultad de interpretación de las definiciones de algunos factores. Respecto a la primera causa, se ha definido una heurística que ayude a una valoración temprana menos dudosa y/o que mida las consecuencias de una adaptación imprecisa y permita redefinir transitoriamente esos valores iniciales para una adaptación del proceso que mitigue futuros cambios (ver siguiente sección). Para una mejor comprensión del significado de los factores situacionales, se ha mejorado la definición de estos en el formulario de registro de datos (ver Anexo B), tratando de reducir ambigüedades y ofreciendo precisiones sobre sus posibles valores. Debe tenerse en cuenta que, si bien las evoluciones a lo largo del proyecto de algunos factores pueden ser impredecibles, las apreciaciones iniciales dudosas o

incorrectas son potencialmente reducibles. Es entonces relevante la calidad con que es posible realizar la estimación de los factores situacionales, y en consecuencia la calidad de la adaptación del proceso de requisitos a su contexto.

En la siguiente sección se describe un proceso para la construcción del propio proceso de requisitos, que permita su adaptación en forma dinámica, tomando en consideración la evolución de los factores situacionales a medida que avanza el proceso mismo de IR.

5.3 Proceso de Adaptación Dinámica

La adaptación del proceso base de IR, representado en la Figura 5-3 implica la asignación de valores a cada factor situacional, estableciendo el nivel de confianza con que se estiman dichos valores. Por otra parte, se debe disponer de un repositorio que contenga la especificación detallada de cada bloque de proceso que conforma el proceso base de IR, además de todos los bloques de proceso que ofrecen flujos variantes. Los puntos de variación y la combinación de los factores situacionales son parte del mecanismo para intercambiar, adicionar u omitir bloques, o realizar ciertos cambios internos en los bloques en función a los parámetros de entrada.

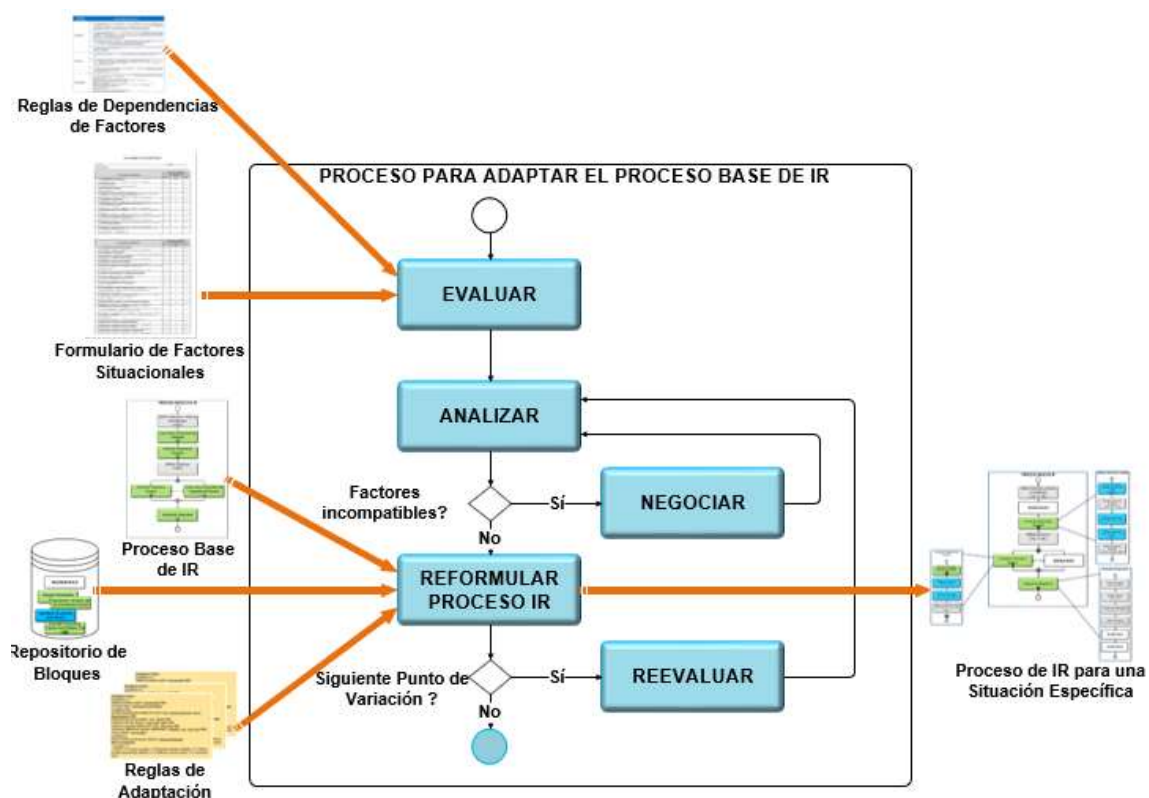


Figura 5-3. Proceso de adaptación del proceso de IR en función de factores situacionales

A continuación, se detallan las actividades involucradas en el proceso de adaptación:

i) Evaluar los factores situacionales

En primer lugar, es necesario evaluar los factores desde la perspectiva del contexto y del proyecto, asignando un valor a cada factor y un nivel de confianza para dicho valor establecido. Se debe prestar atención a las posibles dependencias entre los factores para asignar cuidadosamente los valores, ya que algunos factores pueden anular la evaluación de otros o restringir sus valores admisibles (ver reglas de dependencia limitantes e invalidantes en la Tabla 3-6). También, corresponde dar especial atención a aquellos factores que, en base al análisis de la evolución descrito en la sección anterior, son propensos a tener cambios relevantes. La información se registra en un formulario de factores situacionales (ver Anexo B), el cual brinda una guía sobre los valores admitidos para los factores, como así también, permite identificar dependencias de tipo invalidantes y limitantes entre factores en función de los valores asignados a otros factores. Adicionalmente, el formulario contiene una breve reseña de la definición de cada factor y algunas pautas destinadas a orientar la valoración del factor.

ii) Analizar los factores situacionales

Se analizan los valores asignados a los factores y su nivel de confianza. Por un lado, se verifica el cumplimiento de las reglas de dependencia (ver Tabla 3-6), prestando especial atención a aquellas dependencias incompatibles entre factores, las que requerirán un tratamiento particular. Por otro lado, se analizan aquellos factores con bajo nivel de confianza, es decir, calificados con dudoso. Frente a estos factores, se debe adoptar una posición a fin de componer luego el proceso de IR, hasta que dichos factores se puedan valorar con mayor precisión. Bajo una posición optimista, se asignan a estos factores un valor Muy Bajo (o Sí o el que corresponde al valor extremo del factor en cuestión), sin tener en cuenta el valor asignado originalmente. Bajo una posición conservadora, los factores en cuestión toman un valor Muy Alto (o No o el que corresponde al valor extremo del factor en cuestión). La reasignación de los valores depende del significado del factor en sí mismo, tomando un valor extremo bajo una posición

conservadora y el valor extremo opuesto bajo una posición optimista. Una posición equilibrada implica el análisis de cada factor individual sobre la base de su valor inicial. Esta reasignación de valores es sólo con el propósito de ensamblar el proceso, ya que los valores originales persisten. Sería también prudente tomar alguna medida respecto a aquellos factores cuyo valor fue ponderado como razonablemente seguro, optando por modificar transitoriamente su valor a un valor inmediato superior en la escala de valoración (o inmediatamente inferior según corresponda con el factor y la postura optimista o conservadora adoptada), para compensar equilibradamente esa incertidumbre de menor grado.

iii) Negociar incompatibilidades

Esta actividad se realiza cuando se detectan dependencias de tipo incompatibles entre dos o más factores (ver en la Tabla 3-6). El objetivo de la negociación es llegar a un acuerdo entre las partes interesadas, tanto por parte de los clientes como del equipo desarrollador, para resolver los conflictos que dan origen a la incompatibilidad. Luego de lo cual se reasignan valores a uno o varios de estos factores intervinientes. Después de la negociación, se debe hacer un análisis parcial de los factores involucrados, retornando a la actividad previa.

iv) Reformular el proceso de IR

Esta actividad consiste en armar el proceso de IR o rearmarlo si hubo nuevas evaluaciones de los factores. Esta actividad se realiza a partir del proceso base de IR, siguiendo el conjunto de reglas de adaptación para aplicar operaciones sobre cada bloque de proceso en cada punto de variación no alcanzado aún. Dichas reglas han sido descritas en el capítulo 4, las mismas involucran “condiciones” que combinan de modo específico aquellos factores situacionales afectados en cada punto de variación, y “parámetros” asociados a los bloques de proceso. El armado o rearmado consiste entonces en mantener, eliminar, sustituir o agregar bloques de proceso al proceso base, y asignar valores a parámetros de bloques de proceso de acuerdo a su especificación. Estos bloques se obtienen de un repositorio que contiene todos los bloques de proceso.

v) Reevaluar los factores situacionales

Al finalizar cada etapa del proceso de IR, en el siguiente punto de variación, se reevalúan los factores. Se debe poner especial atención en aquellos factores con bajo nivel de confianza y, si es necesario, se les reasigna un nuevo valor. En general, el nivel de confianza aumenta debido a una mejor comprensión de la situación mientras se avanza en el proceso de requisitos. Asimismo, se debe establecer si se produjo algún cambio en la situación del proyecto, para ello se revisan aquellos factores que se consideran contingentes o propensos a evolucionar (según lo presentado en la sección previa). Si el valor de algún factor se modifica o su nivel de confianza, se debe actualizar el formulario de factores situacionales, realizándose nuevamente la actividad de análisis.

En resumen, el proceso de IR adaptado se ejecuta, y en hitos predefinidos (puntos de variación) este proceso se revisa y, de ser necesario, se reformula, como se muestra en el segundo ciclo en la Figura 5-3.

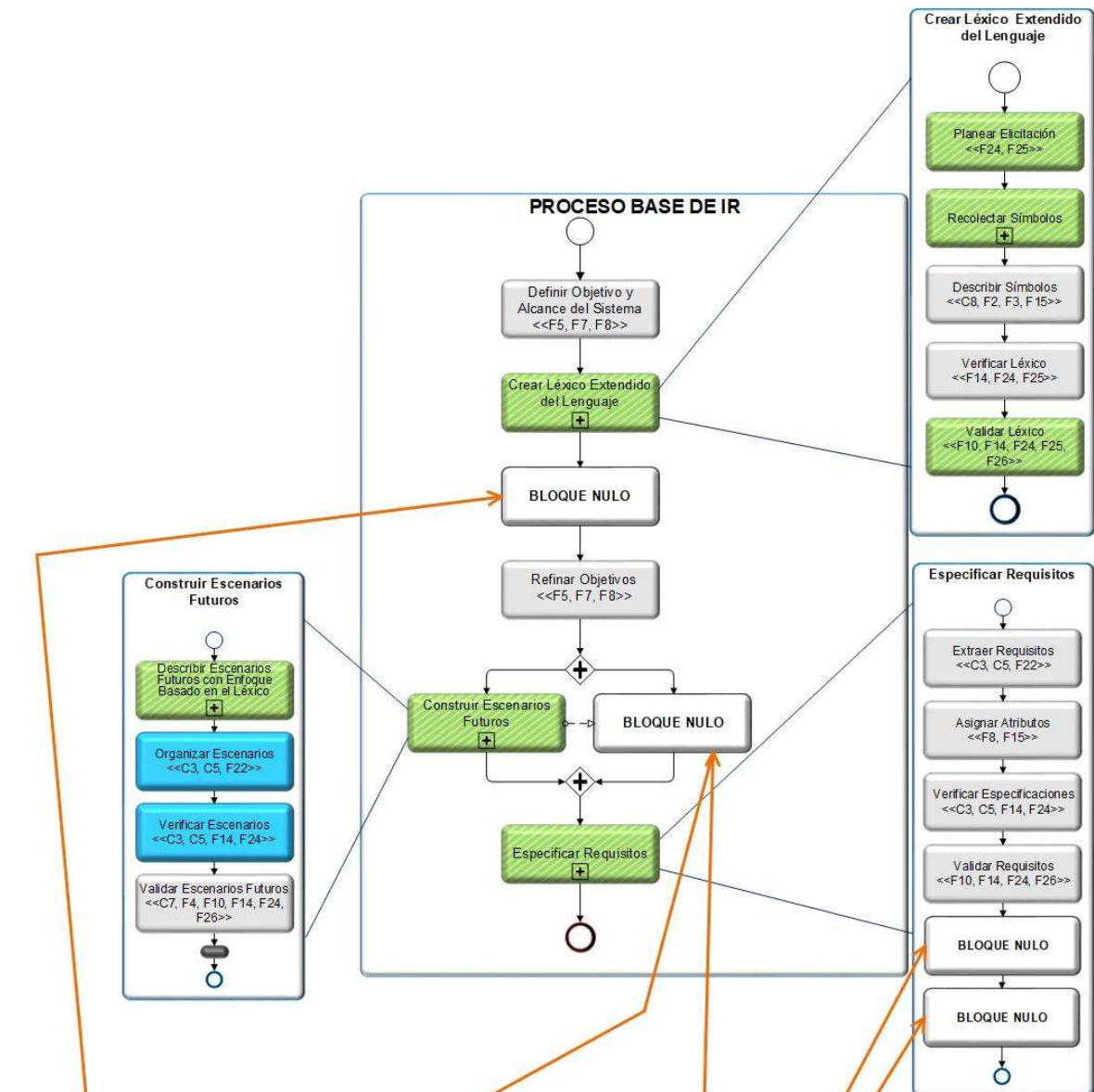
La Figura 5-4 muestra un ejemplo de la construcción del proceso para una situación específica, la cual se representa por la asignación de valores a los factores situacionales al inicio del proyecto. Las principales adaptaciones del *Proceso Base de IR* que se pueden observar son:

- El bloque *Crear Léxico Extendido del Lenguaje* se preserva en el proceso debido a que el factor *Creación de artefactos de requisitos para reuso* tiene valor “LEL”, lo que obliga a que se deba construir dicho modelo (en términos de las reglas de adaptación la condición C3 es falsa).
- El bloque de proceso *Construir Escenarios Actuales* se omite dado que el factor *Volatilidad del contexto* tiene valoración “Muy Alta”, con lo cual no se justifica construir este modelo (la condición C4 es verdadera).
- En el subproceso *Construir Escenarios Futuros* se reemplaza el bloque de proceso *Describir Escenarios Futuros con Enfoque Procedural* por el bloque *Describir Escenarios Futuros con Enfoque Basado en el Léxico*, ya que no se dispone de EA pero sí del LEL del UdeD, y además el factor *Reingeniería en el proceso de negocio* tiene valor “Media” (la condición C13 es verdadera).
- El bloque de proceso *Crear Léxico Extendido del Sistema* se omite pues no hay exigencia de construir tal artefacto para reuso y, el equipo tiene *Conocimiento*

del dominio "Alto", prácticamente no hay diferencias idiomáticas ni temporales entre los involucrados, los usuarios están colocalizados y además la *Reingeniería en el proceso de negocio* tiene valor "Media" (la condición C5 es verdadera).

- El bloque de proceso *Especificar Requisitos* se mantiene como parte del proceso, dado que el factor *Envergadura del proyecto* tiene valor "Mediana" (la condición C6 es falsa). Los bloques internos *Organizar Requisitos* y *Verificar Documento de Requisitos* se reemplazan por el *Bloque Nulo* debido a que el factor *Exigencia de producir el documento ERS* tiene valor "No" (la condición C16 es verdadera).

Aparte de lo anterior, se deben considerar los parámetros que reciben los distintos bloques. Por ejemplo, el bloque de proceso *Organizar Escenarios* del subproceso *Construir Escenarios Futuros* recibe tres parámetros, entre ellos el factor *Modelo de proceso de software* (F22) que en este ejemplo tiene valor "Incremental", lo que implica aplicar las operaciones y realizar la integración de los escenarios afectados por el incremento. Además, los EF Integradores deberán contener los vínculos a los términos del LEL del UdeD (parámetro C3 con valor falso) debido a que, según las adaptaciones del proceso detalladas antes, este es el léxico disponible (parámetro C5 con valor verdadero).



