

# Adaptación Dinámica de un Proceso de Requisitos Orientado al Cliente

Viviana A. Ledesma  
*Escuela de Informática*  
*Universidad Nacional del*  
*Oeste*  
vledesma@unlam.edu.ar

Graciela D. S. Hadad  
*Facultad de Ingeniería y*  
*Tecnología Informática*  
*Universidad de Belgrano*  
*Escuela de Informática*  
*Universidad Nacional del*  
*Oeste*  
graciela.hadad@comunidad  
.ub.edu.ar

Jorge H. Doorn  
*Escuela de Informática*  
*Universidad Nacional del*  
*Oeste*  
jdoorn@exa.unicen.edu.ar

## Abstract

*La creación de un producto software de calidad depende en gran medida de la correcta especificación de sus requisitos. La experiencia indica que cada proyecto tiene características particulares, las que podrían ser definidas a partir de factores situacionales observables relacionados con el contexto de aplicación y con el proyecto en sí mismo. Para un tratamiento adecuado de estas características, es altamente recomendable efectuar cambios en los procesos de desarrollo para tener en cuenta las mismas, siendo por ende necesario disponer de un método claro que guíe la decisión sobre cómo adaptar el proceso y los modelos a construir. Al inicio del proceso de requisitos no es sencillo percibir las particularidades que cada caso presenta. Durante la aplicación por más de una década de una estrategia de Ingeniería de Requisitos basada en modelos en lenguaje natural, se han realizado de modo informal algunas adaptaciones al proceso mismo, sin haber elaborado un método cuidadoso que guíe las decisiones sobre qué aspectos se deben considerar en cada caso. En este artículo se propone optimizar la aplicación de la estrategia de requisitos a través de un mecanismo que adapte la misma en forma dinámica, según un conjunto de factores definidos para una situación particular.*

## 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 [1] y Open Process Framework [2].

La Ingeniería de Métodos Situacional se dedica precisamente a la construcción de métodos ajustados a situaciones específicas para el desarrollo de software [3]. 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 [4].

Parte de la tarea se refiere a identificar dichos indicadores; muchos de estos se basan en factores observables, como por ejemplo, el grado de reingeniería del proceso de negocio, la experiencia en el dominio de la aplicación por parte del equipo de desarrollo, o la complejidad del contexto, entre otros.

Lo ideal sería conocer y tomar en cuenta dichos factores antes de comenzar el proceso de desarrollo, pero en la realidad esto no es siempre posible, debido a que algunos factores no se conocen con precisión desde un principio, mientras que otros pueden cambiar a medida que el proyecto avanza. Por lo tanto, una visión dinámica de la adaptación de un proceso de desarrollo de software logra un mejor rendimiento del mismo y, por ende, del producto.

A menudo surgen cuestiones relacionadas con la obligación de llevar a cabo todos los pasos del proceso de producción de requisitos, independientemente de la experiencia y conocimiento de los involucrados, aún cuando este proceso haya sido concebido para un contexto con otras experiencias y conocimientos. En algunas circunstancias, surge la oportunidad de acortar el proceso mediante la supresión o la simplificación 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 actividades. Sería recomendable que los gerentes de

proyecto tomasen decisiones sobre estas elecciones, refinamientos o ajustes al proceso en cada proyecto particular a realizar.

En este artículo se presenta la adaptación de un proceso de Ingeniería de Requisitos (IR) basado en modelos descritos en lenguaje natural, considerando los factores situacionales relevantes en cada caso. Además, se propone un método para la construcción del propio proceso de requisitos, a partir de la evaluación de estos factores mientras avanza el proceso mismo de IR. En la sección siguiente, se exponen conceptos de variabilidad de procesos, creación de procesos según situaciones específicas y el proceso de Ingeniería de Requisitos basado en modelos escritos en lenguaje natural, sobre el que se elabora la presente propuesta. En la sección 3, se presenta la propuesta de un proceso de IR adaptable en base a las condiciones del contexto de aplicación y del proyecto de software, mientras que en la sección 4 se describe un método para realizar dicha adaptación en forma dinámica, basada en la evolución de los factores situacionales. Finalmente, se exponen conclusiones y trabajos futuros.

## 2. Antecedentes

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 (producto) de alta calidad. La primera actividad de un proceso de desarrollo de un producto consiste en definir el resultado esperado. Cuando el producto es un sistema de software, esta actividad inicial es un proceso de IR, cuyo resultado es una Especificación de Requisitos de Software.

El proceso de IR es particularmente diferente de otras actividades 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 [5]. 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.

A continuación se describen aspectos relacionados con la creación y adaptación de procesos de software, en general, y la aplicación de un proceso de Ingeniería de Requisitos, en particular, el cual es foco de la presente propuesta de adaptación.

### 2.1. Adaptación de Procesos de Software

Algunas áreas, como por ejemplo Líneas de Producto de Software [6], Computación Autónoma [7] y Modelado de Procesos del Negocio [8], aplican el análisis de variabilidad para hacer frente a los cambios que se producen en el contexto y en el software.

Se conoce como *variabilidad* a la habilidad de un proceso o un producto para ser extendido, personalizado o configurado para ser reutilizado bajo un determinado contexto, cuya detección involucra identificar qué varía, porqué y cuáles son las variantes posibles [9-11]. La gestión de la variabilidad incluye:

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

En la práctica, suele ocurrir que los procesos a menudo son ajustados debido al acaecimiento de diversas circunstancias, a veces de una manera improvisada mientras el proyecto está en marcha. Por lo tanto, se debe considerar la situación asociada al contexto y al proyecto mismo y la forma en que ésta puede evolucionar a lo largo del desarrollo del proyecto, de manera tal de poder adaptar el proceso del mejor modo posible. Es probable que se pueda anticipar y planificar la adaptación del proceso en base a ciertas características que se conocen desde un principio, sin embargo, algunas de estas pueden cambiar de forma dinámica, requiriendo una re-adaptación del proceso [12].

La Ingeniería de Métodos se enfoca en el diseño, construcción y adaptación de métodos, técnicas y herramientas de soporte para el desarrollo de sistemas de información [13]. 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 optimizar la productividad.

Según Brinkkemper [13], un proceso se conforma de fragmentos o bloques más pequeños, que se caracterizan por tener un objetivo final 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 la existencia de bloques de producto, o incluso bloques que agrupan tanto proceso como producto [12, 14, 15].

Una rama importante dentro de la Ingeniería de Métodos es la Ingeniería de Métodos Situacional (IMS)

[3], cuyo objetivo es la construcción de métodos específicos para un proyecto, que se ajusten a las situaciones particulares de su contexto, de modo flexible, adaptable, ágil y repetible. Se entiende como *situación* a la combinación de circunstancias en un momento dado y probablemente para una organización específica, 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 características del proyecto de software [16].

En este sentido, el proceso se define como un conjunto de bloques, donde algunos de estos 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 bloques para una situación particular [14].

Los principios de la IMS se han aplicado en la IR para definir procesos de requisitos adaptables a circunstancias particulares mediante el uso de componentes modulares existentes [2, 17, 18].

## 2.2. Un Proceso de Ingeniería de Requisitos

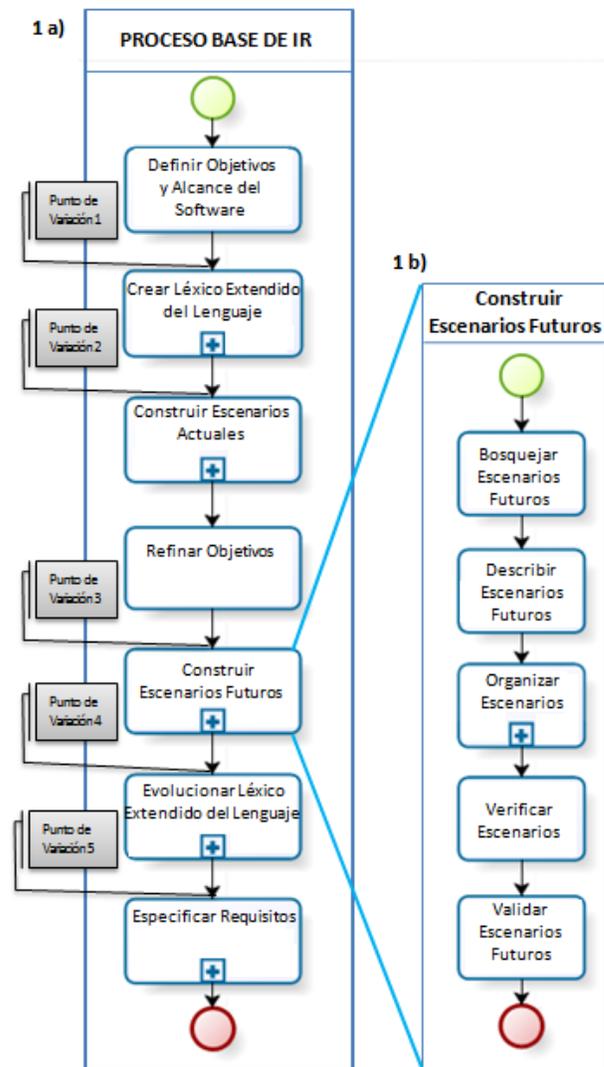
Dado que el conocimiento del dominio del problema se expresa sobre todo en lenguaje natural, el uso de un enfoque basado en representaciones de requisitos en lenguaje natural favorece el compromiso de los clientes y usuarios, lo que puede incrementar la probabilidad de éxito del proyecto [19]. Una manera 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, en particular, el vocabulario empleado en el contexto de aplicación [20]. 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 [21-26].

Las estrategias de IR rara vez se adaptan a su contexto [20, 22, 24, 27], en general, presentan un enfoque top-down, bottom-up y, eventualmente, middle-out para la construcción de modelos, indicando heurísticas de elicitación y técnicas de verificación y/o validación a utilizar. Algunos autores consideran variantes casi exclusivamente en la elicitación de información, teniendo en cuenta algunas características contextuales, como el número de fuentes de información, la distribución geográfica del usuario, la disponibilidad de tiempo del usuario, su experiencia en el dominio, entre otras, con el fin de seleccionar las técnicas de recolección de hechos

más apropiada [28-31]. Aunque han surgido algunas propuestas para diseñar un proceso de IR para un proyecto específico, basándose en la selección de diversas técnicas de IR existentes [32-34].

El proceso de IR basado en modelos escritos en lenguaje natural, representado en la Figura 1(a), involucra todas las actividades propias de cualquier proceso de requisitos: elicitación, modelado, análisis (verificación y validación) y gestión en sus sub-procesos.