FACTORES DEL CONTEXTO		NIVEL DE CONFIANZA		
		Seguro	Raz. Seguro	Dudoso
1.	Complejidad del contexto Muy Alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2.	Tipo de Cliente Mercado <input type="checkbox"/> Potencial <input type="checkbox"/> Cliente Específico <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	Novedad del contexto Si <input type="checkbox"/> No <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	Reingeniería en el proceso de negocio Muy Alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
5.	Volatilidad del contexto Muy Alta <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	Volatilidad de los requerimientos del cliente Muy Alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input checked="" type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7.	Inconsistencias en el contexto Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input checked="" type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8.	Conflicto de intereses de usuarios Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input checked="" type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
9.	Rotación de usuarios Muy Alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input checked="" type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
10.	Localización de usuarios Distribuidos <input type="checkbox"/> Colocalizados <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

FACTORES DEL PROYECTO		NIVEL DE CONFIANZA		
		Seguro	Raz. Seguro	Dudoso
11.	Conocimiento previo del dominio Muy Alto <input type="checkbox"/> Alto <input checked="" type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12.	Envergadura del proyecto Muy Grande <input type="checkbox"/> Grande <input type="checkbox"/> Mediana <input checked="" type="checkbox"/> Pequeña <input type="checkbox"/> Muy Pequeña <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
13.	Rotación del equipo desarrollado Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input checked="" type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14.	Fiabilidad necesaria del software Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input checked="" type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15.	Reuso de artefactos de requisitos existentes LEL <input type="checkbox"/> EA <input type="checkbox"/> EF <input type="checkbox"/> LEL-S <input type="checkbox"/> Requisitos <input checked="" type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16.	Creación de artefactos de requisitos para reuso LEL <input checked="" type="checkbox"/> EA <input type="checkbox"/> LEL-S <input type="checkbox"/> Requisitos <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17.	Pre-Rastreabilidad de los requisitos Total <input type="checkbox"/> Parcial <input checked="" type="checkbox"/> No <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18.	Post-Rastreabilidad de los requisitos Si <input type="checkbox"/> No <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19.	Granularidad de Rastreabilidad de los requisitos Individual <input type="checkbox"/> Grupal <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20.	Exigencia de producir el documento ERS Si <input type="checkbox"/> No <input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
21.	Restricciones de tiempo y recursos para el proyecto Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input checked="" type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
22.	Modelo de proceso de software Cascada <input type="checkbox"/> Incremental <input checked="" type="checkbox"/> Iterativo <input type="checkbox"/> Transf. Formal <input type="checkbox"/> Otro <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
23.	Tipo de incremento Nueva Funcionalidad <input checked="" type="checkbox"/> Mejora a Funcionalidad <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
24.	Experiencia en el proceso de requisitos Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input checked="" type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
25.	Diferencias idiomáticas y/o culturales Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input checked="" type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
26.	Diferencias temporales entre los involucrados Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input checked="" type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Formulario de Evaluación de los Factores Situacionales

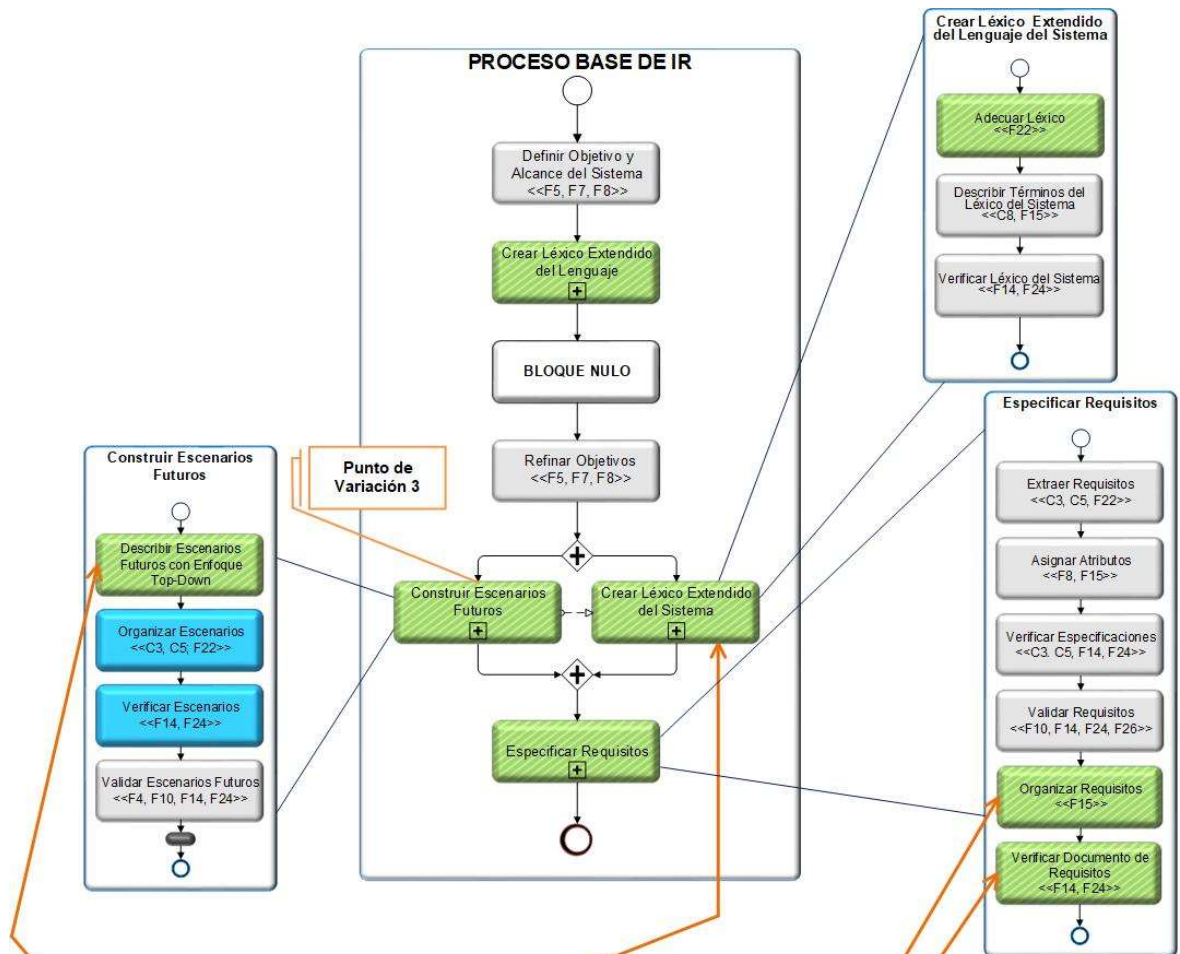
Figura 5-4. Ejemplo de adaptación del proceso de IR para la situación inicial de un proyecto

Mientras progresa el proyecto, algunos factores pueden evolucionar o precisarse con mayor confiabilidad. Esto se ejemplifica en la Figura 5-5, donde la situación antes descrita se modifica en el Punto de Variación 3.

Como se puede observar en el formulario de la Figura 5-5, se han modificado los valores asignados a los factores *Reingeniería en el proceso de negocio* (F4) pasó a “Alta” y *Exigencia de producir el documento ERS* (F20) pasó a valor “Sí”, donde ambos mejoraron su nivel de confianza (además de otros factores). La nueva asignación de valores causa la readaptación del proceso de IR a partir del Punto de Variación 3, de la siguiente manera:

- En el subproceso *Construir Escenarios Futuros* el bloque de proceso *Describir Escenarios Futuros con Enfoque Basado en el Léxico* es reemplazado por el bloque *Describir Escenarios Futuros con Enfoque Top-Down* que se utiliza dado que no se dispone de EA y el factor *Reingeniería en el proceso de negocio* tiene valor “Alta” (la condición C12 es verdadera).
- El bloque de proceso *Crear Léxico Extendido del Sistema*, que inicialmente se omitía, debe utilizarse dado que hay “Alta” *Reingeniería en el proceso de negocio* (la condición C5 es falsa).
- En el bloque de proceso *Especificar Requisitos* se utilizan los bloques internos *Organizar Requisitos* y *Verificar Documento de Requisitos* (ambos omitidos inicialmente) debido a que el factor *Exigencia de producir el documento ERS* tiene valor “Sí” (la condición C16 es falsa).

Con estas modificaciones también cambian los valores pasados como parámetros a los bloques de proceso. Por ejemplo, en el bloque de proceso *Organizar Escenarios*, mencionado previamente, los EF Integradores deberán contener los vínculos a los términos del LEL del Sistema dado que en la nueva situación se dispone de tal artefacto.



	FACTORES DEL CONTEXTO	NIVEL DE CONFIANZA		
		Seguro	Raz. Seguro	Dudoso
1.	Complejidad del contexto Muy Alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input checked="" type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.	Tipo de Cliente Mercado <input type="checkbox"/> Potencial <input type="checkbox"/> Cliente Especifico <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	Novedad del contexto Si <input type="checkbox"/> No <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	Reingeniería en el proceso de negocio Muy Alta <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	Volatilidad del contexto Muy Alta <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	Volatilidad de los requerimientos del cliente Muy Alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input checked="" type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
7.	Inconsistencias en el contexto Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input checked="" type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
8.	Conflicto de intereses de usuarios Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input checked="" type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	Rotación de usuarios Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Baja <input checked="" type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	Localización de usuarios Distribuidos <input type="checkbox"/> Colocalizados <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

	FACTORES DEL PROYECTO	NIVEL DE CONFIANZA		
		Seguro	Raz. Seguro	Dudoso
11.	Conocimiento previo del dominio Muy Alto <input type="checkbox"/> Alto <input checked="" type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12.	Envergadura del proyecto Muy Grande <input type="checkbox"/> Grande <input type="checkbox"/> Mediana <input checked="" type="checkbox"/> Pequeña <input type="checkbox"/> Muy Pequeña <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13.	Rotación del equipo desarrollador Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Baja <input checked="" type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14.	Fiabilidad necesaria del software Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input checked="" type="checkbox"/> Baja <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15.	Reuso de artefactos de requisitos existentes LEL <input type="checkbox"/> EA <input type="checkbox"/> EF <input type="checkbox"/> LEL-S <input type="checkbox"/> Requisitos <input checked="" type="checkbox"/> No <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16.	Creación de artefactos de requisitos para reuso LEL <input checked="" type="checkbox"/> EA <input type="checkbox"/> LEL-S <input type="checkbox"/> Requisitos <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
17.	Pre-Rastreabilidad de los requisitos Total <input type="checkbox"/> Parcial <input checked="" type="checkbox"/> No <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18.	Post-Rastreabilidad de los requisitos Si <input type="checkbox"/> No <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19.	Granularidad de Rastreabilidad de los requisitos Individual <input type="checkbox"/> Grupal <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20.	Exigencia de producir el documento ERS Si <input checked="" type="checkbox"/> No <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
21.	Restricciones de tiempo y recursos para el proyecto Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input checked="" type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
22.	Modelo de proceso de software Cascada <input type="checkbox"/> Incremental <input checked="" type="checkbox"/> Iterativo <input type="checkbox"/> Transf. Formal <input type="checkbox"/> Otro <input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
23.	Tipo de incremento Nueva Funcionalidad <input checked="" type="checkbox"/> Mejora a Funcionalidad <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
24.	Experiencia en el proceso de requisitos Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input checked="" type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
25.	Diferencias idiomáticas y/o culturales Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input checked="" type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
26.	Diferencias temporales entre los involucrados Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Formulario de Evaluación de los Factores Situacionales

Figura 5-5. Ejemplo de readaptación del proceso de IR en el Punto de Variación 3 para la nueva situación

Capítulo 6

Validación de la Propuesta de Adaptación

6.1 Método de Validación

El propósito de este capítulo es presentar la validación realizada de la adaptación de la estrategia de IR propuesta en esta tesis. Para ello se ha optado por planificar y llevar a cabo un estudio de caso controlando algunos aspectos. Se ha elegido este método de validación debido a la limitación para realizarla mediante un experimento controlado, el cual implicaría disponer de suficientes proyectos que involucrara una combinación de valores de los factores situacionales efectivamente irrealizable.

Fundamentalmente el estudio se elaboró para comparar las tres variantes del proceso de requisitos que se consideran más significativas: i) producir los requisitos generando cada uno de los modelos propuestos en el proceso base; ii) producir los requisitos omitiendo la construcción del LEL del UdeD; y iii) producir los requisitos omitiendo la construcción de los EA. El objetivo de este estudio era poder establecer qué variante lograba una mejor calidad de los requisitos cuando se trataba de un dominio de aplicación relativamente conocido por los ingenieros de requisitos, considerando también el esfuerzo relativo insumido.

6.2 Descripción del Estudio de Caso

La idea principal ha sido realizar un estudio de carácter descriptivo y parcialmente explicativo, desarrollándose 5 proyectos de IR en un contexto de aplicación y 7 proyectos más en otro contexto, utilizando las tres variantes del proceso de IR antes mencionadas en cada contexto.

El estudio se ha desarrollado en un contexto académico, en el marco de la asignatura Ingeniería de Requerimientos que se dicta para estudiantes de 4º año de la carrera de Ingeniería en Informática de Universidad Nacional de La Matanza. Todos los sujetos son estudiantes que realizan la experiencia como parte de un proyecto integrador de la asignatura, lo cual implica que estén involucrados y comprometidos en la realización de sus tareas. Se conformaron en forma aleatoria

12 grupos, de 4 a 6 alumnos. Hay que destacar que ningún estudiante tenía experiencia en el proceso de IR, y todos recibieron la misma capacitación a lo largo de la cursada.

Se seleccionaron dos casos referidos a sistemas en funcionamiento en la Universidad Nacional de La Matanza, cuyo dominio es conocido por los sujetos, a saber:

Caso C_A: Sistema de Autogestión de Alumnos

Caso C_B: Sistema de Gestión de Biblioteca

Cada grupo desarrolló uno de los dos casos, disponiendo del documento objetivo y alcance del sistema y la posibilidad de acceso al uso del sistema existente en cada caso. Este proyecto se completó durante un cuatrimestre (16 semanas) en 2018, planteando una solución a través de un Documento de Especificación de Requisitos bajo el estándar IEEE Std 29148:2011.

Dadas las características de los casos seleccionados y de los grupos de alumnos que participaron, en la Tabla 6-1 se indican los factores situacionales relevantes para decidir la construcción del LEL del UdeD y de los EA.

Tabla 6-1. Detalle de la situación para los casos bajo estudio

Factor Situacional	Caso C _A	Caso C _B
Complejidad del contexto (F1)	Baja	Baja
Tipo de cliente (F2)	Cliente Específico	Cliente Específico
Novedad del contexto (F3)	No	No
Inconsistencias en el contexto (F7)	Baja	Baja
Conocimiento previo del dominio (F11)	Muy Alto	Alto
Envergadura del proyecto (F12)	Pequeña	Pequeña
Volatilidad del contexto (F5)	Baja	Baja
Experiencia en el proceso de requisitos (F24)	Muy Baja	Muy Baja
Rotación de usuarios (F9)	Muy Baja	Muy Baja
Rotación del equipo desarrollador (F13)	Baja	Baja
Localización de usuarios (F10)	Colocalizados	Colocalizados
Diferencias idiomáticas y/o culturales (F25)	Muy Baja	Muy Baja
Diferencias temporales entre los involucrados (F26)	Muy Baja	Muy Baja
Creación de artefactos de requisitos para reuso (F16)	No	No
Pre-Rastreabilidad de los requisitos (F17)	Parcial	Parcial

Cabe mencionar que los alumnos usan habitualmente el Sistema de Autogestión de Alumnos para toda su actividad administrativa en la universidad, mientras que el uso del Sistema de Gestión de Biblioteca suele ser menos frecuente, conociendo algunas actividades con cierto nivel de detalle y desconociendo algunas otras, principalmente las vinculadas a otros tipos de lectores no alumnos. Esto es lo que marca la diferencia en la situación entre el Caso C_A y el Caso C_B: el nivel del factor situacional Conocimiento previo del dominio (F11).

En base a los valores de estos factores, la recomendación de adaptación del proceso de IR para el caso Sistema de Autogestión de Biblioteca es no construir el LEL del UdeD (condición C3=V) y no construir los EA (condición C4=V), mientras que para el caso Sistema de Gestión de Biblioteca es no construir el LEL del UdeD (condición C3=V) y construir los EA (condición C4=F). Ver estas condiciones en la subsección 4.3.1.

Para el estudio se han seleccionado las tres variantes del proceso de mayor relevancia en lo que respecta a los modelos que se construyen. Seguidamente se enumeran los modelos generados en cada variante:

Variante V₁: LEL del UdeD – EA – EF – LEL del Sistema – ERS

Variante V₂: LEL del UdeD – EF – LEL del Sistema – ERS

Variante V₃: EA – EF – LEL del Sistema – ERS

En base a lo recomendado para cada caso, esto indica que el caso C_A está más cercano a la variante V₂, mientras que el caso C_B se desarrollaría siguiendo la variante V₃, o eventualmente la V₁.

La Tabla 6-2 detalla el caso y la variante del proceso con la que trabajó cada grupo.

Tabla 6-2. Asignación de casos y variantes por grupo

Caso	Grupos	Variante del Proceso
C _A : Sistema de Autogestión de Alumnos	G ₆ - G ₇ - G ₈	V ₁
	G ₉ - G ₁₀ - G ₁₁	V ₂
	G ₁₂	V ₃
C _B : Sistema de Gestión de Biblioteca	G ₁ - G ₂	V ₁
	G ₃ - G ₄	V ₂
	G ₅	V ₃

Un buen indicador que sirve de base para comparar los resultados al aplicar las distintas variantes del proceso son los EF obtenidos por cada grupo. Esto es así debido a que, por un lado, son elaborados en cualquier variante y, por otro lado, tal como se ha mencionado anteriormente, dichos escenarios concentran los requisitos funcionales del software, como también los no funcionales en sus descripciones (ver subsecciones 2.3.4 y 4.3.7). En otras palabras, los requisitos se obtienen a partir de los EF.

No obstante, comparar únicamente los valores de las mediciones tomadas de cada modelo de EF obtenido no permite visualizar el resultado real alcanzado, dado que dichos valores dependen de cómo los grupos organizaron los escenarios y con qué nivel de detalle los describieron. Por ejemplo, para el Sistema de Autogestión de Alumnos el grupo G_8 definió un EF para la funcionalidad *Consultar historial académico*, mientras que para cubrir la misma funcionalidad el grupo G_9 definió 4 EF y el grupo G_{11} definió 3 EF. Por lo tanto, se hizo necesario establecer las funcionalidades esperadas para cada sistema, para lo cual se consolidaron los objetivos de los EF generados por cada grupo, teniendo como base el objetivo y alcance inicial establecido. Así es que para el Sistema de Autogestión de Alumnos se identificaron 26 funcionalidades, y para el Sistema de Gestión de Biblioteca 22 funcionalidades (ver detalles en el Anexo D). Cabe mencionar que no debe interpretarse que estas dos listas de funcionalidades abarcan la totalidad de servicios que cada sistema debería efectivamente cubrir, sino que es un mero parámetro de comparación a utilizar respecto al grado relativo de cobertura alcanzado por cada grupo, debiéndose tener presente la imposibilidad de establecer con certeza el nivel de completitud alcanzado. Este aspecto de la falacia de la completitud ha sido ampliamente tratado en la literatura (Rochowiak, 1988).

Teniendo presente lo anterior, se definió un conjunto de métricas para luego permitir comparar los resultados conseguidos con la aplicación de cada variante del proceso en cada caso. La Tabla 6-3 detalla cada una de las métricas definidas.

Tabla 6-3. Métricas para comparar la aplicación de las variantes del proceso de IR¹⁶

#	Código	Definición	Fórmula	Definición de Parámetros
1	TFC	Total de funcionalidades cubiertas con los EF	A	A: Cantidad de funcionalidades para el alcance del sistema que se cubren con los EF del grupo
2	PFC	Porcentaje de funcionalidades cubiertas	A/T*100	A: Cantidad de funcionalidades cubiertas por los EF del grupo T: Total de funcionalidades para el alcance del sistema
3	PFCxV	Porcentaje promedio de funcionalidades cubiertas por variante del proceso	A/N	A: Suma del porcentaje de funcionalidades cubiertas con los EF por cada grupo que aplica la variante del proceso en cada caso N: Cantidad de grupos que aplican la variante del proceso en cada caso
4	TEF	Total de EF	A	A: Cantidad de EF del grupo
5	TEP	Total de episodios en los EF	A	A: Cantidad total de episodios en los EF del grupo
6	EPxF	Promedio de episodios por funcionalidad	T/A	T: Cantidad total de episodios en los EF del grupo A: Total de funcionalidades definidas para el alcance del sistema
7	EPxEF	Promedio de episodios por EF	T/A	T: Cantidad total de episodios en los EF del grupo A: Cantidad de EF del grupo
8	EPxEFxV	Promedio de episodios por EF por variante del proceso	A/N	A: Suma del promedio de episodios por EF de cada grupo que aplica la variante del proceso en cada caso N: Cantidad de grupos que aplican la variante del proceso en cada caso
9	TEX	Total de excepciones en los EF	A	A: Cantidad total de excepciones en los EF del grupo
10	EXxEF	Promedio de excepciones por EF	T/A	T: Cantidad total de excepciones en los EF del grupo A: Cantidad de EF del grupo
11	TACT	Total de actores en los EF	A	A: Cantidad total de actores sin repetición en todos los EF del grupo

6.3 Análisis de los Resultados

Una vez concluidos los proyectos de IR, se tomaron las mediciones según las métricas definidas en la Tabla 6-3. Cada proyecto implica la realización de una variante del proceso de IR para un caso dado y por un grupo dado. En las Tablas 6-4 y 6-5 se presenta un resumen de las mediciones obtenidas para el análisis de cada caso.

Tabla 6-4. Resumen de mediciones obtenidas para el Sistema de Autogestión de Alumnos

Proyecto	Total Func.	TFC (1)	PFC (2) %	PFCxV (3) %	TEF (4)	TEP (5)	EPxF (6)	EPxEF (7)	EPxEFxV (8)	TEX (9)	EXxEF (10)	TACT (11)
G ₆ V ₁ CA	26	16	61,5	69,23	12	129	8,06	10,75	8,06	17	1,42	3
G ₇ V ₁ CA		18	69,2		18	133	7,39	7,39		20	1,11	4
G ₈ V ₁ CA		20	76,9		25	151	7,55	6,04		19	0,76	4
G ₉ V ₂ CA		17	65,3	66,67	25	137	8,06	5,48	5,45	15	0,60	6
G ₁₀ V ₂ CA		19	73,0		20	126	6,63	6,30		26	1,30	5
G ₁₁ V ₂ CA		16	61,5		18	82	5,13	4,56		38	2,11	4
G ₁₂ V ₃ CA		25	96,1		96,15	27	173	6,92		6,41	6,41	24

(1) TFC: Total de funcionalidades cubiertas con los EF

(2) PFC: Porcentaje de funcionalidades cubiertas

(3) PFCxV: Porcentaje promedio de funcionalidades cubiertas por variante del proceso

(4) TEF: Total de EF

(5) TEP: Total de episodios en los EF

(6) EPxF: Promedio de episodios por funcionalidad

(7) EPxEF: Promedio de episodios por EF

(8) EPxEFxV: Promedio de episodios por EF por variante del proceso

(9) TEX: Total de excepciones en los EF

(10) EXxEF: Promedio de excepciones por EF

(11) TACT: Total de actores en los EF

¹⁶ Todos los cálculos excluyen a los EF Integradores.

Tabla 6-5. Resumen de mediciones obtenidas para el Sistema de Gestión de Biblioteca

Proyecto	Total Func.	TFC (1)	PFC (2) %	PFCxV (3) %	TEF (4)	TEP (5)	EPxF (6)	EPxEF (7)	EPxEFxV (8)	TEX (9)	EXxEF (10)	TACT (11)
G ₁ V ₁ C _B	22	14	63,6	65,91	14	124	8,86	8,86	8,40	33	2,36	8
G ₂ V ₁ C _B		15	68,1		14	111	7,40	7,93		14	1,00	9
G ₃ V ₂ C _B		14	63,6	59,09	15	57	4,07	3,80	3,94	12	0,80	7
G ₄ V ₂ C _B		12	54,5		14	57	4,75	4,07		19	1,36	9
G ₅ V ₃ C _B		19	86,3		24	137	7,21	5,71		5,71	28	1,17

(1) TFC: Total de funcionalidades cubiertas con los EF
 (2) PFC: Porcentaje de funcionalidades cubiertas
 (3) PFCxV: Porcentaje promedio de funcionalidades cubiertas por variante del proceso
 (4) TEF: Total de EF
 (5) TEP: Total de episodios en los EF

(6) EPxF: Promedio de episodios por funcionalidad
 (7) EPxEF: Promedio de episodios por EF
 (8) EPxEFxV: Promedio de episodios por EF por variante del proceso
 (9) TEX: Total de excepciones en los EF
 (10) EXxEF: Promedio de excepciones por EF
 (11) TACT: Total de actores en los EF

Tanto en el caso del Sistema de Autogestión de Alumnos como en el Sistema de Gestión de Biblioteca, se observa en las Tablas 6-4 y 6-5 que la cobertura de funcionalidades entre la V₁ y la V₂ son bastante similares (ver métrica 3) con un nivel de confianza superior al 95%, sin embargo, esto no ocurre en el nivel de detalle de la descripción de dichas funcionalidades (ver métrica 8), donde se observa una diferencia pronunciada en el promedio de episodios por EF entre ambas variantes (también con un nivel de confianza superior al 95%). En ambos casos, el mayor nivel de detalle con que se describen las funcionalidades se alcanza con la V₁ que construye los EA, respecto de la V₂. No es posible realizar estas comparaciones con la V₃ dado la única muestra disponible en cada caso.

Debe aclararse que para los cálculos de nivel de confianza se ha utilizado la distribución t de Student (Cavazos, 1988), considerando que las curvas para las variantes V₁ y V₂ son aproximadamente gaussianas y se tienen muestras pequeñas en ambos casos. Ver en el Anexo D las Tablas D-3 y D-4 con el detalle de resultados estadísticos con un nivel de significancia $\alpha = 0,05$.

En el caso C_B se observa una cantidad similar de actores identificados entre las variantes V₁ y V₂ con una probabilidad superior al 95%, lo que no ocurre en el caso C_A (ver métrica 11 en Tablas 6-4 y 6-5). Mientras que al estudiar la tasa de excepciones por EF (ver métrica 10) surge que no hay diferencias significativas entre la V₁ y la V₂ para el caso C_A, con un nivel de confianza superior al 95%, pero ocurre lo opuesto para el caso C_B. Al no observarse causas aparentes provenientes de ambos dominios, se puede decir que estas observaciones están dando un indicio de falencias en las heurísticas referidas a la elaboración de excepciones e identificación de actores.

En lo que sigue se analizan las recomendaciones de adaptación del proceso de IR para la situación de cada caso, en cuanto a la construcción o no del LEL del UdeD y de los EA, estudiando los resultados obtenidos en el promedio de episodios por EF para cada variante aplicada, pudiendo o no sobrar el LEL del UdeD, pudiendo o no sobrar los EA, e incluso pudiendo faltar los EA para cada caso según la variante utilizada (ver Tabla 6-6).

Comparando la variante V_1 con la variante V_2 , surge con una probabilidad superior al 95% que hacer los EA es siempre beneficioso, tanto si se hubiera recomendado hacerlos, como si se hubiera recomendado no hacerlos.

Por su parte, comparando el caso C_A con el caso C_B en la variante V_2 , se detecta, con una probabilidad superior al 90%, que el no hacer los EA es más perjudicial cuando se ha recomendado hacerlos.

En otras palabras, surge que, si bien las reglas que deciden sobre la necesidad de hacer o no los EA están correctamente enfocadas, son demasiado laxas, recomendando omitir esa actividad en situaciones en las que no corresponde.

Sin evidencia estadística, debido a la escasa cantidad de casos de la variante V_3 , es posible observar que aparentemente las reglas que recomiendan la omisión del LEL del UdeD son también algo laxas, pero en un grado menor.

Tabla 6-6. Análisis de la recomendación de adaptación del proceso de IR para cada caso

	RECOMENDACIÓN DE ADAPTACIÓN	
	Caso C_A	Caso C_B
	No LEL y No EA	No LEL y EA
Variante V_1	Sobra LEL y Sobra EA	Sobra LEL
LEL y EA	10,75 - 7,39 - 6,04 \Rightarrow 8,06	8,86 - 7,93 \Rightarrow 8,39
Variante V_2	Sobra LEL	Sobra LEL y Falta EA
LEL y No EA	5,48 - 6,30 - 4,56 \Rightarrow 5,45	3,80 - 4,07 \Rightarrow 3,94
Variante V_3	Sobra EA	-----
No LEL y EA	6,41	5,71

Dadas las diferencias en la cantidad promedio de episodios para cada variante, se realizó un análisis más profundo sobre dos de las funcionalidades más relevantes

de cada caso, realizando un estudio estadístico basado en la distribución gaussiana y un estudio semántico de cada uno de los EF que representan dichas funcionalidades. En la Tabla 6-7 se muestra la cantidad de episodios y excepciones de los EF *Inscribir a materias* e *Inscribir a finales* del Sistema de Autogestión de Alumnos. Por su parte, en la Tabla 6-8 se presenta el mismo detalle para los EF *Realizar un préstamo* y *Devolución de un préstamo*, del Sistema de Gestión de Biblioteca.

Tabla 6-7. Resultados del análisis de dos EF del Sistema de Autogestión de Alumnos

Proyecto	EF: INSCRIBIR A MATERIAS			EF: INSCRIBIR A FINALES		
	Episodios	Promedio de episodios por variante	Excepciones	Episodios	Promedio episodios por variante	Excepciones
G ₆ V ₁ CA	31	32,67	2	13	12,67	0
G ₇ V ₁ CA	28		4	16		2
G ₈ V ₁ CA	39		1	9		0
G ₉ V ₂ CA	12	11,33	1	5	7,00	2
G ₁₀ V ₂ CA	13		0	10		0
G ₁₁ V ₂ CA	9		1	6		0
G ₁₂ V ₃ CA	18	18	1	11	11,00	1

Tabla 6-8. Resultados del análisis de dos EF del Sistema de Gestión de Biblioteca

Proyecto	EF: REALIZAR UN PRÉSTAMO			EF: DEVOLUCIÓN DE UN PRÉSTAMO		
	Episodios	Promedio de episodios por variante	Excepciones	Episodios	Promedio episodios por variante	Excepciones
G ₁ V ₁ CB	12	16,50	3	25	24,50	2
G ₂ V ₁ CB	21		6	24		1
G ₃ V ₂ CB	6	6,00	4	10	10,50	1
G ₄ V ₂ CB	6		3	11		2
G ₅ V ₃ CB	20	20	4	16	16,00	1

En base al estudio estadístico, surge para ambos casos, con una probabilidad superior al 80%, que construir los EA (V_1) frente a no construirlos (V_2), habiéndose recomendado construirlos o no, ha sido más beneficioso en cuanto a lograr una mayor cantidad de episodios por EF, lo que está en concordancia con lo estudiado previamente para el conjunto total de EF. En cuanto a las excepciones, se observa con una probabilidad superior al 80% una gran variabilidad en el nivel de detalle de excepciones por cada EF entre la V_1 y la V_2 . Obviamente, no es posible realizar ninguna observación estadística relativa a la V_3 .

Para ambos casos, se analizó semánticamente los dos EF seleccionados, observándose que los episodios de quienes siguieron las variantes V_1 y V_3 son más completos y bien detallados. En tanto que los grupos que aplicaron la variante V_2 , tienen episodios muy generales de alto nivel, sin mucho detalle sobre cómo efectivamente deberían realizarse las actividades a través del nuevo sistema de software, ni qué información específica debería manejarse, ya sea para control, presentación, carga o consulta. Esto podría indicar que la heurística de construcción de EF cuando se parte de un LEL del UdeD sin EA no es lo suficientemente precisa (V_2).

En cuanto a los tiempos promedios insumidos al aplicar el proceso en cada una de sus variantes, se ha tomado como punto de referencia el tiempo insumido para realizar la estrategia completa (variante V_1). En el caso del Sistema de Gestión de Biblioteca, los equipos que aplicaron la variante V_2 tardaron un 41% menos respecto a la variante V_1 , y los que aplicaron la variante V_3 un 18% menos, lo cual es lógico debido al tiempo requerido para producir todos los modelos de la estrategia de IR en la variante V_1 . Por otro lado, la diferencia entre las variantes V_2 y V_3 significa que llevó más tiempo construir los EA que el LEL del UdeD, lo cual también es razonable debido a que los EA exigen elicitar y elaborar más información y de distinto nivel de granularidad. Sin embargo, los grupos que construyeron los EA tuvieron mejores resultados en cuanto a la cobertura de funcionalidades con los EF y el nivel de detalle alcanzado en sus descripciones (en base al estudio estadístico de cantidad promedio de episodios por EF y al análisis semántico realizado sobre los EF más relevantes). En cuanto al Sistema de Autogestión de Alumnos, se observa que aquellos que aplicaron la variante V_2 tardaron un 44% menos con relación a la variante V_1 , en cambio, los que aplicaron la variante V_3 tardaron un 15% más de tiempo respecto a la variante V_1 , lo cual podría tener relación con que quienes aplicaron la variante V_3 tuvieron una cobertura de las funcionalidades muy superior a la obtenida por los que aplicaron las variantes V_1 y V_2 .

En otras palabras, en el caso del Sistema de Autogestión de Alumnos surge, con una probabilidad superior al 80%, que existen notorias diferencias de tiempo entre la V_1 y la V_2 ; esto también se refleja en la única muestra de la V_3 respecto de las otras dos variantes. Por el contrario, en el caso del Sistema de Gestión de

Biblioteca no se observan las mismas notorias diferencias, con la misma probabilidad.

6.4 Conclusiones de la Validación

El estudio realizado arroja indicios preliminares para poder concluir que la calidad de los requisitos no es independiente del modo en que se aplica la estrategia de IR. Se observa que partiendo de sujetos con la misma experiencia en el dominio de aplicación y sin experiencia en el proceso de requisitos, los EF generados tuvieron mayor calidad cuando estos se construyeron con una fase de aprendizaje basada en la construcción de los EA, en comparación con aquellos EF construidos con una fase de aprendizaje basada únicamente en la creación del modelo LEL del UdeD. Aunque, para esta última variante, también es posible la existencia de una heurística débil para construir EF sin disponer de EA.

En otros términos, los resultados estadísticos apoyan la recomendación dada por las reglas de adaptación de crear los EA para un caso, mientras que esto no ocurre para el otro caso donde según los factores situacionales no era necesario construirlos. Esto podría también ser interpretado en el sentido que la nula experiencia de los ingenieros de requisitos en el proceso de IR puede ser un factor más relevante de lo supuesto frente al conocimiento del dominio de aplicación, aun cuando las heurísticas fuesen lo suficientemente precisas.

Por otro lado, considerando el tamaño de la muestra obtenida para cada variante, el estudio debería replicarse con nuevas muestras en las variantes estudiadas para arribar a conclusiones relativamente generalizables.

Finalmente, a fin de realizar una correcta interpretación de los resultados, considerando sus limitaciones, se ha analizado las amenazas a la validez a las que podría estar sujeto este estudio. A continuación, se listan las principales amenazas identificadas y la forma en que han sido tratadas:

- *Capacidad de los sujetos*: la diferencia de capacidad y experiencia de los sujetos puede sesgar los resultados. Para mitigar esta amenaza los grupos fueron formados al azar lo cual minimiza el problema de que no todos los sujetos tenían las mismas capacidades.

- *Seguimiento del proceso*: es posible que los sujetos no cumplan con el proceso establecido, o incluso que lo apliquen de distinta forma. Para evitar esta amenaza los alumnos fueron supervisados semanalmente por los docentes con los mismos lineamientos. De este modo esta amenaza se minimiza, pero no se neutraliza.
- *Sujetos sin experiencia*: el estudio se ha realizado en un contexto académico. Los sujetos no tienen o tienen poca práctica en la industria. Además, no cuentan con experiencia en la estrategia de IR. Los grupos recibieron instrucciones generales del proceso de IR a llevar a cabo. Para minimizar esta amenaza, a medida que avanzaban en el proceso se fueron dando explicaciones detalladas de cada etapa del mismo. A pesar de ello, los resultados no son generalizables de forma directa a profesionales u otro tipo de sujetos.

Debido a limitaciones en la cantidad de grupos utilizada para el estudio, cada variante del proceso no ha sido aplicada por la misma cantidad de grupos, lo que puede introducir una amenaza a la validez en lo referido a las comparaciones sobre las mediciones obtenidas y a las conclusiones arribadas a partir de estas. De todas maneras, las diferencias en la cantidad de muestras no son tan significativas como para imposibilitar las comparaciones, aunque deben ser tenidas en consideración respecto a los resultados obtenidos.

Capítulo 7

Conclusiones y Futuras Líneas de Investigación

7.1 Conclusiones

Proporcionar guías sobre cómo adaptar el proceso de requisitos a una situación particular de un proyecto de software, facilita planear su adaptación antes de arrancar con él, en lugar de realizar correcciones a medida que ocurren contingencias previsible. Como se expuso en el capítulo 2, hay varias propuestas orientadas a la definición y a la mejora del proceso de IR, ajustándolo a las características particulares de un proyecto. Sin embargo, estas aproximaciones no son tratadas con la profundidad necesaria y en la mayoría de los casos no resultan claras en cuanto al modo de llevar dichas adaptaciones a la práctica.

La presente tesis tiene como objetivo aportar una mejora sustentada en flexibilizar una estrategia de IR basada en modelos de lenguaje natural, a partir de un refinamiento del proceso subyacente. La mejora consiste entonces en la adaptación sistemática y dinámica de la estrategia de IR de acuerdo con el contexto situacional en el que dicha estrategia deba ser aplicada, soportando la evolución de dicho contexto. Se considera posible que esta propuesta pueda implementarse con ajustes menores en otras estrategias de IR.

Se ha identificado un conjunto de 26 factores situacionales que comprometen la estrategia de IR, en el sentido que causan algún ajuste en las actividades a realizar o en los artefactos que se deben producir. Algunos de estos factores están relacionados con el contexto de aplicación mientras que otros con el proyecto en sí mismo. Dado que la situación surge de la combinación del conjunto de factores con sus respectivos valores, se ha estudiado los diferentes tipos de dependencias que pueden existir entre estos. Varios de dichos factores pueden estimarse con precisión antes de iniciar el proceso de IR, para permitir que el propio proceso pueda ser adaptado anticipadamente. Sin embargo, otros factores son más difíciles de estimar al inicio del proyecto y, por otro lado, algunos pueden evolucionar a lo largo del proceso debido a cambios del propio contexto. Por lo cual, es importante

tener en cuenta el nivel de confianza con el que los ingenieros de requisitos estiman cada factor situacional.

Se han determinado los hitos en los que la estrategia puede variar y la combinación de factores situacionales que influyen en cada uno de dichos puntos de variación. El proceso de IR se ha fragmentado en bloques de proceso definiéndose los diferentes tipos de bloques de proceso necesarios. Se ha establecido cuáles se utilizan siempre independientemente de la valoración de los factores y cuáles son variantes para los distintos caminos alternativos. Lo anterior, a su vez, favorece el reuso de los bloques de proceso y, por otra parte, brinda mayor flexibilidad a la adaptación del proceso en sí mismo.