**Figura 1. Proceso base de Ingeniería de Requisitos.**

Este proceso de IR consiste básicamente en los siguientes sub-procesos:

- (i) Definir 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 Léxico Extendido del Lenguaje [35].
- (iii) Comprender el contexto de aplicación, con el apoyo de un conjunto de Escenarios Actuales que representan el comportamiento observado en el contexto de aplicación [25].
- (iv) Refinar el objetivo general del sistema, mediante la descomposición en sub-objetivos.
- (v) Definir el contexto del sistema de software, mediante la producción de un conjunto de Escenarios Futuros, que representan el comportamiento proyectado en el contexto de aplicación donde funcionará el sistema de software a desarrollar [36].
- (vi) Evolucionar el Léxico Extendido del Lenguaje, creando el modelo Léxico Extendido del Lenguaje del Sistema, modificando términos existentes, eliminando términos obsoletos, y adicionando nuevos términos utilizados en las descripciones de Escenarios Futuros [37].
- (vii) Explicitar los requisitos, produciendo el documento de Especificación de Requisitos de Software, después de haber extraído los requisitos del conjunto de Escenarios Futuros [38].

La Figura 1(a) describe el proceso base de IR, como fue diseñado originalmente [20] con algunas mejoras posteriores [36], aunque independiente de las características del contexto de aplicación y del proyecto de software. Los principales sub-procesos, que se muestran en la Figura 1(a), se componen de otro nivel de procesos, algunos de los cuales pueden ser procesos atómicos o descomponerse en otros, como se muestra en la Figura 1(b). 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.

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

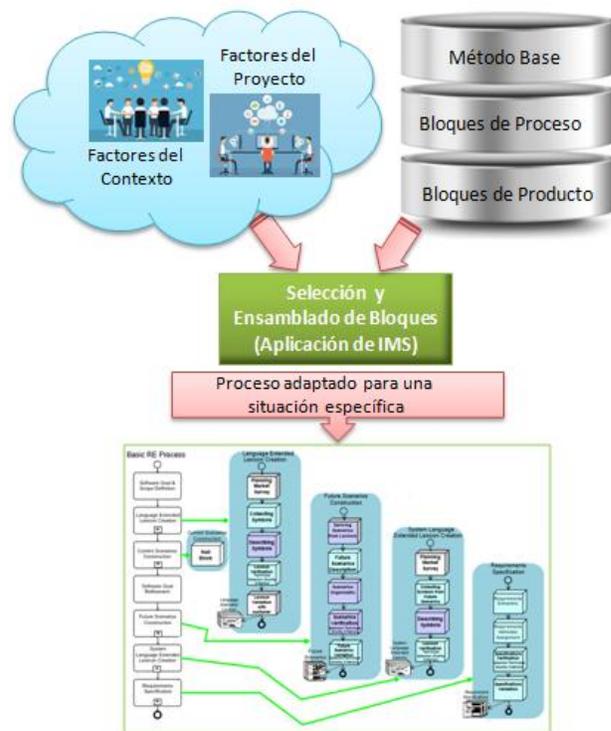
El modelo Léxico Extendido del Lenguaje representa el vocabulario utilizado en el contexto de aplicación, por lo que el modelo de Escenarios Actuales se describe utilizando los términos definidos en este glosario. El modelo Léxico Extendido del Lenguaje del Sistema representa un vocabulario lo más cercano posible al utilizado en el contexto de aplicación aunque considerando los términos introducidos para definir el futuro contexto con la incorporación del software a desarrollar [37]. Por ende, los modelos Escenarios

Futuros y Especificación de los Requisitos del Software 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.

### 3. Proceso de Ingeniería de Requisitos Adaptable a la Situación

La propuesta que se describe en el presente artículo se sustenta en proveer mecanismos que permitan adaptar en forma dinámica un proceso de IR para cada proyecto específico, ajustándolo a las situaciones particulares de su contexto, de modo flexible, ágil y repetible.

La Figura 2 ilustra cómo a partir de una situación dada, en función de factores establecidos, es posible aplicar guías y mecanismos de selección y ensamblado de bloques de proceso y de producto, y así construir un proceso específico para un proyecto particular.



**Figura 2. Obtención de un proceso adaptado a una situación.**

Tal como se mencionó anteriormente, existen dos tipos de factores situacionales: los relacionados al contexto de aplicación y aquellos asociados al proyecto de software. Los primeros tienen que ver con el contexto

del usuario, mientras que los segundos se relacionan con el contexto de desarrollo.

Siguiendo la propuesta de la IMS, en trabajos preliminares se han estudiado numerosos factores considerados en la literatura para distintas actividades de la IR [39, 40]. Como resultado del estudio realizado y a la luz de la experiencia en la aplicación de la estrategia, se han identificado 21 factores situacionales (ver Tabla 1) que podrían provocar algún tipo de adaptación en el proceso de IR.