Se ha definido un método para guiar la adaptación del proceso a cada situación específica, basado en reglas de adaptación que combinan factores situacionales. Para adaptar el proceso, el método establece qué bloques seleccionar para formular o reformular el proceso de IR en función de la valoración asignada a los factores situacionales en cada punto de variación. Así también se ha descrito la influencia que ciertos factores pueden tener en el modo de realizar las actividades involucradas en los bloques de proceso, asegurando de este modo que dichas actividades y las técnicas asociadas sean las más adecuadas para la situación específica. Debe tenerse presente que también la adaptación puede impactar en los productos del proceso, de manera tal, que algunos modelos puedan también requerir ajustes.

Dado que la adaptación situacional del proceso depende fuertemente de la calidad de los factores situacionales, se ha analizado la estimación y evolución de los mismos a lo largo del proceso de requisitos. Para ello se desarrolló un estudio de caso basado en 35 proyectos reales de la industria, lo que ha permitido establecer aquellos factores situacionales que cambian más frecuentemente o que son más difícil de estimar. Como consecuencia, se ha establecido a qué factores se les debe prestar mayor atención antes y durante la definición y readaptación del proceso, teniendo en consideración que la valoración de algunos factores pueden impactar en la valoración de otros y, como consecuencia puede desencadenar otro tipo de ajustes en el proceso.

A partir del estudio de caso mencionado, se observó la evolución natural y contingente de situaciones, lo que remarcó la importancia de poder redefinir las actividades restantes del proceso de IR mientras el proyecto está en curso. Es por ello que se ha diseñado un proceso de adaptación de carácter dinámico del proceso de requisitos, que aplica reglas de adaptación en los puntos de variación según la ponderación del nivel de confianza asignado al valor y al tipo de evolución de los factores situacionales, considerando que dicha evolución puede provenir no solo de un cambio real en la situación sino también de una mejor comprensión de esa situación. Para ello, el proceso de adaptación propuesto realiza reevaluaciones de los factores en los puntos de variación, guiando un reajuste hacia adelante del proceso.

Finalmente, se ha realizado un estudio para comparar las principales variantes del proceso de IR: i) aplicando todas las etapas del proceso de IR, ii) omitiendo la construcción del LEL del UdeD, y iii) omitiendo la construcción de los EA. En el estudio se utilizaron dos dominios de aplicación, disponiendo de un total de 7 muestras. Los sujetos que participaron contaban con la misma experiencia en el dominio de aplicación y sin experiencia en el proceso de requisitos. Se observó que los sujetos obtuvieron mayor calidad en los requisitos cuando llevaron a cabo el proceso de IR con una fase de aprendizaje basada en el comportamiento actual del dominio, es decir, quienes aplicaron las variantes i) y iii) que incluía construir los EA. Sin embargo, estos son indicios preliminares que deben ser ratificados con otras réplicas del estudio, y bajo otros dominios, e incluyendo otras variantes. Las mediciones obtenidas aún son escasas para obtener conclusiones generalizables. No obstante, los resultados alcanzados son válidos para reforzar hipótesis planteadas e investigaciones futuras.

En resumen, el presente trabajo de tesis ha permitido poner en evidencia que los procesos de requisitos son sensiblemente influenciados por el contexto en el que se aplican y que esa influencia puede, en muchas situaciones, ser controlada y aun aprovechada para mejorarlos más allá del proceso base de partida.

7.2 Resultados Publicados de la Investigación

Los resultados alcanzados durante la realización de este trabajo fueron difundidos a través de las siguientes publicaciones:

- *Proceso de Requisitos Adaptable a Factores Situacionales*, Hadad G., Ledesma V. & Doorn J., WICC 2014 – XVI Workshop de Investigadores en Ciencias de la Computación, ISBN: 978-950-34-1084-4, pp.448-452. Tierra del Fuego, Argentina, mayo 2014.
- *Adaptabilidad y Completitud en Procesos de Requisitos*, Doorn J., Hadad G., Litvak C., Ledesma V. & Vera A., 1ra. Jornada de Investigación Interdepartamental, Línea Temática N°12: Desarrollos en Informática, Mesa 19, UNLaM. Buenos Aires, Argentina, septiembre 2014.
- *Adaptación Dinámica de un Proceso de Requisitos Orientado al Cliente*, Ledesma V., Hadad G. & Doorn J., CONAISI 2016, 4to Congreso Nacional de Ingeniería Informática / Sistemas de Información, ISSN: 2347-0372. Salta, Argentina, noviembre 2016.
- *Evolución de los Factores Situacionales durante el Proceso de Requisitos*, Ledesma, V., Hadad, G., Doorn, J., Mighetti J., Bedetti, N. & Elizalde M., WICC 2018 – XX Workshop de Investigadores en Ciencias de la Computación, ISBN: 978-987-3619-27-4, pp.578-582. Corrientes, Argentina, abril 2018.
- *Estimación de los Factores Situacionales del Proceso de Requisitos*, Ledesma, V., Hadad, G., Doorn, J. & Bedetti, N., WER 2018 - 21st Workshop on Requirements Engineering, ISBN: 978-85-907171-1-9. Río de Janeiro, Brasil, septiembre 2018.
- *Dynamic Situational Adaptation of a Requirements Engineering Process*, Hadad G., Doorn J. & Ledesma V., en M. Khosrow-Pour. Encyclopedia of Information Science and Technology, Fourth Edition. Hershey, PA: IGI Global, Information Science Reference, ISBN: 9781522522553, pp.7422 – 7434, 2018.

7.3 Líneas Futuras de Investigación

Durante el desarrollo de la tesis, se han detectado posibles líneas de investigación con las que se puede mejorar y ampliar la presente propuesta. A continuación, se enumeran algunas consideradas de mayor interés:

- Analizar la posibilidad de existencia de otros factores situacionales que impacten en fases posteriores del proceso de desarrollo de software y que pudieran entonces tener una influencia en la definición del proceso mismo de requisitos.

- Identificar la posible existencia de situaciones no planteadas en el presente trabajo que puedan afectar la adaptación propuesta, principalmente en contextos donde la estrategia de IR estudiada no ha sido puesta a prueba o llevada a la práctica profesional. Un caso particular se ha presentado durante la revisión final de la presente tesis, donde se detectó la necesidad de incorporar un factor situacional relacionado con el nivel de informatización de los clientes y usuarios, el cual puede afectar, por ejemplo, la creación o no del LEL del Sistema.
- Analizar la posibilidad que los factores situacionales puedan tener distinto grado de impacto en el desarrollo del proceso de requisitos, con lo cual podría ser necesario ponderar este impacto en las reglas de adaptación asignando pesos a los factores.
- Estudiar factores situacionales que puedan afectar la gestión de producción de requisitos, esto es, analizar el impacto sobre las actividades propiamente de gestión desde que los requisitos se originan hasta que son implementados en el sistema software.
- Estudiar factores situacionales que puedan afectar la gestión de configuración de requisitos, abarcando la administración de cambios, la administración de dependencias entre requisitos, la rastreabilidad de requisitos y el versionado de los mismos.
- Realizar otros estudios de caso aplicando las distintas variantes del proceso de IR, lo que daría un marco para definir qué tan generalizable a otros contextos es la adaptación propuesta.
- Evaluar la conveniencia de realizar reevaluaciones de los factores situacionales en puntos de variación internos de cada subproceso, en vez de evaluar en puntos de variación a nivel del proceso base, para lograr una reformulación más precisa del proceso.
- Dado el detalle con que se ha definido el método de adaptación del proceso de IR, es bien factible construir una herramienta de apoyo para obtener el proceso de IR ajustado a cada situación, permitiendo su reformulación.
- Probar que los mecanismos de adaptación propuestos son aplicables sobre otras estrategias de IR con ajustes menores o específicos a cada estrategia.

Anexo A

Análisis de Factores Situacionales en la Literatura

En este anexo se presenta el análisis realizado sobre 40 factores situacionales sugeridos en distintos trabajos relacionados al proceso de requisitos, se incluyen algunos factores que fueron propuestos por los autores de la estrategia de IR. Los primeros 23 factores están relacionados al contexto de aplicación y los 17 restantes al proyecto en sí mismo. En base a este análisis, se toma una de las siguientes decisiones sobre cada factor: aceptarlo tal cual se ha propuesto o con alguna variación, descartarlo, fusionarlo con otros similares, o separarlo en más de un factor para una evaluación más específica; a continuación se presenta un detalle de cada factor estudiado:

- 1 - **Nivel de experiencia del informante:** Los estudios presentados en (Burton, Shadbolt, Rugg & Hedgecock, 1990; Dhaliwal & Benbasat, 1990; Van de Weerd, Brinkkemper & Versendaal, 2010) tienden a medir cuan efectivas resultan ciertas técnicas de elicitación al aplicarlas con personas de diferentes niveles de pericia en un dominio dado o en el desempeño de su función. Informantes con poca experiencia podrían tener cierto grado de confusión o desconocimiento sobre sus necesidades, lo cual posiblemente provocaría frecuentes cambios en los requisitos o requisitos omitidos; estas características se analizan más adelante en los factores volatilidad de los requerimientos (ítem 15) y grado de confusión (ítem 16). Por otro lado, un informante con mucha experiencia podría considerar que parte de su conocimiento es obvio y no transmitirlo, esto podría vincularse con la capacidad de comunicación del informante (ítem 2). Este factor no sólo afecta en la actividad de elicitación sino en toda actividad del proceso donde se requiera la intervención de los involucrados. Sin embargo, no provoca adaptaciones en el proceso de requisitos en sí mismo, motivo por el cual es descartado.
- 2 - **Capacidad de comunicación del informante:** Este factor considera la facilidad del informante para explicar su conocimiento, es decir, con cuanta claridad puede transmitir sus ideas, se considera determinante para la selección de las

técnicas de elicitación (Dhaliwal & Benbasat, 1990; Carrizo Moreno, 2009). Como en el caso anterior, este aspecto incide al momento de valorar otros factores, por ejemplo, informantes con poca capacidad de comunicación podrían causar cambios frecuentes en la información elicitada lo que podría relacionarse con la volatilidad de los requerimientos (ítem 15). Este factor afecta toda actividad del proceso de requisitos donde participan involucrados, aun así, se considera que por sí mismo no tiene un impacto directo en la adaptación del proceso IR, motivo por el cual se lo descarta.

- 3 - **Variables de personalidad del informante:** El factor es estudiado por Dhaliwal & Benbasat (1990) desde la perspectiva de las características innatas de cada sujeto, donde se considera que estas podrían tener influencia en la eficacia de aplicar determinada técnica de elicitación. El principal problema que presenta este factor es la dificultad para su valoración. Este factor compromete la actividad de recolección de hechos (técnicas de elicitación a utilizar) y toda actividad en la que se requiera intervención de los involucrados. Sin embargo, para este trabajo en particular, no se identifica un impacto directo en la adaptación del proceso de IR, razón por la que se descarta.
- 4 - **Grado de resistencia del informante:** Jafarinezhad & Ramsin (2012) incluyen en su propuesta este factor que intenta medir la disposición del informante para colaborar durante el proceso de IR. Algunas veces pueden existir conflictos de intereses que afectan la actitud de las personas involucradas hacia el sistema a desarrollar. Así también, la flexibilidad y adaptabilidad de los usuarios serán algunos subfactores que pueden tener influencia para que el informante muestre cierta resistencia. Aspectos semejantes a estos son considerados al analizar el grado de conflicto en el dominio (ítem 16), por lo tanto, se analiza posteriormente en conjunto con aquel.
- 5 - **Problemas cognitivos del informante / Habilidades cognitivas del informante:** Los autores que mencionan este factor lo hacen desde la perspectiva de problemas en el proceso de IR que pueden originarse debido a: problemas propios del informante para elaborar una solución a determinado problema, dificultades entre dos personas por no utilizar un lenguaje en común, o problemas cuando intervienen varias personas para consensuar los requisitos

(Byrd, Cossick & Zmud, 1992; Chao & Salvendy, 1995). A partir del análisis se detecta que este factor al igual que el anterior involucra una serie de aspectos que son considerados en los factores grado de confusión (ítem16) y diferencias idiomáticas y/o culturales entre los involucrados (ítem 37). Impacta en toda actividad en la que participan involucrados, sin embargo, dado que el factor de modo aislado no es útil para la adaptación del proceso de IR, no se lo incluye en la lista final de factores seleccionados.

- 6 - **Localización del informante en relación con el equipo de IR/ Involucrados distribuidos geográficamente:** Este factor se refiere a la ubicación de los sujetos que actúan como fuente de información con respecto al equipo de ingenieros de requisitos (Carrizo Moreno, 2009) algunas técnicas de elicitación requieren la presencia del informante para su efectividad, mientras que otras serán más convenientes cuando los informantes se encuentran distribuidos con cierta lejanía. Desde un enfoque más general otros autores plantean la situación en la que los involucrados están distribuidos geográficamente (Ebling, Nicolas Audy & Prikladnicki, 2009; Mighetti & Hadad, 2015). En este último caso, si los informantes estuvieran distribuidos, sería conveniente construir o no el LEL del UdeD, los EA y el LEL del Sistema para que se comprendan y registren las diferentes variantes de trabajo en las regiones. Además, puede afectar la forma de elicitar y de validar los modelos de la estrategia. Por lo tanto, esta se considera suficiente justificación para incorporar este factor en la propuesta con la denominación *Localización de usuarios*.
- 7 - **Tipo de cliente/ Tipo de producto:** El desarrollo de un producto software se puede dividir en dos segmentos según a quien está dirigido: si se desarrollará o comprará a medida para un cliente específico, o si se trata del desarrollo de un paquete de software para un mercado de potenciales clientes donde no existe un cliente específico ni usuarios como principal fuente de información (Keil & Carmel, 1995; Hadad, 2008; Gorschek, 2011). En relación a la estrategia de IR, cuando el producto está dirigido a un cliente específico la realización de cada fase del proceso dependerá de los restantes factores, en cambio, si el producto está concebido para un mercado potencial la forma de llevar a cabo el proceso haría presuponer la no construcción de los EA dado

que no hay situaciones actuales para una organización específica. Por lo tanto, existe suficiente justificación para aceptar el factor *Tipo de cliente*.

- 8 - **Nivel de rotación de usuarios:** Se entiende por rotación el ingreso y el egreso de personas en la organización, que puede deberse a distintos factores, e incluye tanto la desvinculación, como la reubicación dentro de la misma organización, afectándolas como fuentes de información (Hadad, 2008). Por ser el proceso de IR el que mayor interacción tiene con los usuarios, todo cambio en el grupo de usuarios que actúan como fuente de información tendrá su impacto en dicho proceso. Por ejemplo, una alta rotación de los usuarios haría recomendable la construcción de los EA, a fin de evitar malentendidos con los nuevos usuarios que podrían entorpecer el avance normal del proyecto. A la vez, los EA podrían servir como instrumentos de capacitación para los nuevos miembros que ingresen a la organización. Además, afecta indirectamente la negociación de propuestas mediante EF, que el proceso ya lo maneja apropiadamente sin requerir adaptaciones. Por lo anterior, se decide aceptar el factor con el nombre *Rotación de usuarios*.
- 9 - **Participación del usuario/ Disponibilidad de tiempo del informante:** La participación de aquellos que tienen la información relevante para los requisitos depende, entre otras cosas, del tiempo que tengan disponible (Lloyd, 2001; Carrizo, Dieste & Juristo, 2008; Jafarinezhad & Ramsin, 2012), ya sea para proveer información como también brindar las devoluciones necesarias. En particular, el atributo está asociado a las técnicas de elicitación, aunque afecta a toda actividad que requiera intervención de los involucrados. Sin embargo, no provoca adaptaciones en la estrategia de IR, por lo tanto, no se considera pertinente incluirlo en la lista final de factores.
- 10 - **Tipo de fenómeno:** Este factor es propuesto por Maiden & Rug (1996) quienes hacen una distinción entre los hechos que ocurren en el UdeD: algunos pueden ser comunicados por los stakeholders, mientras que otros solo pueden ser percibidos visualmente. Consideran que ciertas técnicas de elicitación, por su propia naturaleza, son mejores que otras para capturar hechos que solo son observables. Aunque el factor afecta el proceso de IR

este lo maneja apropiadamente sin necesidad de adaptaciones, motivo por el cual es excluido.

- 11 - **Tipo de información:** En el marco de selección de técnicas de Maiden & Rug (1996) se clasifica el tipo de información a elicitar en: comportamiento, procesos y datos. Consideran que los dos primeros tipos se pueden obtener de modo observable o comunicados, mientras que si se trata de datos se obtendrían de otros modos, por ejemplo, aplicando técnicas de card sorting o laddering. Se detecta una gran similitud con el factor tipo de fenómeno (ítem10), no causa adaptaciones en el proceso de IR, motivo por el cual también se desestima.
- 12 - **Campos de dominio / Dominio de la aplicación:** Los trabajos que proponen este atributo se refieren al campo de estudio, disciplina o área de negocio a la que se circunscribe el sistema software a desarrollar (Shadbolt & Burton, 1989; Jafarinezhad & Ramsin, 2012), en relación con la efectividad de determinadas técnicas de elicitación. Definir valores posibles para este atributo podría volverse inmanejable debido a la cantidad de dominios existentes. Por otra parte, no se considera posible relacionar la conveniencia de la realización de ciertas actividades o productos del proceso de IR para ciertos dominios, ya que esto dependerá de muchos otros factores. Por lo anterior, se decide eliminar este atributo del conjunto de factores.
- 13 - **Grado de estructura percibida del dominio / Complejidad del contexto:** Algunos autores mencionan el grado de la estructura percibida para referirse a nivel de incertidumbre sobre la complejidad del UdeD, entre los aspectos considerados están el grado de riesgo, la cantidad de alternativas para realizar una tarea, el tipo de relaciones entre sus elementos (Kim & Courtney, 1988; Dhaliwal & Benbasat, 1990). De manera similar, otros se refieren a la complejidad del contexto (Jafarinezhad & Ramsin, 2012), el cual implica el grado de dificultad para imaginar el futuro haciendo una abstracción mental de los problemas actuales o mejoras deseadas respecto a la realidad existente. El principal impacto de este factor en la estrategia de IR se relaciona con la etapa de creación de los EA, pues a mayor complejidad en las actividades que se desarrollan en dicho contexto, es casi inevitable construirlos para

poder comprender el contexto. Por lo tanto, esto justifica mantener este factor con el nombre *Complejidad del contexto*.

- 14 - **Tipo de tareas a elicitar:** Varios autores (Dhaliwal & Benbasat, 1990; Chao & Salvendy, 1995; Wagner, Chung & Najdawi, 2003) consideran que ciertas técnicas de adquisición de conocimiento pueden resultar más efectivas que otras dependiendo de la naturaleza de la tarea que se deba elicitar, si es de predicción, de diagnóstico, de supervisión, de interpretación, de control, entre otras. Una de las complicaciones que presenta este atributo es la dificultad para definir valores con los cuales categorizar el universo de tipos de tareas existentes, y además, que cada tarea a elicitar se pueda relacionar a una única categoría. Por otra parte, más allá de la actividad de selección de las técnicas de elicitación, el proceso de IR no requiere adaptación al respecto, pues dicha actividad es parte intrínseca de los bloques de proceso que involucran elicitación, con lo cual el factor no es incluido en la propuesta.
- 15 - **Volatilidad de los requerimientos / Dominio con cambio constante:** Jafarinezhad & Ramsin (2012) se refieren a los cambios que puedan surgir a lo largo del ciclo de vida de un proyecto de desarrollo de software con respecto a los requisitos definidos inicialmente. Estos pueden originarse por el bajo nivel de madurez del cliente, es decir, cambios continuos de idea respecto a sus necesidades o, tal como menciona Hadad (2008), debido a cambios en el propio contexto de aplicación, ya sea por causas externas o internas al mismo. Alta volatilidad implica cambios continuos en la información elicitada y modelada. En cuanto a su incidencia en el proceso de IR, un UdeD que sufre constantes cambios, puede afectar la definición del objetivo del sistema. También se deberá evaluar conforme a los restantes factores la conveniencia o no de construir y mantener los EA. Inclusive, si la volatilidad de los requerimientos es muy alta podría indicar la conveniencia de adoptar otra metodología, por ejemplo, una metodología ágil. Si bien en un inicio, se han agrupado ambos factores, luego del análisis se considera apropiado desglosarlo en dos bien diferenciados, *Volatilidad del contexto* y *Volatilidad de los requerimientos del cliente*.

- 16 - **Grado de conflicto en el dominio / Grado de confusión:** Ambas denominaciones se refieren al nivel de ambigüedad y de contradicciones en la información elicitada de diversas fuentes de información (Fazlollahi & Tanniru, 1991; Jafarinezhad & Ramsin, 2012). Podría ocurrir que exista un alto grado de contradicciones por parte de las fuentes de información en cuanto a cómo realizan tareas, con lo cual se vuelve imprescindible la construcción de los EA para detectar las inconsistencias y poder establecer la realidad. Otra situación frecuente es que los involucrados no tienen claras sus necesidades o la solución requerida, incluso pueden surgir desacuerdos o diferencias de opiniones e intereses entre ellos (ver ítem 4). Esto puede afectar a la estrategia de IR, pero en relación a la creación de EF, pues pueden surgir diferentes alternativas por distintas maneras de querer trabajar. Además, puede influir en la priorización de los requisitos o de los EF si no se explicitan los requisitos. Con el fin de diferenciar bien estas situaciones, se incorporan dos factores: *Inconsistencias en el contexto* y *Conflicto de intereses de usuarios*.
- 17 - **Grado de reingeniería del proceso de negocio:** Este factor se refiere al impacto que provoca la implantación de un nuevo sistema en la organización cliente o en su contexto de aplicación, aspectos similares a estos son considerados en los ítems 19 y 22. Puede ocurrir que procesos actuales del negocio sufran modificaciones, algunos pueden quedar en desuso mientras que otros pueden ser totalmente redefinidos o nuevos en función del sistema a construir (Hadad, 2008). Con relación a la estrategia de IR, este factor es determinante para el modo en que se van a construir los EF, esto da suficiente justificación para mantener el factor con la denominación *Reingeniería en el proceso de negocio*.
- 18 - **Novedad del negocio:** Indica si se trata de un contexto de aplicación nuevo o de un negocio ya existente (Hadad, 2008). Es determinante para la decisión de construir o no los EA, por ejemplo, si un negocio es nuevo los EA no tendrían sentido, por lo cual se considera que el mismo tiene suficiente relevancia para mantenerlo. Se lo renombra *Novedad del contexto*.

- 19 - **Importancia estratégica:** Algunos aspectos como la prioridad del proyecto o el efecto que ha de tener la implantación del sistema software en relación con los objetivos estratégicos del negocio son subfactores que determinan este factor (Jafarinezhad & Ramsin, 2012). Cuestiones similares se han analizado junto al factor grado de reingeniería del proceso de negocio (ítem 17), por lo cual se lo unifica con aquel.
- 20 - **Número de usuarios:** Maiden & Rug (1996) mencionan la cantidad de usuarios que participan en el proceso de elicitación, como factor a considerar para la selección de técnicas de elicitación. De modo similar, Carrizo Moreno (2009) con un poco más de detalle se refiere a la cantidad de individuos por sesión. Se trata de un factor que se evalúa exclusivamente para determinar las técnicas a utilizar en la elicitación, el proceso no requiere adaptación al respecto pues es parte intrínseca de los bloques de proceso que involucran elicitación, por tal motivo no se considera pertinente para este trabajo.
- 21 - **Compromiso de la dirección de la organización del cliente:** Jafarinezhad & Ramsin (2012) incluyen este atributo que se evalúa en función al nivel de apoyo que brinda la dirección, su disponibilidad y su grado de involucramiento en la gerencia del proyecto. Puede afectar la definición del objetivo y la definición de la solución en términos de cuáles serán los EF apropiados. También puede afectar la visión del UdeD desde la definición del vocabulario en el LEL del UdeD. Sin embargo, no impacta en la estrategia de IR como para requerir una adaptación específica del proceso, por lo tanto, este factor se desestima.
- 22 - **Impacto organizacional del proyecto:** Jafarinezhad & Ramsin, (2012) incluyen este factor que mide las consecuencias de la implantación del sistema para los involucrados y para los objetivos de la organización. Aspectos semejantes a estos se consideraron al analizar el grado de reingeniería del proceso de negocio (ítem 17), por lo cual se decide unirlo con dicho factor.
- 23 - **Grado de definición del problema:** Carrizo Moreno (2009) se refiere al conocimiento que tienen los demandantes sobre las características generales del problema y su dominio, si existe información contradictoria o ambigua, o una definición clara del problema o necesidad. Aunque se aplica para la

selección de las técnicas de elicitación más convenientes, puede afectar principalmente la definición del objetivo del sistema. No obstante, este factor involucra en parte *Inconsistencias en el contexto* (ítem 4), la *Volatilidad del contexto* (ítem 15), *Volatilidad de los requerimientos del cliente* (ítem 15) y el *Conflicto de intereses* (ítem 16), por lo cual se desglosa en los factores mencionados.

- 24 - **Tipo de proyecto:** Este atributo categoriza los proyectos de acuerdo con la naturaleza del software. Algunos ejemplos que dan los autores son: sistema de información, de tiempo real, de seguridad crítica, web, entre otros (Jafarinezhad & Ramsin, 2012). La valoración del factor no es sencilla, ya que habría que asegurar que los tipos definidos puedan servir para clasificar cualquier proyecto con el que se trabaje y además que cada uno de estos pueda relacionarse a una sola categoría. El tipo de proyecto podría incidir en las exigencias de calidad y la rigurosidad con la que se debe realizar el proceso de IR. Sin embargo, este es un aspecto que se toma en cuenta al evaluar el nivel de criticidad (ítem 40), que se analiza posteriormente. Este factor por sí mismo no impacta en el proceso de IR, por tal motivo se decide descartarlo.
- 25 - **Envergadura del proyecto:** La envergadura o tamaño del proyecto se puede definir en términos de la cantidad de miembros del equipo de IR, el cronograma, los recursos asignados, y el tamaño estimado del software (Jafarinezhad & Ramsin, 2012). Hadad (2008) destaca la evaluación de esta característica al momento de definir las etapas de la estrategia de IR que se deben realizar. Por ejemplo, en un proyecto de gran envergadura se vuelve imprescindible construir los EA, estos se ponen a disposición del equipo como fuente de conocimiento para evitar sobrecargar de información la memoria de los ingenieros y, por otro lado, se facilita la rastreabilidad de los requisitos a sus orígenes, algo muy necesario en proyectos grandes. Por lo tanto, el factor *Envergadura del proyecto* se incluye para la propuesta.
- 26 - **Grado de reusabilidad de los artefactos:** Jafarinezhad & Ramsin (2012) destacan la importancia de utilizar en el proyecto artefactos de software generados en proyectos de desarrollo anteriores con algunas características

similares. Esto aplica también al proceso de IR en el que se podrían aprovechar modelos, documentos y especificaciones de requisitos. Inversamente, podría darse la exigencia de obtener productos del proceso de requisitos para reusar en otros proyectos de software. Por lo tanto, se considera apropiado generar dos factores que diferencien bien estas situaciones, denominados *Reuso de artefactos de requisitos existentes* y *Creación de artefactos de requisitos para reuso*.

- 27 - **Nivel de madurez de los procesos de IR:** Jafarinezhad & Ramsin (2012) mencionan este factor que se relaciona con el grado de estandarización de los procesos, incluye el mejoramiento de los procesos, el uso de metodologías y herramientas, entre otros. En el caso particular de la estrategia de IR, se podría decir que el proceso está bien definido y estandarizado. Sin embargo, el conocimiento y la experiencia del equipo de IR en su aplicación favorece la obtención de un resultado exitoso. Este aspecto es analizado en el ítem 34, por lo tanto, este atributo como tal es descartado.
- 28 - **Restricciones al proyecto:** Maiden & Rug (1996) analizan esta característica para recomendar técnicas de elicitación. Las restricciones en el contexto de aplicación incluyen: cantidad de reuniones necesarias, tiempo de preparación de las sesiones y de análisis y ciertas restricciones técnicas, entre otras. El problema con estos aspectos es que pueden resultar dificultosos de valorar. Aunque este factor no permanece tal cual fue planteado por los autores, algunas de sus características se consideran durante el análisis del factor escasez de personal y recursos (ítem 31), por lo que se fusiona con aquel factor.
- 29 - **Estándares de especificación exigidos:** Hadad (2008) es quien menciona la posibilidad de usar estándares de especificación como parte del contrato. Puntualmente hace referencia a la construcción de la ERS. Se considera importante su inclusión en la propuesta, y para mayor claridad se cambia su denominación a *Exigencia de producir el documento ERS*.
- 30 - **Nivel de rotación del grupo desarrollador:** Anteriormente se analizó la importancia de la rotación de los usuarios debido a la fluctuación de las personas en las organizaciones, del mismo modo ocurre en los equipos de

desarrollo de software, donde pueden ingresar nuevos recursos, algunos irse, o incluso pueden ser reasignados a otros proyectos (Hadad, 2008). Si hay alta rotación del personal sería muy beneficioso disponer de los EA para el entrenamiento de los nuevos integrantes del equipo en los procesos de negocio actuales. Por lo tanto, hay justificación para incluir el factor denominado *Rotación del equipo desarrollador*.

- 31 - **Escasez de personal y recursos:** Jafarinezhad & Ramsin (2012) proponen este atributo para hacer referencia a limitaciones en los plazos de entrega, así como en la disponibilidad de recursos humanos y de presupuesto, algo que no se puede obviar al momento de llevar adelante cualquier proceso de desarrollo (en el ítem 28 se incluyen además las restricciones técnicas). Un nivel alto de restricciones de tiempo o recursos podría resultar incompatible con otros factores, por ejemplo, si hay alta volatilidad de requerimientos del cliente o alta rotación de los usuarios. Esto justifica incluir este factor en la propuesta, aunque se restringe su enfoque a restricciones temporales o de recursos humanos, la denominación que se le da es *Restricciones de tiempo y recursos para el proyecto*. Se debe tener presente que, a modo particular, este factor por sí mismo no provoca cambios en la estrategia de IR, sin embargo, cuando existan altas restricciones, se deberá prestar atención a la valoración de determinados factores, dado que en algunos casos podrá requerir algún tipo de negociación.
- 32 - **Momento del proceso:** Carrizo Moreno (2009) se refiere a la etapa cronológica en que se encuentra el proyecto, entonces dependiendo de si se trata de una fase inicial o si el proyecto está avanzado propone ciertas técnicas de elicitación más convenientes. Fuera de la etapa de elicitación no se le encuentra aplicación para la adaptación de la estrategia de IR en general, pues ya es tratado como parte del proceso, por este motivo este atributo no se considera pertinente y es descartado.
- 33 - **Método de desarrollo de software:** Dhaliwal & Benbasat (1990) son quienes analizan la conveniencia de ciertas técnicas de elicitación y de las herramientas más adecuadas dependiendo del enfoque adoptado para el desarrollo del sistema. En un sentido más amplio, cuando se analiza el factor

en función al proceso de IR en general, se entiende que incide directamente en la forma en que este debe ser realizado, por ejemplo, hay variaciones en la forma de ejecutar el proceso si se sigue un modelo en cascada, incremental o iterativo, incluso puede pasar que en otros modelos de proceso no sea posible aplicar un proceso de IR. Por lo tanto, se considera justificada la reorientación del factor en tal sentido, con la denominación *Modelo de proceso de software*.

- 34 - **Experiencia en IR / Conocimiento en técnicas de educación / Experiencia en métodos de educación:** La experiencia en las actividades de IR ha sido valorada por varios autores (Agarwal & Tanniru, 1990; Dhaliwal & Benbasat, 1990; Lloyd, Rosson & Arthur, 2002; Jafarinezhad & Ramsin, 2012) en base a cuantificar la participación del equipo de IR en las actividades de requisitos o el tiempo que se lleva realizando las mismas. Para aplicar eficazmente determinadas técnicas puede ser requerida cierta experiencia previa. Lo mismo aplica a las actividades de la estrategia de IR, por ejemplo, en el caso de ingenieros de requisitos con poca experiencia en IR, aun cuando estén trabajando con un dominio conocido, podrían tener dificultad para escribir directamente los EF sin haber construido los EA. También afecta las técnicas a utilizar en las distintas actividades del proceso. Por tal motivo se incluye el factor en la propuesta con el nombre *Experiencia en el proceso de IR*.
- 35 - **Familiaridad con el dominio / Grado de conocimiento de los requisitos / Conocimiento previo del dominio:** Los trabajos que analizan estos atributos (Hadad, 2008; Jafarinezhad & Ramsin, 2012) destacan la importancia para el proceso de IR del nivel de conocimiento y experiencia del equipo de requisitos sobre el dominio del problema (aspectos similares se tratan en el ítem 39). Por su parte, Jafarinezhad & Ramsin (2012) añaden el factor grado de conocimiento de los requisitos, refiriéndose al nivel de disponibilidad de las especificaciones de requisitos, la precisión de las descripciones de los procesos de negocio y cuán fácil resulta la actividad de elicitación. Lo cierto es que, si se debe abordar un proyecto relacionado a un dominio poco familiar o desconocido, se requerirá más rigurosidad al aplicar el proceso, en cambio con un dominio altamente conocido por el equipo, dependiendo de otros factores podría omitirse, por ejemplo, la construcción del LEL del UdeD, de los

EA y del LEL del Sistema. Con lo cual, se considera que existen motivos suficientes para mantener este factor con la denominación *Conocimiento previo del dominio*.

36 - **Diferencias temporales:** Cuando se trabaja en entornos donde los involucrados están distribuidos geográficamente la comunicación cara a cara e informal es reemplazada por el uso de medios electrónicos, incluso de forma asincrónica si existen diferencias temporales mayores, lo cual puede causar posibles desentendidos, retrabajo y ambigüedades en los artefactos generados (Ebling, Nicolas Audy & Prikladnicki, 2009; Mighetti & Hadad, 2015). Esto afecta principalmente a aquellas actividades que requieren comunicación constante entre los involucrados. En cuanto al proceso de IR, a mayor diferencia temporal en el equipo de desarrollo es más conveniente trabajar sobre documentos que hacer preguntas y obtener respuestas puntuales, construir el LEL del UdeD, los EA y el LEL del Sistema permite resolver dudas o tomar decisiones individualmente aprovechando los escasos momentos de comunicación para consultas más profundas y de difícil comprensión. Por otra parte, si hay diferencias temporales de los usuarios, deberá evaluarse la conveniencia de aplicar técnicas de validación que no requieran comunicación sincrónica entre los involucrados. Lo anterior justifica la necesidad de incluir el factor, denominándolo *Diferencias temporales entre los involucrados*.

37 - **Diferencias idiomáticas y/o culturales entre los involucrados:** Los significados de algunos términos específicos del contexto pueden ser diferentes dependiendo del idioma, incluso de la cultura, impactando directamente en la calidad de requisitos (Ebling, Nicolas Audy & Prikladnicki, 2009; Mighetti & Hadad, 2015). En el caso de la estrategia de IR, afecta en especial la creación del LEL del UdeD, de los EA y del LEL del Sistema, a mayor divergencia culturales y/o idiomáticas entre los involucrados, estos modelos deberían ser bien detallados, en un idioma a compartir. Además, para ambos modelos se requeriría aplicar técnicas de verificación y validación de alta efectividad. Lo anterior da suficiente justificación para mantener el factor con la denominación *Diferencias idiomáticas y/o culturales*.

- 38 - **Nivel de innovación esperada del proyecto y del proceso:** Este factor se refiere a la innovación requerida en el proyecto y en el proceso de desarrollo, incluyendo las necesidades de equipamiento y herramientas (Jafarinezhad & Ramsin, 2012). Los autores que lo proponen no dan una idea precisa de la manera en que puede afectar al proceso de IR. No se han encontrado razones para tomarlo en consideración en la propuesta.
- 39 - **Nivel de información disponible antes de aplicar la técnica de elicitación:** Este factor se refiere al tipo de información con que se cuenta antes de aplicar alguna técnica de adquisición de conocimiento. Es posible que no se tenga ninguna información, solo algunos conceptos generales, o en algunos casos se parta de bastante información sobre los procesos, funciones y otras características del contexto de aplicación (Carrizo Moreno, 2009). Este autor propone ciertas técnicas de elicitación según la información inicial, que pueden resultar más efectivas que otras. Los aspectos considerados en este factor son similares a los analizados en el factor familiaridad o conocimiento previo del dominio (ítem 35) analizado previamente, por tal motivo se lo unifica con aquel.
- 40 - **Nivel de criticidad:** Jafarinezhad & Ramsin (2012) se refieren a la criticidad en términos del impacto de una falla en el sistema, si hay peligros para el ambiente, pérdida de vidas humanas, de recursos financieros, equipamientos, y demás. Consideran que un mayor nivel de criticidad puede requerir aumentar el formalismo en el proceso. Siendo que la mayor parte de las actividades de IR se realizan al inicio del proceso de desarrollo de software, evaluar la confiabilidad esperada del software podría transmitir mejor la idea que pensar en las consecuencias de posibles fallas. En el caso de la estrategia de IR, cuanto más alta sea la exigencia de confiabilidad, por la misma naturaleza del software a analizar, podría implicar un mayor formalismo o rigurosidad en el proceso. Por lo tanto, se considera justificado adoptar este factor. Para aportar mayor claridad al concepto se lo denomina *Fiabilidad necesaria del software*.