**Tabla 1. Factores situacionales que impactan en el proceso de Ingeniería de Requisitos.**

#	Factores del Contexto	Tipo de Evolución del Factor
1	Complejidad del contexto	Invariante
2	Tipo de cliente	Invariante
3	Novedad del contexto	Invariante
4	Reingeniería del proceso del negocio	Contingente
5	Volatilidad del contexto	Invariante
6	Volatilidad de los requerimientos del cliente	Contingente
7	Rotación de usuarios	Invariante
8	Inconsistencias en el contexto	Invariante
9	Conflicto de intereses de usuarios	Contingente
#	Factores del Proyecto	Tipo de Evolución del Factor
10	Conocimiento previo del dominio	Contingente
11	Envergadura del proyecto	Contingente
12	Rotación del equipo desarrollador	Invariante
13	Calidad exigida en el software	Invariante
14	Reuso de artefactos de requisitos existentes	Invariante
15	Creación de artefactos de requisitos para reuso	Invariante
16	Pre-rastreabilidad de requisitos	Invariante
17	Post-rastreabilidad de requisitos	Invariante
18	Granularidad de rastreabilidad de los requisitos	Invariante
19	Exigencia de producir el documento Especificación de Requisitos de Software	Invariante
20	Restricciones de tiempo y recursos para el proyecto	Contingente
21	Modelo de proceso de software	Invariante

Desde la dimensión de la evolución, los factores pueden clasificarse en dos tipos: *Invariante*, aquel factor que no tiene cambios durante el proceso de IR, mientras que se considera *Contingente* al factor que puede evolucionar naturalmente debido a cambios en el contexto de aplicación y/o en el contexto del proyecto (ver Tabla 1).

A su vez, cada factor tiene un conjunto acotado de valores aceptables, tal como se muestran en la Tabla 2.

**Tabla 2. Valores posibles para los factores situacionales.**

# Factor Situacional	Valores Admisibles
1, 4 - 13, 20	{Muy Alto   Alto   Medio   Bajo   Muy Bajo}
2	{Cliente Específico   Mercado Potencial}
3, 14 - 17, 19	{Sí   No}
18	{Individual   Grupal}
21	{Cascada   Incremental   Iterativo   Ágil   Otro }

Tal como se ha mencionado anteriormente una situación no depende de un factor en particular sino que surge de la combinación del conjunto de factores con sus respectivos valores, por lo tanto, es relevante estudiar la influencia recíproca que puede existir entre algunos de ellos. Así es que se debe tener en cuenta la existencia de diferentes tipos de interacciones entre los factores, las que se deben considerar al momento de adaptar el proceso de IR:

- *Interacción invalidante* indica que un factor no debe ser tenido en cuenta cuando otro factor toma un valor específico.
- *Interacción limitante* señala que un factor puede restringir los valores que puede tomar otro factor.
- *Interacción incompatible* implica que dos o más factores no pueden aplicar simultáneamente a un mismo proyecto con determinados valores, entonces se requiere una negociación con los clientes para modificar el valor de uno o de todos los factores involucrados, a fin de subsanar la incompatibilidad.

En la Tabla 3 se presentan algunos ejemplos de cada tipo de interacción. Éstas pueden establecerse una vez que se ha evaluado cada factor situacional y se le ha asignado el valor correspondiente. Debe tenerse en cuenta que estos valores podrían modificarse durante el

avance del proceso y, por ende, dichas interacciones también podrían cambiar.

**Tabla 3. Ejemplos de interacciones entre factores situacionales.**

Tipo de Interacción	Interacciones
Invalidante	Si el valor de <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> y <i>Rotación de usuarios</i> .
Limitante	El factor <i>Modelo de proceso de software</i> no puede tomar el valor “Cascada” si la <i>Volatilidad del contexto</i> fuese “Alta” o “Muy Alta”
Incompatible	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: <i>Rotación de usuarios</i> es “Alta” o “Muy Alta”, o <i>Calidad exigida en el software</i> es “Alta” o “Muy Alta”, o <i>Exigencia de producir el documento ERS</i> es “Sí”.

Desde la perspectiva del nivel de confiabilidad, los valores de los factores se tipifican como:

- *Seguro*, cuando es posible establecer el valor del factor desde el inicio y este no cambia durante el proceso de IR.
- *Razonablemente seguro*, cuando el valor del factor puede cambiar de forma natural debido a cambios en el contexto de aplicación o en el proyecto, o debido a la influencia mutua con otro factor.
- *Dudoso*, cuando no es posible establecer el valor del factor con precisión, sino recién cuando se logra una mejor comprensión de la situación.

Entonces, se deberá evaluar cada factor y estimar su valor, otorgándole además su nivel de confiabilidad. Esto es necesario debido a que, tal como antes se ha mencionado, en algunos casos los factores no se conocen de manera precisa. Por ejemplo, los valores para los factores como *Volatilidad de los requerimientos del cliente* o *Conflictos de intereses de usuarios*, podrían no ser conocidos al iniciar el proceso. Aquellos factores cuyo nivel de confianza es *Dudoso* requerirán ser observados a medida que el proceso avanza, de manera

tal de establecer su valor con mayor precisión y, de ser necesario, re-adaptar el proceso de IR.

Además de identificar los factores situacionales, es necesario determinar de qué modo dichos factores, por sí solos o agrupados, podrán afectar el proceso de IR. En consecuencia, se han identificado cinco puntos de variación en el proceso base de IR, como se muestra en la Figura 1(a). Cada uno de estos puntos de variación se relaciona con la capacidad de un sub-proceso para ajustarse a una situación particular.

Existen algunas actividades del proceso de requisitos que no son afectadas por los factores situacionales, por lo cual, ellas se deberán realizar siempre, mientras que otras pueden ser modificadas, eliminadas o reemplazadas. No sólo las actividades están involucradas en estas adaptaciones, los modelos a generar o utilizar en el proceso suelen ajustarse también a la situación [11].