Anexo B

Formulario para Evaluar Factores Situacionales

FACTORES SITUACIONALES

Proyecto: _____ Fecha: ___/___/___

Responsables: _____

FACTORES DEL CONTEXTO		NIVEL DE CONFIANZA		
		Seguro	Razonablemente Seguro	Dudoso
1.	Complejidad del contexto Muy Alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2.	Tipo de cliente Mercado Potencial <input type="checkbox"/> Cliente Específico <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3.	Novedad del contexto Sí <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4.	Reingeniería en el proceso de negocio (Completar sólo si Novedad del contexto es "No" y Tipo de cliente es "Cliente Específico") Muy Alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5.	Volatilidad del contexto Muy Alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6.	Volatilidad de los requerimientos del cliente (Completar sólo si Tipo de cliente es "Cliente Específico") Muy alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7.	Inconsistencias en el contexto (Completar sólo si Novedad del contexto es "No") Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
8.	Conflicto de intereses de usuarios (Completar sólo si Novedad del contexto es "No" y Tipo de cliente es "Cliente Específico") Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
9.	Rotación de usuarios (Completar sólo si Novedad del contexto es "No" y Tipo de cliente es "Cliente Específico") Muy Alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
10.	Localización de usuarios (Completar sólo si Novedad del contexto es "No" y Tipo de cliente es "Cliente Específico") Distribuidos <input type="checkbox"/> Colocalizados <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FACTORES DEL PROYECTO		NIVEL DE CONFIANZA		
		Seguro	Razonablemente Seguro	Dudoso
11.	Conocimiento previo del dominio Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
12.	Envergadura del proyecto Muy Grande <input type="checkbox"/> Grande <input type="checkbox"/> Mediana <input type="checkbox"/> Pequeña <input type="checkbox"/> Muy Pequeña <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
13.	Rotación del equipo desarrollador Muy Alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
14.	Fiabilidad necesaria del software Muy Alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Baja <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15.	Reuso de artefactos de requisitos existentes LEL <input type="checkbox"/> EA <input type="checkbox"/> EF <input type="checkbox"/> LEL-S <input type="checkbox"/> Requisitos <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16.	Creación de artefactos de requisitos para reuso LEL <input type="checkbox"/> EA <input type="checkbox"/> LEL-S <input type="checkbox"/> Requisitos <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17.	Pre-Rastreabilidad de los requisitos Total <input type="checkbox"/> Parcial <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18.	Post-Rastreabilidad de los requisitos Sí <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19.	Granularidad de rastreabilidad de los requisitos (Completar sólo si Pre-Rastreabilidad de los requisitos es "Total" o "Parcial" o Post-Rastreabilidad de los requisitos es "Sí") Individual <input type="checkbox"/> Grupal <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20.	Exigencia de producir el documento ERS (No puede ser "Sí" si Modelo de proceso de software es "Incremental" o "Iterativo") Sí <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
21.	Restricciones de tiempo y recursos para el proyecto Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

22.	Modelo de proceso de software (No puede ser "Cascada" ni "Transformación Formal" si Volatilidad de los requerimientos del cliente es "Muy Alta" o "Alta", o Volatilidad del contexto es "Muy Alta" o "Alta") Cascada <input type="checkbox"/> Incremental <input type="checkbox"/> Iterativo <input type="checkbox"/> Transf. Formal <input type="checkbox"/> Otro <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
23.	Tipo de incremento (Completar sólo si Modelo de proceso de software es "Incremental") Nueva Funcionalidad <input type="checkbox"/> Mejora a Funcionalidad <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
24.	Experiencia en el proceso de requisitos Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
25.	Diferencias idiomáticas y/o culturales Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
26.	Diferencias temporales entre los involucrados Muy Alto <input type="checkbox"/> Alto <input type="checkbox"/> Medio <input type="checkbox"/> Bajo <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Aclaraciones para completar el formulario:

- Complejidad del contexto:** Nivel de dificultad en comprender el problema bajo estudio por parte de los ingenieros de requisitos. El nivel de complejidad puede establecerse en base a la cantidad de procesos, cantidad de áreas intervinientes en la organización, interdependencia no evidente de áreas, y de actividades entre áreas, roles y responsabilidades no precisas, entre otros.
- Tipo de cliente:** Destinatario del software, si se desarrollará o comprará a medida para un cliente específico, o si se trata de un paquete de software para un mercado de potenciales clientes.
- Novedad del contexto:** Indica si se trata de un contexto de aplicación nuevo o ya existente. En otros términos, si el negocio ya existe o se va a crear a partir del software a construir.
Sí: cuando el sistema será parte de un nuevo contexto
No: cuando el sistema se utilizará en un contexto que existe previamente
- Reingeniería en el proceso de negocio:** Nivel de cambios esperados en los procesos del negocio a través del nuevo software, es decir, el impacto en la organización cliente o en el contexto de aplicación del sistema a construir.
Muy Alta: cuando habrá procesos del negocio totalmente redefinidos o nuevos en función del software a construir
Medía: por ejemplo, cuando se informatizará lo manual y cambiarán algunos procesos
Muy Baja: por ejemplo, cuando solo se informatizará lo manual
- Volatilidad del contexto:** Nivel de cambios que ocurren en el contexto de aplicación, ya sea por causas externas o internas al contexto. Alta o Muy Alta volatilidad implica cambios continuos en la información elicitada y modelada.
- Volatilidad de los requerimientos del cliente:** Nivel de cambios solicitados por el cliente en cuanto a las necesidades a cubrir con el software. Alta o Muy Alta volatilidad indica bajo nivel de madurez del cliente, es decir, cambios continuos de idea respecto a sus necesidades con poca aceptación de propuestas de los ingenieros de requisitos.
- Inconsistencias en el contexto:** Nivel de contradicciones en la información elicitada proveniente de diversas fuentes de información.
- Conflicto de intereses de usuarios:** Nivel de desacuerdo o diferencias de opiniones e intereses entre los involucrados.
- Rotación de usuarios:** Nivel de cambio en los usuarios que actúan como fuente de información. Esto debe ser informado por el cliente.
- Localización de usuarios:** Distribución geográfica de los usuarios, si están dispersos o en un único lugar.
Distribuidos: usuarios en distintas regiones
Colocalizados: usuarios en un mismo lugar físico
- Conocimiento previo del dominio:** Familiaridad del equipo de requisitos con el dominio al cual pertenece el software a construir. Se considera la experiencia promedio del equipo de ingenieros de requisitos en participar en proyectos referidos a dicho dominio o en tener formación en el dominio. El conocimiento sobre el dominio puede ir desde niveles generales abstractos a niveles de detalle.
Muy Alto: el equipo tiene en promedio vasta experiencia en las actividades del contexto de aplicación, cómo se desenvuelven y las responsabilidades de sus actores en dicho contexto particular
Alto: el equipo tiene en promedio vasta experiencia en el dominio, pero no en el contexto de aplicación
Medio: el equipo tiene en promedio cierta experiencia en el dominio
Bajo: el equipo tiene en promedio algún conocimiento general del dominio, por tener formación formal en el mismo o poca experiencia en el dominio
Muy Bajo: el equipo tiene en promedio solo alguna formación informal en el dominio
- Envergadura del proyecto:** Tamaño del proyecto, puede estimarse por la cantidad de miembros del equipo de ingenieros de requisitos y la duración estimada del proyecto.
Muy Grande: el proyecto requiere un equipo de más de 15 personas con una planificación superior a 4 años
Grande: el proyecto requiere un equipo de más de 10 personas con una planificación superior a 2 años
Mediana: el proyecto requiere de un equipo de hasta 6 personas, con un cronograma planificado que no exceda un año
Pequeña: el proyecto requiere de un equipo de no más de 3 personas, con un cronograma planificado que no exceda un año
Muy Pequeña: el proyecto requiere de un equipo de no más de 3 personas, con un cronograma planificado que no exceda los 3 meses
- Rotación del equipo desarrollador:** Nivel de cambio en los miembros del equipo de desarrollo.
- Fiabilidad necesaria del software:** Nivel de fiabilidad necesaria en los artefactos producidos en el proceso de software, en este caso particular en el proceso de requisitos. Por ejemplo, se requiere un muy alto nivel de fiabilidad en caso de un software de tiempo real, de seguridad crítica o que frente a una falla pueda derivar en peligros ambientales, daño o pérdida de vidas humanas, de recursos financieros o de equipamientos.

- 15. Reuso de artefactos de requisitos existentes:** Posibilidad de reusar en este proyecto modelos, documentos o especificaciones de requisitos, generadas en proyectos anteriores de desarrollo de software de similares características. Se debe indicar qué artefactos se pueden utilizar o ninguno.
- LEL:* cuando hay posibilidad de reutilizar un LEL del UdeD
EA: cuando hay posibilidad de reutilizar Escenarios Actuales
EF: cuando hay posibilidad de reutilizar Escenarios Futuros
LEL-S: cuando hay posibilidad de reutilizar un LEL del Sistema
Requisitos: cuando hay posibilidad de reutilizar especificaciones de requisitos
No: no existen artefactos para reutilizar
- 16. Creación de artefactos de requisitos para reuso:** Exigencia de obtener productos reusables del proceso de requisitos para otros proyectos de software. Se debe indicar qué artefactos de reuso se exige crear o ninguno.
- LEL:* cuando se exige crear el LEL del UdeD para reutilizar
EA: cuando se exige crear los Escenarios Actuales para reutilizar
LEL-S: cuando se exige crear un LEL del Sistema para reutilizar
Requisitos: cuando se exige crear especificaciones de requisitos para reutilizar
No: no se exige la creación de artefactos para reutilizar
- 17. Pre-Rastreabilidad de los requisitos:** Exigencia de mantener rastreabilidad entre los requisitos y los artefactos previos al establecimiento de los requisitos, incluyendo las fuentes de información.
- Total:* se debe mantener rastreabilidad entre los requisitos y cada uno de los artefactos propuestos por la estrategia de IR (LEL del UdeD, EA, EF, LEL del Sistema)
Parcial: se debe mantener rastreabilidad entre los requisitos y los artefactos creados previamente, solo con aquellos creados a partir de la adaptación de la estrategia de IR
No: no hay exigencia de mantener rastreabilidad entre los requisitos y los artefactos previos
- 18. Post-Rastreabilidad de los requisitos:** Exigencia de mantener rastreabilidad entre los requisitos y los artefactos posteriores a especificar los requisitos.
- 19. Granularidad de rastreabilidad de los requisitos:** Nivel de detalle en la rastreabilidad de los requisitos. Se considera Individual cuando se necesita rastrear individualmente cada requisito, y Grupal cuando alcanza con rastrear conjuntos de requisitos, por ejemplo, rastrear por escenarios.
- Individual:* cuando se necesita rastrear individualmente cada requisito
Grupal: cuando alcanza con rastrear conjuntos de requisitos, por ejemplo, rastrear por escenarios
- 20. Exigencia de producir el documento ERS:** Si se exige la construcción de un documento de Especificación de Requisitos del Software, por ejemplo, disponer del documento como parte del contrato entre las partes.
- 21. Restricciones de tiempo y recursos para el proyecto:** Limitaciones en los plazos de entrega, así como en la disponibilidad de recursos humanos para el proceso de desarrollo.
- 22. Modelo de proceso de software:** Uso de determinado proceso de software que impacta en la forma en que se debe entregar la documentación de requisitos.
- Cascada:* cada fase del proceso comienza cuando se ha finalizado la anterior, por lo cual, este modelo involucra la entrega total de la documentación de requisitos
Incremental: el modelo envuelve sucesivos incrementos, en cada uno se agrega una nueva funcionalidad o se mejora una funcionalidad existente, con lo cual, según se haya planificado, se hace una parte detallada y completa del proceso de requisitos, correspondiente a cada uno de los incrementos
Iterativo: el modelo consiste en realizar iteraciones que van refinando el sistema hasta lograr su versión final. En cuanto al proceso de requisitos, en cada iteración se refina cada modelo ya construido, aportando mayor detalle en diversas partes o con partes nuevas, que no son funcionalidades completas
Transf. Formal: este modelo aplica una serie de transformaciones para convertir una especificación formal en un sistema ejecutable. En cuanto al proceso de requisitos, el modelo implica la entrega total de la documentación de requisitos
Otro: se refiere a modelos de procesos en los que no se aplica el proceso de requisitos, por ejemplo, modelos de prototipado o métodos ágiles
- 23. Tipo de incremento:** Se relaciona al modelo de proceso de software incremental, indica si se trata de una nueva funcionalidad o de una mejora a una funcionalidad existente.
- Nueva Funcionalidad:* en el incremento se agrega a cada modelo ya construido otra parte detallada y completa
Mejora a Funcionalidad: en el incremento se mejora una parte de los modelos ya construidos
- 24. Experiencia en el proceso de requisitos:** Familiaridad del equipo de requisitos con el proceso de requisitos y las técnicas a aplicar. Se mide en años de experiencia promedio del equipo de ingenieros de requisitos, en función de la experiencia individual de cada integrante.
- Muy Alto:* más de 4 años promedio de experiencia
Alto: entre 3 y 4 años promedio de experiencia
Medio: entre 2 y 3 años promedio de experiencia
Bajo: entre 1 y 2 años promedio de experiencia
Muy Bajo: menos de 1 año promedio de experiencia
- 25. Diferencias idiomáticas y/o culturales:** Grado de divergencia entre los involucrados en el proyecto de software respecto al idioma que utilizan en sus ámbitos de trabajo y/o a sus culturas.
- Muy Alto:* los involucrados pertenecen a culturas muy diferentes con idiomas variados. Ej. Argentina-India
Alto: los involucrados usan diferentes idiomas con ciertas diferencias culturales. Ej. Argentina-Brasil
Medio: los involucrados usan el mismo idioma con ciertas diferencias idiomáticas y culturales regionales. Ej. Argentina-Perú
Bajo: los involucrados usan el mismo idioma con alguna diferencias idiomáticas y culturales locales. Ej. Buenos Aires-Uruguay, Buenos Aires-La Rioja
Muy Bajo: los involucrados usan habitualmente el mismo idioma y pertenecen a la misma cultura o muy similar

26. **Diferencias temporales entre los involucrados:** Grado de diferencias horarias para la comunicación entre los involucrados en el proyecto de software. Se relaciona directamente con la ubicación geográfica de los involucrados.

Muy Alto: brecha temporal entre las ubicaciones geográficas de los involucrados que impide la comunicación sincrónica

Alto: brecha temporal entre las ubicaciones geográficas de los involucrados que permite una comunicación sincrónica de no más de 2 horas diarias

Medio: brecha temporal entre las ubicaciones geográficas de los involucrados que permite una comunicación sincrónica de no más de 4 horas diarias

Bajo: brecha temporal entre las ubicaciones geográficas de los involucrados que permite una comunicación sincrónica de no más de 6 horas diarias

Muy Bajo: brecha temporal nula o casi nula entre las ubicaciones geográficas de los involucrados

Anexo C

Información Adicional del Estudio de Caso sobre Evolución de Factores

Este anexo contiene información adicional relacionada al estudio de caso sobre la estimación y evolución de los factores situacionales descrito en el capítulo 5.

La Tabla C-1 presenta una evaluación del nivel de confianza con el que realizó la valoración de cada factor situacional (no se han considerado aquellos casos no válidos, donde los datos eran incompletos o inconsistentes). Para ello se calcularon el total de proyectos con apreciaciones seguras, razonablemente seguras y dudosas en los valores iniciales y finales de cada factor. Además, se muestra el porcentaje de proyectos en los que el factor mantuvo su calificación de confianza (discriminando entre los que se mantuvieron en seguro o razonablemente seguro y los que se mantuvieron en dudosos), los que mejoraron (por ejemplo, pasando de dudoso a seguro), y los que empeoraron (discriminando los que pasaron a razonablemente seguro o a dudoso).

Tabla C-1. Resultados de nivel de confianza por factor

Factores Situacionales	Proyectos Incompletos	% Proyectos Incompletos	Total Proyectos Válidos	Nivel de Confianza										
				Inicial			Final			Mejóro	Se mantuvo en		Empeoró a ...	
				S	RS	D	S	RS	D		S/RS	D	RS	D
Complejidad del contexto	0	0%	35	13	19	3	27	8	0	54%	37%	0%	9%	0%
Tipo de cliente	0	0%	35	26	7	2	29	6	0	17%	74%	0%	9%	0%
Novedad del contexto	0	0%	35	28	7	0	27	8	0	11%	74%	0%	14%	0%
Reingeniería en el proceso de negocio	7	20%	28	11	17	0	20	8	0	43%	46%	0%	11%	0%
Volatilidad del contexto	1	3%	34	21	11	2	20	13	1	26%	50%	0%	21%	3%
Volatilidad de los requerimientos del cliente	2	6%	33	15	14	4	23	8	2	42%	45%	0%	6%	6%
Inconsistencias en el contexto	6	17%	29	11	13	5	18	8	3	45%	41%	0%	3%	10%
Conflicto de intereses de usuarios	8	23%	27	16	9	2	23	3	1	33%	59%	0%	4%	4%
Rotación de usuarios	9	26%	26	18	6	2	19	6	1	23%	58%	0%	15%	4%
Conocimiento previo del dominio	1	3%	34	24	10	0	26	6	2	24%	53%	0%	18%	6%
Envergadura del proyecto	0	0%	35	13	19	3	22	10	3	40%	37%	0%	14%	9%
Rotación del equipo desarrollador	0	0%	35	18	10	7	19	13	3	34%	37%	6%	20%	3%
Calidad exigida en el software	1	3%	34	14	16	4	22	10	2	35%	53%	6%	6%	0%
Reuso de artefactos de requisitos existentes	0	0%	35	21	6	8	28	7	0	37%	51%	0%	11%	0%
Creación de artefactos de requisitos para reuso	1	3%	34	13	16	5	23	9	2	50%	32%	0%	12%	6%
Pre-Rastreabilidad de los requisitos	1	3%	34	10	15	9	20	12	2	50%	32%	6%	12%	0%
Post-Rastreabilidad de los requisitos	1	3%	34	12	13	9	16	14	4	41%	32%	9%	15%	3%
Granularidad de rastreabilidad de requisitos	22	63%	13	3	6	4	6	6	1	46%	38%	8%	8%	0%
Exigencia de producir el documento ERS	2	6%	33	18	10	5	28	4	1	42%	52%	0%	3%	3%
Restricciones de tiempo y recursos del proyecto	1	3%	34	13	16	5	17	16	1	35%	44%	3%	18%	0%
Modelo de proceso de software	1	3%	34	11	19	4	22	11	1	44%	44%	3%	9%	0%

S: Seguro, RS: Razonablemente Seguro y D: Dudoso

La Tabla C-2 presenta el resultado del análisis de los cambios en el nivel de confianza de aquellos factores que sufrieron cambios relevantes. Para cada factor

se calculó la cantidad de proyectos cuya valoración inicial fue percibida dudosa, razonablemente segura o segura, para los proyectos con cambios a valores no próximos. Se consideran valores próximos o cercanos a los pares: Alto y Muy Alto, Bajo y Muy Bajo, Grande y Muy Grande, Pequeña y Muy Pequeña; otros cambios en las valoraciones asignadas se consideran cambios a valores no próximos.