Cada sub-proceso se define por una secuencia de bloques de proceso, algunos de los cuales incluyen bloques de producto, los que son variaciones menores de los modelos del proceso base de IR.

En resumen, el proceso base de IR está compuesto por sub-procesos, donde algunos de ellos pueden ser descompuestos en otros niveles de sub-procesos. Se denominará *bloque de proceso básico* a aquel que es una unidad compacta que no puede ser descompuesta y que pertenece al proceso base de IR.

Al construir el proceso de IR para una situación específica, en cada punto de variación, un bloque de proceso básico puede ser: i) omitido; ii) reemplazado en su totalidad por un bloque de proceso específico; iii) reemplazado por un bloque de proceso con una variación interna parcial basada en parámetros; o iv) conservarse sin variación.

Un bloque de proceso reemplazado o parametrizado puede contener otro modelo, diferente al original. La combinación predefinida de factores situacionales que intervienen en cada punto de variación determina cuál de estas cuatro operaciones se debe llevar a cabo.

Por otra parte, hay cuatro tipos de bloques de proceso:

- *Independiente*, aquel que es autónomo, no depende de los factores situacionales y además no es compartido por otros sub-procesos, tales como el bloque *Describir Escenarios Futuros* que se muestra en la Figura 1(b).
- *Compartido*, aquel que es común a diferentes sub-procesos del proceso de IR, pudiendo depender de factores situacionales. Por ejemplo, el bloque de proceso *Organizar Escenarios* es compartido por dos sub-procesos principales: *Construir Escenarios Actuales* y *Construir Escenarios Futuros*.
- *Variante*, aquel que es un bloque que pertenece a un único sub-proceso específico y depende de

factores situacionales. Es el caso del bloque de proceso *Bosquejar Escenarios Futuros* que aparece en la Figura 3, el cual puede ser reemplazado por uno de cinco diferentes bloques de proceso, de acuerdo al valor de los factores situacionales involucrados.

- *Nulo*, es un bloque auxiliar utilizado cuando un sub-proceso (en cualquier nivel) debe suprimirse o saltarse en el proceso base de IR, entonces cada bloque de proceso a omitir se sustituye por el bloque nulo.

Cabe mencionar que un bloque parametrizable puede ser un bloque independiente o un bloque compartido. Por ejemplo, el bloque *Validar Escenarios Futuros* del sub-proceso *Construir Escenarios Futuros* (ver Figura 3) es un bloque independiente, mientras que el bloque *Verificar Escenarios* es compartido por los sub-procesos *Construir Escenarios Actuales* y *Construir Escenarios Futuros*, cuyo parámetro es también el nivel de calidad exigida.

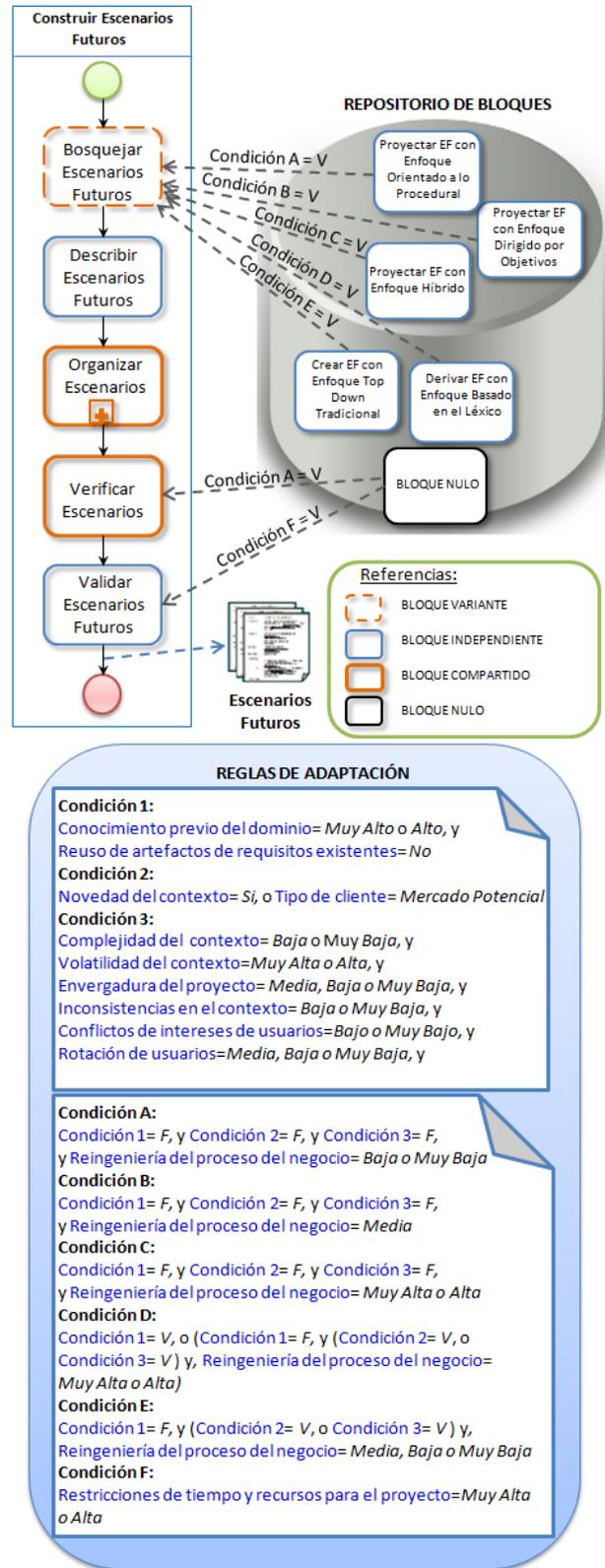
La Tabla 4 muestra las operaciones aplicables a los bloques de proceso básicos; no se considera el bloque nulo debido a que no forma parte del proceso base.

**Tabla 4. Operaciones válidas en los bloques de proceso básicos.**

Tipo de Bloque	OPERACIONES			
	Omitir	Reemplazar	Parametrizar	Conservar
Independiente	✓		✓	✓
Compartido	✓		✓	✓
Variante	✓	✓		

El proceso base de IR se adapta en los puntos de variación mostrados en la Figura 1(a), dependiendo de una combinación de factores situacionales específica para cada punto de variación. Un punto de variación se considera un hito para revisar el proceso de IR adaptado, y reformularlo de ser necesario.

La Figura 3 muestra las variantes que se presentan para el sub-proceso *Construir Escenarios Futuros* del proceso base de IR, donde se ejemplifica el mecanismo para la adaptación en base a los valores que puedan adoptar los factores de situación asociados al punto de variación 3 de la Figura 1(a). En la parte inferior de la Figura 3, se muestran las reglas de adaptación para este punto de variación, las cuales indican las combinaciones de factores que se aplican en cada parte del sub-proceso, y que determinan las operaciones a aplicar en cada bloque básico de este sub-proceso.



**Figura 3. Sub-proceso de construcción de escenarios futuros.**

El conjunto de reglas de adaptación establecido para cada punto de variación ha sido definido inicialmente en función de la experiencia adquirida en la aplicación del proceso base de IR y ha sido ajustado posteriormente al probar dichas reglas en algunos casos testigo, donde se obtuvieron buenos resultados, aunque se planifican ulteriores revisiones.

El sub-proceso *Construir Escenarios Futuros* depende especialmente del factor *Reingeniería del proceso de negocio*, junto con algunas condiciones de arrastre de sub-procesos anteriores (ver Condiciones 1, 2 y 3 en la parte inferior de la Figura 3). Estas condiciones de arrastre determinan si se crearon previamente el Léxico Extendido del Lenguaje y/o los Escenarios Actuales, entonces, una vez evaluadas las mismas, el bloque variante *Bosquejar Escenarios Futuros* se reemplaza por uno de los cinco bloques de proceso que permitan esbozar los escenarios según los siguientes enfoques:

- Derivándolos directamente de los Escenarios Actuales existentes: enfoque orientado a lo procedural.
- Aplicando un enfoque dirigido por los objetivos fijados para el sistema de software, apoyándose en los Escenarios Actuales de alto nivel.
- Combinando las dos estrategias anteriores, se trata de un enfoque híbrido.
- Aplicando la técnica de derivación de escenarios a partir del Léxico Extendido del Lenguaje.
- Utilizando un enfoque top-down en el marco de los objetivos del sistema de software, generando inicialmente escenarios de alto nivel.

Los bloques *Verificar Escenarios* y *Validar Escenarios Futuros* en la Figura 3 pueden ser omitidos según condiciones específicas para cada uno, mientras que el bloque *Describir Escenarios Futuros* se conserva independientemente de los valores de los factores situacionales imperantes.

#### 4. Proceso de Adaptación Dinámica del Proceso de Ingeniería de Requisitos

La adaptación del proceso base de IR implica la asignación de valores a cada factor situacional, estableciendo su nivel de confianza. Por otra parte, se debe disponer de un repositorio que contenga cada bloque 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 bloques u omitirlos o asignar valores a los parámetros de entrada de los bloques parametrizables. Por lo tanto, el proceso de adaptación del proceso de IR para una

situación específica involucra las actividades que se detallan a continuación (ver Figura 4):

- **Evaluar los factores situacionales:** Se evalúan 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. En esta fase se toman en consideración las posibles interacciones entre los factores. La información se registra en un formulario de factores situacionales (ver Figura 5), el cual brinda una guía sobre los valores admitidos para cada factor, como así también, permite identificar interacciones entre factores en función de los valores asignados a otros factores.

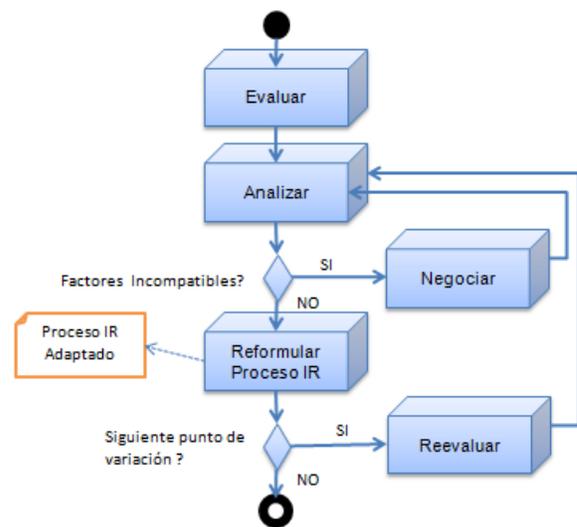


Figura 4. Proceso de adaptación del proceso de IR en función de factores situacionales.