Tabla C-2. Resultados de la evolución de valores y de nivel de confianza

Factores Situacionales	Total de Proyectos Válidos	% Nivel de Confianza modificado	% Dudosos al inicio	% Dudosos al final	Análisis sobre Proyectos con Cambios Relevantes					
					Proyectos con Cambios Relevantes	% Valores con Cambios Relevantes	% Dudosos al inicio	% Seguros al inicio	% Raz. Seguros al inicio	% Seguros y Raz. Seguros al inicio
1 Complejidad del contexto	35	63%	9%	0%	13	37%	0%	38%	62%	100%
2 Tipo de cliente	35	26%	6%	0%	3	9%	67%	0%	33%	33%
3 Novedad del contexto	35	26%	0%	0%	4	11%	0%	50%	50%	100%
4 Reingeniería en el proceso de negocio	28	54%	0%	0%	15	54%	0%	33%	67%	100%
5 Volatilidad del contexto	34	50%	6%	3%	6	18%	17%	33%	50%	83%
6 Volatilidad de los requerimientos del cliente	33	55%	12%	6%	11	33%	18%	27%	55%	82%
7 Inconsistencias en el contexto	29	59%	17%	10%	8	28%	13%	50%	38%	88%
8 Conflicto de intereses de usuarios	27	41%	7%	4%	4	15%	25%	25%	50%	75%
9 Rotación de usuarios	26	42%	8%	4%	7	27%	14%	71%	14%	86%
10 Conocimiento previo del dominio	34	47%	0%	6%	12	35%	0%	67%	33%	100%
11 Envergadura del proyecto	35	63%	9%	9%	16	46%	6%	38%	56%	94%
12 Rotación del equipo desarrollador	35	57%	20%	9%	14	40%	14%	64%	21%	86%
13 Calidad exigida en el software	34	41%	12%	6%	18	53%	17%	28%	56%	83%
14 Reuso de artefactos de requisitos existentes	35	49%	23%	0%	12	34%	33%	50%	17%	67%
15 Creación de artefactos de requisitos para reuso	34	68%	15%	6%	14	41%	21%	43%	36%	79%
16 Pre-Rastreabilidad de los requisitos	34	62%	26%	6%	15	44%	20%	33%	47%	80%
17 Post-Rastreabilidad de los requisitos	34	59%	26%	12%	12	35%	17%	33%	50%	83%
18 Granularidad de rastreabilidad de requisitos	13	54%	31%	8%	3	23%	33%	33%	33%	67%
19 Exigencia de producir el documento ERS	33	48%	15%	3%	10	30%	20%	60%	20%	80%
20 Restricciones de tiempo y recursos del proyecto	34	53%	15%	3%	13	38%	8%	46%	46%	92%
21 Modelo de proceso de software	34	53%	12%	3%	13	38%	8%	23%	69%	92%

Anexo D

Información Adicional del Estudio de Caso sobre Variantes de Adaptación

Este anexo complementa la información relacionada al estudio descrito en el capítulo 6. Habiendo establecido las funcionalidades derivadas del alcance de cada caso, se examinó cuáles de dichas funcionalidades fueron cubiertas por los EF construidos por cada grupo. Este análisis se detalla en las Tablas D-1 y D-2 para cada caso, indicando la cantidad de EF que cubren cada funcionalidad según cada grupo.

Tabla D-1. Funcionalidades cubiertas para el Sistema de Autogestión de Alumnos

#	FUNCIONALIDADES SEGÚN ALCANCE DEL SISTEMA	Cantidad de EF que cubren la funcionalidad						
		G ₆ V ₁	G ₇ V ₁	G ₈ V ₁	G ₉ V ₂	G ₁₀ V ₂	G ₁₁ V ₂	G ₁₂ V ₃
1	Iniciar sesión	1	1	2	1	1	1	1
2	Cerrar sesión			1				1
3	Modificar datos personales	1		2	1	2	1	2
4	Recuperar la contraseña del alumno		1			1	1	1
5	Inscribir a materias	1	3	3	2	1	1	1
6	Inscribir a finales	1	1	1	2	1	1	1
7	Verificar materias		1	1	1	1	1	1
8	Verificar finales			1	1	1	1	1
9	Consultar calendario académico		1	1	1	1	1	1
10	Consultar el historial académico		1	1	4	3	3	4
11	Consultar la oferta de materias	1		2	2	1	1	1
12	Solicitar el certificado de alumno regular	1	1	1		1		1
13	Solicitar la reincorporación	1	1	1	2	1	2	1
14	Solicitar el certificado de materias aprobadas	1	2	1	1	1	2	1
15	Solicitar el título de grado/pregrado	1	2	1	2	1	2	1
16	Actualización de datos del sistema Guaraní	1	2	3	2	2		1
17	Cargar las fechas de inscripción	1	1					1
18	Cargar las fechas de finales	1	1					1
19	Cargar los precios de trámites							1
20	Enviar notificaciones	1	4	3	1	1	1	1
21	Consultar el historial de trámites	1						1
22	Inscripción a finales automática							1
23	Agregar materias durante la verificación		1	1				1
24	Consultar fechas de finales		1	2	1	1	1	1
25	Consultar datos personales alumno	1		1	1	1		
26	Generación de factura	1	1	1	1	3	1	1
	TOTAL DE FUNCIONALIDADES CUBIERTAS	16	18	20	17	19	16	25
	% DE FUNCIONALIDADES CUBIERTAS	61,54	69,23	76,92	65,38	73,08	61,54	96,15

Tabla D-2. Funcionalidades cubiertas para el Sistema de Gestión de Biblioteca

#	FUNCIONALIDADES SEGÚN ALCANCE DEL SISTEMA	Cantidad de EF que cubren la funcionalidad				
		G ₁ V ₁	G ₂ V ₁	G ₃ V ₂	G ₄ V ₂	G ₅ V ₃
1	Realizar un préstamo	2	1	1	1	1
2	Devolución de un préstamo	1	1	1	1	1
3	Registrar ingreso a la sala de lectura	1	1	1	1	1
4	Consultar el catálogo bibliográfico	1	1	1	1	1
5	Registrar el usuario	1		1	1	1
6	Modificar el usuario					1
7	Consultar el usuario	1				1
8	Registrar el material	1	1	1	1	1
9	Modificar el material					1
10	Dar de baja el material			1		1
11	Consultar el material	1				1
12	Generar libre deuda	1	1	1		1
13	Registrar solicitud de libre deuda	1	1	1		1
14	Generar el listado de usuarios morosos		1	1	1	1
15	Registrar la demanda por insatisfacción		2			1
16	Enviar reclamo de material	1	1			1
17	Actualizar estado del préstamo y del usuario		1			1
18	Consultar demandas insatisfechas	1		1	1	
19	Sancionar a usuario		1	1	1	2
20	Registrar pérdida o rotura del material		1	2	1	2
21	Reponer material por pérdida o rotura	1	2		1	
22	Extender préstamo	1	1	1	1	
TOTAL DE FUNCIONALIDADES CUBIERTAS		14	15	14	12	19
% DE FUNCIONALIDADES CUBIERTAS		63,64	68,18	63,64	54,55	86,36

En las Tablas D-3 y D-4 se presentan parte de los datos estadísticos utilizados en el análisis de resultados de este estudio de caso (ver sección 6.3), aplicando la distribución de t de Student con nivel de significancia $\alpha = 0.05$ para pequeñas muestras. En el caso del Sistema de Autogestión de Alumnos se aplicó $t = 2,920$ con 2 grados de libertad (Tabla D-3) y en el caso del Sistema de Gestión de Biblioteca se aplicó $t = 6,314$ para 1 grado de libertad (Tabla D-4).

Tabla D-3. Resultados estadísticos para el Sistema de Autogestión de Alumnos

Proyecto	TFC (1)	PFC % (2)	TEF (4)	TEP (5)	EPxF (6)	EPxEF (7)	TEX (9)	EXxEF (10)	TACT (11)
VARIANTE V₁									
G ₆ V ₁ CA	16	61,54	12	129	8,06	10,75	17	1,42	3
G ₇ V ₁ CA	18	69,23	18	133	7,39	7,39	20	1,11	4
G ₈ V ₁ CA	20	76,92	25	151	7,55	6,04	19	0,76	4
Media	18,00	69,23	18,33	137,67	7,67	8,06	18,67	1,10	3,67
Desvío Estándar	2,00	7,69	6,51	11,72	0,35	2,43	1,53	0,33	0,58
Límite confianza superior	21,37	82,19	29,30	157,42	8,26	12,15	21,24	1,65	4,64
Límite confianza inferior	14,63	56,27	7,36	117,91	7,08	3,97	16,09	0,54	2,69
VARIANTE V₂									
G ₉ V ₂ CA	17	65,38	25	137	8,06	5,48	15	0,6	6
G ₁₀ V ₂ CA	19	73,08	20	126	6,63	6,30	26	1,3	5
G ₁₁ V ₂ CA	16	61,54	18	82	5,13	4,56	38	2,11	4
Media	17,33	66,67	21,00	115,00	6,61	5,45	26,33	1,34	5,00
Desvío Estándar	1,53	5,88	3,61	29,10	1,47	0,87	11,50	0,76	1,00
Límite confianza superior	19,91	76,57	27,08	164,06	9,08	6,91	45,73	2,61	6,69
Límite confianza inferior	14,76	56,76	14,92	65,94	4,14	3,98	6,94	0,06	3,31
VARIANTE V₃									
G ₁₂ V ₃ CA	25	96,15	27	173	6,92	6,41	24	0,89	6

Tabla D-4. Resultados estadísticos para el Sistema de Gestión de Biblioteca

Proyecto	TFC (1)	PFC % (2)	TEF (4)	TEP (5)	EPxF (6)	EPxEF (7)	TEX (9)	EXxEF (10)	TACT (11)
VARIANTE V₁									
G ₁ V ₁ C _B	14	63,64	14	124	8,86	8,86	33	2,36	8
G ₂ V ₁ C _B	15	68,18	14	111	7,40	7,93	14	1,00	9
Media	14,5	65,91	14	117,5	8,13	8,40	23,5	1,68	8,5
Desvío Estándar	0,71	3,21	0,00	9,19	1,03	0,66	13,44	0,96	0,71
Límite confianza superior	17,66	80,24	14,00	158,54	12,74	11,33	83,48	5,97	11,66
Límite confianza inferior	11,34	51,58	14,00	76,46	3,52	5,46	-36,48	-2,61	5,34
VARIANTE V₂									
G ₃ V ₂ C _B	14	63,64	15	57	4,07	3,80	12	0,80	7
G ₄ V ₂ C _B	12	54,55	14	57	4,75	4,07	19	1,36	9
Media	13,00	59,10	14,50	57,00	4,41	3,94	15,50	1,08	8,00
Desvío Estándar	1,41	6,43	0,71	0,00	0,48	0,19	4,95	0,40	1,41
Límite confianza superior	19,31	87,79	17,66	57,00	6,56	4,79	37,60	2,85	14,31
Límite confianza inferior	6,69	30,40	11,34	57,00	2,26	3,08	-6,60	-0,69	1,69
VARIANTE V₃									
G ₅ V ₃ C _B	19	86,36	24	137	7,21	5,71	28	1,17	6

Referencias

- Abdullah, A., & Khan, H. (2015). *FreGsd: A Framework for Global Software Requirement Engineering*. *Journal of Software*, 10, 10, pp. 1189-1198.
- Agarwal, R., & Tanniru, M. (1990). *Knowledge Acquisition Using Structured Interviewing: An Empirical Investigation*. *Journal of Management Information System*, 7, 1, pp. 123-140.
- Aguilar, J., Garrigós, I., & Mazón, J. (2016). *Requirements Engineering in the Development Process of Web Systems: A Systematic Literature Review*. *Journal of Applied Sciences. Acta Polytechnica Hungarica*, 13, 3, pp. 61-80.
- Alexander, I., & Robertson, S. (2004). *Understanding Project Sociology by Modeling Stakeholders*. *IEEE Software*, 21, 1, pp. 23-27.
- Alexander, I., & Beus-Dukic, L. (2009). *Discovering Requirements: How to Specify Products and Services*. Chichester, West Sussex, England: John Wiley & Sons.
- Al-Zawahreh, H., & Almakadmeh, K. (2015). *Procedural Model of Requirements Elicitation Techniques*. *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication (IPAC '15)*. Article No. 65. Batna, Algeria.
- Antonelli, L., Rossi, G., Leite, J., & Oliveros, A. (2012). *Deriving Requirements Specifications from the Application Domain Language Captured by Language Extended Lexicon*. *Proceedings of XV Workshop on Requirements Engineering*. Buenos Aires, Argentina.
- Arias, M., Buccella, A., & Cechich, A. (2017). *A Framework for Managing Requirements of Software Product Lines*. *Proceedings of XLIII CLEI, Conferencia Latinoamericana de Informática*. Córdoba, Argentina.
- Ayora, C., Alférez, G., Torres, V., & Pelechano, V. (2011). *Procesos de Negocio Auto-Adaptables al Contexto*. *Actas de VII Jornadas de Ciencia e Ingeniería de Servicios*. Coruña, España.
- Bakhat, K., Sarwar, A., Motla, Y., & Akhta, M. (2015). *A Situational Requirement Engineering Model for an Agile Process*. *Bahria University Journal of Information & Communication Technology*, 8, 1, pp. 21-26.

- Ballejos, L. (2009). *Herramientas para el Análisis de Sistemas de Información en Ambientes Interorganizacionales*, Tesis Doctoral. Facultad Regional Santa Fe, Universidad Tecnológica Nacional, Argentina.
- Beck, K., & Fowler, M. (2000). *Planning Extreme Programming. First Edition*. Addison Wesley.
- Bilal, H., Ilyas, M., Tariq, Q., & Hummayun, M. (2016). *Requeriments Validation Techniques: An Empirical Study*. International Journal of Computer Applications, 148, 14. Nueva York, EEUU.
- Brinkkemper, S. (1996). *Method Engineering: Engineering of Information Systems Development Methods and Tools*. Information & Software Technology, 38, 4, pp. 275-280.
- Bucher, T., Klesse, M., Kurpjuweit, S., & Winter, R. (2007). *Situational Method Engineering*. Situational method engineering: fundamentals and experiences, pp. 33-48. Springer US.
- Burton, A., Shadbolt, N., Rugg, G., & Hedgecock, A. (1990). *The Efficacy of Knowledge Elicitation Techniques: A Comparison Across Domains and Levels of Expertise*. Knowledge Acquisition, University of Nottingham, UK: Department of Psychology, 2, pp. 167-168.
- Byrd, T., Cossick, K., & Zmud, R. (1992). *A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques*. MIS Quarterly, 16, 1, pp. 117-138.
- Cavazos, G. (1988). *Probabilidades y Estadística. Aplicaciones y métodos*, 1º edición. Mc. Graw Hill. ISBN: 968-451-856-0.
- Carrizo, D., Dieste, O., & Juristo, N. (2008). *Study of Elicitation Techniques Adequacy*. Proceedings of 11th Workshop on Requirements Engineering, pp. 104-114. Barcelona, España.
- Carrizo Moreno, D. (2009). *Marco para la Selección de Técnicas para Educación de Requisitos*. Tesis Doctoral. Universidad Politécnica de Madrid, Facultad de Informática, Madrid, España.
- Cetina, C., Giner, P., Fons, J., & Pelechano, V. (2009). *Autonomic Computing Through Reuse of Variability Models at Runtime: The Case of Smart Homes*. Computer, IEEE Computer Society, 42, 10, pp. 37-43.

- Chao, J., & Salvendy, G. (1995). *Impact of Cognitive Abilities of Experts on the Effectiveness of Elicited Knowledge*. Behaviour and Information Technology, 14, 3, pp. 174-182.
- Chen, L., & Babar, M. (2010). *Variability Management in Software Product Lines: An Investigation of Contemporary Industrial Challenges*. Proceedings of the 14th International Conference on Software Product Lines: going beyond (SPLC 2010), pp. 166-180. Springer-Verlag Berlin Heidelberg.
- Clements, P., & Northrop, L. (2002). *Software Product Lines: Practices and Patterns*. Addison-Wesley.
- Cockburn, A. (2000). *Writing Effective Use Cases*. Addison-Wesley Professional.
- Coulin, C., Zowghi, D., & Sahraoui, A. (2006). *A Situational Method Engineering Approach to Requirements Elicitation Workshops in the Software Development Process*. Software Process: Improvement and Practice, 11, 5, pp. 451–464.
- Coulin, C. (2007). *A Situational Approach and Intelligent Tool for Collaborative Requirements Elicitation*. Tesis Doctoral, University of Technology. Sydney, Australia.
- Delgado, A., & Calegari, D. (2017). *BPMN 2.0 Based Modeling and Customization of Variants in Business Process Families*. Proceedings of XLIII CLEI, Conferencia Latinoamericana de Informática. Córdoba, Argentina.
- Dhaliwal, J., & Benbasat, I. (1990). *A Framework for the Comparative Evaluation of Knowledge Acquisition Techniques*. Knowledge Acquisition, 2, 2, pp. 145-166.
- Doorn, J., Hadad, G., & Kaplan, G. (2002). *Comprendiendo el Universo de Discurso Futuro*. Proceedings of V Workshop en Ingeniería de Requisitos, pp. 117-131. Valencia, España.
- Ebling, T., Nicolas Audy, J., & Prikladnicki, R. (2009). *Towards a Requirements Reuse Method Using Product Line in Distributed Environments*. Proceedings of 12th Workshop on Requirements Engineering, Universidad Técnica Federico Santa María, pp. 91-102. Valparaíso, Chile.
- Escalona, M., & Koch, N. (2004). *Requirements Engineering for Web Applications – A Comparative Study*. Journal of Web Engineering, 2, 3, pp.193-212.