- **Analizar los factores situacionales:** Se analizan los valores asignados a los factores teniendo en cuenta, en especial, las interacciones y aquellos factores con bajo nivel de confianza. Al detectarse factores cuyo nivel de confianza es *Dudoso*, se debe adoptar una posición. Bajo una posición optimista, se asignan a estos factores un valor *Muy Bajo* (o *Sí*), sin tener en cuenta el valor asignado originalmente. Bajo una posición conservadora, los factores en cuestión toman un valor *Muy Alto* (o *No*). 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. En caso de identificar interacciones

incompatibles entre factores, se realiza la actividad de negociación.

FACTORES SITUACIONALES			
Proyecto: _____		Fecha: ___/___/___	
Responsable: _____			
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 Especifico <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3. Novedad del contexto Si <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 Especifico") Muy alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Bajo <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 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 Especifico") 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 Especifico") Muy alta <input type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja <input type="checkbox"/> Muy Bajo <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FACTORES DEL PROYECTO	NIVEL DE CONFIANZA		
	Seguro	Razonablemente Seguro	Dudoso
10. 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"/>
11. Envergadura del proyecto 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"/>
12. Rotación de 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"/>
13. Calidad exigida en el 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"/>
14. Reuso de artefactos de requisitos existentes Si <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15. Creación de artefactos de requisitos para reuso Si <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
16. Pre-Rastreabilidad de los requisitos Si <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
17. Post-Rastreabilidad de los requisitos Si <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
18. Granularidad de Rastreabilidad de los requisitos (Completar sólo si Pre-Rastreabilidad es "Si" o Post-Rastreabilidad es "Si") Individual <input type="checkbox"/> Grupal <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
19. Exigencia de producir el documento ERS Si <input type="checkbox"/> No <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20. 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"/>
21. Modelo de proceso de software Cascada <input type="checkbox"/> Incremental <input type="checkbox"/> Iterativo <input type="checkbox"/> Ágil <input type="checkbox"/> Otro <input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 5. Formulario de registro de valores de factores situacionales.

- **Negociar incompatibilidades:** Esta actividad se realiza cuando se detectan interacciones de tipo incompatibles entre dos o más factores. El objetivo de la negociación es llegar a un acuerdo entre las partes interesadas para reasignar valores a uno o varios de estos factores para solucionar la incompatibilidad. Después de la negociación, se debe hacer un análisis parcial de los factores.

- **Reformular el proceso de IR:** Esta actividad consiste en armar el proceso de IR o re-armarlo si hubo nuevas evaluaciones de los factores. La reformulación del proceso se realiza a partir del proceso base de IR, siguiendo un conjunto de *reglas de adaptación* para aplicar operaciones sobre cada bloque de proceso en cada punto de variación no alcanzado aún. Dichas reglas involucran una combinación específica de factores situacionales afectados en cada punto de variación (ver Tabla 5). El armado o re-armado consiste entonces en mantener, eliminar, sustituir o asignar valores a parámetros de bloques de proceso básicos, obtenidos de un repositorio que contiene todos los bloques de proceso.

No se ha presentado en este artículo todo el conjunto de reglas de adaptación debido al excesivo espacio que ello requeriría y, además, un registro enciclopédico de las mismas no aportaría un mayor conocimiento sobre el núcleo del tema tratado aquí. Solo a modo de ejemplo, en la parte inferior de la Figura 3 se muestran las reglas de adaptación a aplicar en el punto de variación 3, las que involucran los factores situacionales marcados en la Tabla 5 en la columna correspondiente a dicho punto de variación.

- **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. También se deben revisar aquellos factores que se consideran *Contingentes* a fin de establecer si se produjo algún cambio en la situación. Si el valor de algún factor se modifica, se debe actualizar el formulario de factores situacionales y se realiza nuevamente la actividad de análisis.

**Tabla 5. Factores situacionales involucrados en cada punto de variación.**

# Factor Situacional	Punto Var. 1	Punto Var. 2	Punto Var. 3	Punto Var. 4	Punto Var. 5
1		✓	✓		
2	✓	✓	✓	✓	
3	✓	✓	✓	✓	
4			✓	✓	
5		✓	✓		
6		✓			✓
7		✓	✓		
8		✓	✓		
9		✓	✓		
10	✓	✓	✓	✓	
11		✓	✓		✓
12		✓	✓		
13	✓	✓	✓	✓	✓
14	✓	✓	✓	✓	✓
15	✓	✓	✓	✓	✓
16	✓	✓	✓	✓	
17					✓
18					✓
19					✓
20	✓	✓	✓	✓	✓
21	✓	✓	✓	✓	✓

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 4.

## 5. Conclusión

La propuesta presentada trata con la variabilidad de un proceso de IR en particular, impulsado por una adaptación dinámica basada en factores de situación típicos.

La mayoría de los problemas tienen características distintivas que deben ser tomadas en cuenta para llevar a cabo un proceso de requisitos con éxito. Se debería adaptar el proceso de requisitos mediante la selección de las técnicas más adecuadas para situaciones específicas. Proporcionar guías sobre cómo adaptar el proceso de requisitos, hace probable que se piense en la adaptación antes de iniciarlo y no como medida de mitigación según vayan acaeciendo contingencias.