- Fazlollahi, B., & Tanniru, M. (1991). *Selecting a Requirement Determination Methodology-Contingency Approach Revisited*. Information & Management 21, 5, pp. 291-303.
- Firesmith, D., & Henderson-Sellers, B. (2002). *The OPEN Process Framework: An Introduction*. Harlow, Herts, UK: Addison-Wesley.
- Firesmith, D. (2004). *Creating a Project-Specific Requirements Engineering Process*. Journal of Object Technology, 3, 5, pp. 31-44.
- Freivalds, A., & Niebel, B. (2013). *Niebel's Methods, Standards, & Work Design*. McGraw-Hill Education.
- Galster, M., Weyns, D., Tofan, D., Michalik, B., & Avgeriou, P. (2014). *Variability in Software Systems - A Systematic Literature Review*. IEEE Transactions on Software Engineering, IEEE, 40, 3, pp. 282-306.
- Gorschek, T. (2011). *Requirements Engineering Process Maturity Model*. Uni-REPM Version: 0.9 CR.
- Grenon, P. (2009). *Defining Extensions to WSMO for Service Oriented Architectures for all Capturing Contextual Information*. Technical Report D3.4.7, Service Oriented Architectures for All. European Commission within the Seventh Framework Programme.
- Groefsema, H., Bulanov, P., & Aiello, M. (2011). *Declarative Enhancement Framework for Business Processes*. Proceedings of International Conference on Service-Oriented Computing, pp.495-504. New York: Springer.
- Hadad, G. (2008). *Uso de Escenarios en la Derivación de Software*. Tesis Doctoral, Facultad de Ciencias Exactas, Universidad Nacional de La Plata, Argentina.
- Hadad, G., Doorn, J., & Kaplan, G. (2009). *Creating Software System Context Glossaries*. Encyclopedia of Information Science and Technology, Second Edition, Hershey. IGI Global, pp.789-794.
- Hadad, G., Doorn, J., & Kaplan, G. (2009 b). *Explicitar Requisitos del Software Usando Escenarios*. Proceedings of 12th Workshop on Requirements Engineering, Universidad Técnica Federico Santa María, pp. 63-74. Valparaíso, Chile.
- Hadad, G., & Doorn, J. (2013). *Introducing Variability in a Client-Oriented Requirements Engineering Process*. Proceedings of ER@BR2013 -

- Requirements Engineering @ Brazil, Pontifícia Universidade Católica do Rio de Janeiro, ISSN: 1613-0073, pp. 8-13. Río de Janeiro, Brasil.
- Hadad, G., Ledesma, V., & Doorn, J. (2014). *Proceso de Requisitos Adaptable a Factores Situacionales*. Memorias de XVI Workshop de Investigadores en Ciencias de la Computación, pp. 448-452. Ushuaia, Argentina.
- Hadad, G. (2017). *Notas de Clase: Ingeniería de Requisitos Orientada al Cliente*. Asignatura Ingeniería de Requerimientos, DIIT, UNLaM.
- Hallerbach, A., Bauer, T., & Reichert, M. (2008). *Managing Process Variants in the Process Lifecycle*. Proceedings of 10th International Conference on Enterprise Information Systems, pp. 154-161. Barcelona, España.
- Harmsen, F., Brinkkemper, H., & Oei, H. (1994). *Situational Method Engineering for Information System Project Approaches*. Methods and Associated Tools for the Information Systems Life Cycle. Proceedings of the IFIP WG 8.1 Working Conference. IFIP Transactions A- 55, pp. 169-194. Maastricht, Netherlands: A.A. Verrijn Stuart and T.W. Olle (Eds.).
- Haugen, Ø., Møller-Pedersen, B., Oldevik, J., Olsen, G.K., Svendsen, A. (2008) *Adding Standardized Variability to Domain Specific Languages*. Proceedings of the 12th International Software Product Line Conference (SPLC 2008), vol. 1, pp. 139–148. IEEE Computer Society, Limerick.
- Haugen, Ø., & Øgård, O. (2014). *BVR – Better Variability Results*. Proceedings of International Conference on System Analysis and Modeling: Models and Reusability (SAM 2014), Lecture Notes in Computer Science, Vol. 8769, pp. 1-15. Springer, Cham.
- Haumer, P. (2005). *IBM Rational Method Composer: Part 1: Key concepts*. Recuperado el 17/3/2016, <https://www.ibm.com/developerworks/rational/library/dec05/haumer/>
- Henderson-Sellers, B., & Ralyté, J. (2010). *Situational Method Engineering: State-of-the-Art Review*. Journal of Universal Computer Science, 16, 3, pp. 424-478.
- Henderson-Sellers, B., Ralyté, J., Ågerfalk, P., & Rossi, M. (2014). *Situational Method Engineering*. Springer-Verlag Berlin Heidelberg.

- Hickey, A., & Davis, A. (2003). *Elicitation Technique Selection: How do Experts do It?* Proceedings of 11th IEEE International Requirements Engineering Conference, IEEE, pp.169-178.
- Hickey, A., & Davis, A. (2004). *A Unified Model of Requirements Elicitation*. Journal of Management Information Systems. Information Systems Design- Theory and Methodology, 20, 4, pp. 65-84.
- IBM. (2005). *An Architectural Blueprint for Autonomic Computing*. Recuperado el 21/3/2017, <http://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf>
- Jafarinezhad, O., & Ramsin, R. (2012). *Development of Situational Requirements Engineering Processes: A Process Factory Approach*. Proceedings of 36th IEEE International Conference on Computer Software and Applications, pp. 279-288.
- Jiang, L., & Eberlein, A. (2008). *A Framework for Requirements Engineering Process Development (FRERE)*. Proceedings of 19th Australian Conference on Software Engineering (ASWEC 2008), pp. 507-516. Perth, Australia.
- Kabaale, E., & Mayoka Kituyi, G. (2015). *A Theoretical Framework for Requirements Engineering and Process Improvement in Small and Medium Software Companies*. Business Process Management Journal, 21, 1, pp.80-99.
- Kaindl, H. (2000). *A Design Process Based on a Model Combining Scenarios with Goals and Functions*. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 30, 5, pp. 537-551.
- Kaplan, G., Doorn, J., & Gigante, N. (2013). *Evolución Semántica de Glosarios en los Procesos de Requisitos*. Memorias del XVIII Congreso Argentino de Ciencias de la Computación. Mar del Plata, Argentina.
- Keil, M., & Carmel, E. (1995). *Customer- Developer Links in Software Development*. Communications of the ACM, 38, 5, pp.33-44.
- Khan, H., bin Mahrin, M., & bt Chuprat, S. (2014). *Factors for Tailoring Requirement Engineering Process: A Review*. International Journal of Software Engineering and Technology, 1, 1, pp. 7-18.
- Khan, H., bin Mahrin, M., & Mali, M. (2016). *Situational Requirement Engineering Framework for Global Software Development: Formulation and Design*.

- Bahria University Journal of Information & Communication Technologies, 9, 1, pp. 74-84.
- Kim, J., & Courtney, J. (1988). *A Survey of Knowledge Acquisition Techniques and their Relevance to Managerial Problem Domains*. Decision Support Systems, 4, 3, pp.269-284.
- Kotonya, G., & Sommerville, I. (1998). *Requirements Engineering: Processes and Techniques*. John Wiley & Sons.
- Kumar, K., & Welke, R. (1992). *Methodology Engineering: a Proposal for Situation-Specific Methodology Construction*. W.W. Cotterman & J.A. Senn (Eds.), Challenges and strategies for research in systems development, John Wiley & Sons, pp. 257-269.
- Kyburz-Graber, R. (2004). *Does case-study methodology lack rigour? The need for quality criteria for sound case-study research, as illustrated by a recent case in secondary and higher education*. Journal of Environmental Education Research, 10, 1, pp.53-65.
- Lauesen, S. (2002). *Software Requirements: Styles and Techniques*. Addison-Wesley.
- Ledesma, V., Hadad, G, Doorn, J., & Bedetti, N. (2018). *Estimación de los Factores Situacionales del Proceso de Requisitos*. WER 2018 - 21st Workshop on Requirements Engineering, Río de Janeiro, Brasil.
- Leffingwell, D., & Widrig, D. (2003). *Managing Software Requirements - A Unified Approach*. Second edition. Addison-Wesley Object Technology Series.
- Leite, J. (1994). *Engenharia de Requisitos. Notas Tutoriales*, material de enseñanza en el curso Requirements Engineering, Computer Science Department of PUC- Río, Brasil.
- Leite, J., Hadad, G., Doorn, J., & Kaplan, G. (2000). *A Scenario Construction Process*. Requirements Engineering Journal, 5, 1, pp.38-61.
- Leite, J., Doorn, J., Kaplan, G., Hadad, G., & Ridao, M. (2004). *Defining System Context Using Scenarios*. Perspectives on Software Requirements, Kluwer Academic Publishers, chapter 8, pp.169-199. EEUU.
- Leite, J., Moraes, E., & Castro, C. (2007). *A Strategy for Information Sources Identification*. Proceedings of X Workshop on Requirements Engineering (WER'07), pp. 25-34. Toronto, Canadá.

- Lloyd, W. (2001). *Tools and Techniques for Effective Distributed Requirements Engineering: An Empirical Study*. Master Thesis. Virginia Tech.
- Lloyd, W., Rosson, M., & Arthur, J. (2002). *Effectiveness of Elicitation Techniques in Distributed Requirements Engineering*. Proceedings of 10th IEEE Joint International Conference on Requirements Engineering. Essen, Alemania.
- Lobo, L., & Arthur, J. (2005). *An Objectives-Driven Process for Selecting Methods to Support Requirements Engineering Activities*. Proceedings of 29th Annual IEEE/NASA Software Engineering Workshop, IEEE, pp. 118-130. Greenbelt, EEUU.
- Macaulay, L. (1993). *Requirements Capture as a Cooperative Activity*. Proceedings of IEEE International Symposium on Requirements Engineering, IEEE, pp. 174-181. San Diego, EEUU.
- Maiden, N., & Rug, G. (1996). *ACRE: Selecting Methods for Requirements Acquisition*. Software Engineering Journal, 11, 3, pp.183-192.
- Martínez-Ruiz, T., García, F., Piattini, M., & Munch, J. (2011). *Modelling Software Process Variability: An Empirical Study*. IET Software, 5, 2, pp. 172-187.
- Maynard, H. (1991). *Manual de Ingeniería y Organización Industrial*. Reverté.
- McGregor, J. (2004). *Software Product Lines*. Journal of Object Technology, 3, 3, pp. 65-74. Zurich, Suiza.
- Mighetti, J., & Hadad, G. (2015). *Proceso de Requisitos Adaptado para Mitigar Amenazas en el Desarrollo Global de Software*. Memorias de 3er Congreso Nacional de Ingeniería Informática / Sistemas de Información (CONAISSI). Buenos Aires, Argentina.
- Montero, I., Peña, J., & Ruiz-Cortés, A. (2008). *Representing Runtime Variability in Business-Driven Development Systems*. Proceedings of the Seventh International Conference on Composition-Based Software Systems (ICCBSS08), IEEE, pp. 605-608. Madrid, España.
- Oliveros, A., & Antonelli, L. (2015). *Técnicas de Elicitación de Requerimientos*. Memorias de XXI Congreso Argentino de Ciencias de la Computación (CACIC). Junín, Argentina.
- Olsson, T., Doerr, J., Koenig, T., & Ehresmann, M. (2005). *A Flexible and Pragmatic Requirements Engineering Framework for SME*. Proceedings of 1st

- International Workshop on Situational Requirements Engineering Processes (SREP'05). Paris, France.
- Pacini, K., & Braga, R. (2015). *An Approach for Reusing Software Process Elements based on Reusable Asset Specification: A Software Product Line Case Study*. Proceedings of the 10th International Conference on Software Engineering Advances (ICSEA 2015), pp. 200-206. Barcelona: España.
- Pol'la, M., Buccella, A., Cechich, A., & Arias, M. (2014). *Un Modelo de Metadatos para la Gestión de la Variabilidad en Líneas de Productos de Software*. Memorias de XLIII Jornadas Argentinas de Informática e Investigación Operativa (43JAIO), XV Simposio Argentino de Ingeniería de Software. Buenos Aires, Argentina.
- Potts, C. (1995). *Using Schematic Scenarios to Understand User Needs*. Proceedings of Symposium on Designing Interactive Systems: Processes, Practices and Techniques, pp.247-256. Ann Arbor, EEUU.
- Pressman, R. (2010). *Ingeniería del Software. Un enfoque práctico*. Séptima edición. México: McGraw-Hill.
- Puhlmann, F., Schnieders, A., Weiland, J., & Weske, M. (2005). *Variability Mechanisms for Process Models*. PESOA-Report No. TR 17/2005.
- Ralyté, J., Deneckère, R., & Rolland, C. (2003). *Towards a Generic Model for Situational Method Engineering*. Proceedings of the 15th International Conference on Advanced Information Systems Engineering (CAISE'03), pp. 95-110. Klagenfurt, Austria.
- Ralyté, J. (2013). *Situational Method Engineering in Practice: A Case Study in a Small Enterprise*. Proceedings of 25th International Conference on Advanced Information Systems Engineering, pp. 17-24. Valencia, España.
- Razavian, M., & Khosravi, R. (2008). *Modeling Variability in Business Process Models Using UML*. Proceedings of the 5th International Conference on Information Technology: New Generations. ITNG 2008. Las Vegas, EEUU: IEEE.
- Rochowiak, D. (1988). *Extensibility and Completeness: An Essay on Scientific Reasoning*. The Journal of Speculative Philosophy, 2, 4, pp. 241-266.
- Rodríguez-Lora, V., Henao-Cálad, M., & Valencia Arias, A. (2016). *Taxonomías de Técnicas y Herramientas para la Ingeniería del Conocimiento: Guía para el*

- Desarrollo de Proyectos de Conocimiento*. Ingeniare. Revista Chilena de Ingeniería, 24,2, pp. 351-360.
- Rolland, C. (2008). *Method Engineering: Towards Methods as Services*. Making Globally Distributed Software Development a Success Story. Proceedings of International Conference on Software Process (ICSP 2008), pp.10–11. Leipzig, Alemania.
- Ross, D., & Schoman, A. (1977). "Structured analysis for requirements definition", IEEE TSE, Special Issue on Requirements Analysis, Vol.3, N°1, pp. 6-15.
- Saeki, M. (2003). *Embedding Metrics into Information Systems Development Methods: An Application of Method Engineering Technique*. Proceedings of 15th International Conference on Advanced Information Systems Engineering (CAiSE'03). Lecture Notes in Computer Science, Vol. 2681, pp. 374-389. Springer, Berlin, Heidelberg.
- Santos, E., Castro, J., Sánchez, J., & Pastor, O. (2010). *A Goal-Oriented Approach for Variability in BPMN*. Proceedings of 13th Workshop on Requirements Engineering, pp. 17-28. Cuenca, Ecuador.
- Schnieders, A., & Puhlmann, F. (2006). *Variability Mechanisms in E-Business Process Families*. Proceedings of 9th International Conference on Business Information Systems (BIS 2006), pp. 583-601. Klagenfurt, Austria.
- Seyff, N., Maiden, N., Karlsen, K., Lockerbie, J., Grünbacher, P., Graf, F., & Ncube, C. (2009). *Exploring How to Use Scenarios to Discover Requirements*. Requirements Engineering Journal, 14, 2, pp.91-111.
- Shadbolt, N., & Burton, A. (1989). *The Empirical Study in Knowledge Elicitation*. ACM-SIGART Special Issue on Knowledge Acquisition, Issue 108, pp. 15-18. Nueva York, EEUU.
- Schmid, K. (2004). *Strategically Defining and Exploiting Product Portfolios with a Product Line Approach*. En el libro Perspectives on Software Requirements, Kluwer Academic Publishers, Leite, J.C.S.P., & Doorn, J.H. (eds.), Estados Unidos, ISBN: 1-4020-7625-8, capítulo 10, 2004, pp.69-90.
- Simidchieva, B., Clarke, L., & Osterweil, L. (2007). *Representing Process Variation with a Process Family*. Proceedings of the International Conference on Software Process (ICSP 2007), pp. 109-120. Minneapolis, EEUU.

- Simmonds, J., Bastarrica, M., Silvestre, L., & Quispe, A. (2011). *Analyzing Methodologies and Tools for Specifying Variability in Software Processes*. Computer Science Department, Universidad de Chile, Santiago. Recuperado el 2/5/2017, http://swp.dcc.uchile.cl/TR/2011/TR_DCC-20111104-012.pdf
- Sommerville, I., & Sawyer, P. (1997). *Requirements Engineering, a Good Practice Guide*. Wiley.
- Sommerville, I. (2010). *Software Engineering*. 9th edition. Addison-Wesley.
- Souer, J., van de Weerd, I., Versendaal J., & Brinkkemper, S. (2005). *Situational Requirements Engineering for the Development of Content Management System-based Web Applications*. Institute of Information and Computing Sciences, Utrecht University Technical Report UU-CS-2005-050. Utrecht, Netherlands.
- Thiel, S. (2002). *On the Definition of a Framework for an Architecting Process Supporting Product Family Development*. Proceedings of International Workshop on Software Product-Family Engineering. Lecture Notes in Computer Science, Vol. 2290. Springer, Berlin, Heidelberg.
- Van de Weerd, I., Brinkkemper, S., & Versendaal, J. (2010). *Incremental Method Evolution in Global Software Product Management: A Retrospective Case Study*. Information and Software Technology, 52, 7, pp.720-732. Utrecht, Netherlands.
- Wagner, W., Chung, Q., & Najdawi, M. (2003). *The Impact of Problem Domains and Knowledge Acquisition Techniques: A Content Analysis of P/ OM Expert System Case Studies*. Expert Systems with Applications, 24, 1, pp.79-86. Villanova, EEUU.
- Wheelen, T., & Hunger, D. (2012). *Strategic Management and Business Policy. Toward Global Sustainability*. 13th Edition. Pearson.
- Wieggers, K., & Beatty, J. (2013). *Software Requeriments*. Third Edition. Washington, EEUU: Microsoft Press.
- Yin, R. K. (1981). *The case study crisis: some answers*. Administrative Science Quarterly, 26, 1, pp. 58-65.
- Zanlorenci, E., & Burnett, R. (1998). *Modelo para Qualificação da Fonte de Informação do Cliente e de Requisito Funcional*. Proceedings of 1st

Workshop on Requirements Engineering (WER'98), pp. 39-48. Maringá, Brasil.

Zowghi, D., Firesmith, D., & Henderson-Sellers, B. (2005). *Using the OPEN Process Framework to Produce a Situation-Specific Requirements Engineering Method*. Proceedings of 1st International Workshop on Situational Requirements Engineering Processes: Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes (SREP'05), pp.59-74. Paris, Francia, J. Ralyté, P.J. Ågerfalk, N. Kraiem (Eds.).