El proceso para la construcción de un proceso de IR consiste en un conjunto de factores que tipifica las situaciones más probables, teniendo en cuenta tanto el

contexto de aplicación como el contexto del proyecto de software. Este proceso de adaptación establece cuándo el proceso de IR debe ser revisado y re-adaptado sobre la base de reglas y operaciones en cada punto de variación. Dado que con frecuencia se observa la evolución natural y contingente de situaciones, es importante tener la posibilidad de redefinir las actividades restantes del proceso de IR mientras el proyecto está en curso.

Jafarinezhad y Ramsin [17] han propuesto un modelo situacional utilizando el lenguaje de modelado i\* con una cantidad similar de factores a la utilizada en la presente propuesta (y una buena coincidencia en los mismos), que permiten la selección de actividades concretas en base a fases, etapas y tareas de un proceso genérico de IR. En cuanto al grado de confiabilidad en los valores de los factores situacionales, los manejan aplicando lógica difusa al crear el proceso de IR, sin considerar una adaptación dinámica del mismo.

Los factores situacionales se han estudiado de forma exhaustiva en la literatura, aunque este conjunto podría extenderse. Hay algunos factores en estudio, tales como equipos distribuidos, diferencias culturales e idiomáticas, distancias geográficas y temporales, provenientes en gran parte del desarrollo global de software [41-43], los que podrían ampliar esta lista. Asimismo, se estudiarán los factores que influyen en las fases posteriores del proceso de desarrollo de software para determinar si podrían realimentarse sobre el proceso de IR.

La descomposición del proceso de IR en bloques de proceso básicos, con un conjunto de factores de situación, ha permitido la identificación y, por lo tanto, la definición de los diferentes tipos de bloques de proceso necesarios, estableciendo cuáles se ejecutan siempre y cuáles son variables. Está previsto evaluar la conveniencia de utilizar bloques mixtos, o bloques de proceso separados de bloques de producto. Se considera también que se deben llevar a cabo más casos para confirmar los diferentes flujos posibles del proceso adaptable.

## 6. Referencias

- [1] Haumer, P., *IBM Rational Method Composer: Part 1: Key concepts*, IBM developerWorks, Diciembre 2005. Accedido el 30/07/2016 en: <http://www.ibm.com/developerworks/rational/library/dec05/haumer/index.htm>.
- [2] Firesmith, D., "Creating a project-specific requirements engineering process", *Journal of Object technology*, 3, 5, 2004, pp. 31-44.
- [3] Kumar, K., y Welke, R.J., "Methodology Engineering: a proposal for situation-specific methodology construction", en W.W.Cotterman & J.A.Senn (Eds.), *Challenges and strategies for research in systems development*, John Wiley & Sons, 1992, pp. 257-269.

- [4] Khan, H.H., bin Mahrin, M.N. y bt Chuprat, S., "Factors for Tailoring Requirement Engineering Process: A Review", *International Journal of Software Engineering and Technology*, 1, 1, 2014, pp. 7-18.
- [5] Carrizo, D., *Marco para la selección de técnicas para educación de requisitos*, Universidad Politécnica de Madrid, Madrid, España, Tesis Doctoral, 2009.
- [6] Pol'la, M., Buccella, A., Cechich, A., y Arias, M., "Un modelo de metadatos para la gestión de la variabilidad en líneas de productos de software", en *Memorias de XLIII Jornadas Argentinas de Informática e Investigación Operativa (43JAIIO) - XV Simposio Argentino de Ingeniería de Software*, Buenos Aires, 2014.
- [7] Cetina, C., Giner, P., Fons, J., y Pelechano, V., "Autonomic computing through reuse of variability models at runtime: The case of smart homes", *Computer*, IEEE Computer Society, 42, 10, Octubre 2009, pp. 37-43.
- [8] Santos, E., Castro, J., Sánchez, J. y Pastor, O., "A Goal-Oriented Approach for Variability in BPMN", en *Proceedings of the 13th Workshop on Requirements Engineering*, 2010, pp. 17-28.
- [9] Schnieders, A., y Puhlmann, F., "Variability Mechanisms in E-Business Process Families", en *Proceedings of International Conference on Business Information Systems (BIS 2006)*, 2006, pp. 583-601.
- [10] Hallerbach, A., Bauer, T., y Reichert, M., "Managing Process Variants in the Process Lifecycle", en *Proceedings of 10th International Conference on Enterprise Information Systems*, Barcelona, España, Junio 2008, pp. 154-161.
- [11] Galster, M., Weyns, D., Tofan, D., Michalik, B., y Aygeriou, P., "Variability in Software Systems - A Systematic Literature Review", *IEEE Transactions on Software Engineering*, IEEE, 40, 3, 2014, pp. 282-306.
- [12] Rolland, C., "Method engineering: towards methods as services", en *Making Globally Distributed Software Development a Success Story*, Springer Berlin Heidelberg, 2008, pp. 10-11.
- [13] Brinkkemper, S., "Method Engineering: Engineering of Information Systems Development Methods and Tools", *Information and Software Technology*, 38, 4, 1996, pp. 275-280.
- [14] Henderson-Sellers, B., y Ralyté, J., "Situational Method Engineering: State-of-the-Art Review", *Journal of Universal Computer Science*, 16, 3, 2010, pp. 424-478.
- [15] Ralyté, J., "Situational Method Engineering in Practice: A Case Study in a Small Enterprise", en *Proceedings of 25th International Conference on Advanced Information Systems Engineering*, Valencia, España, 2013, pp.17-24.
- [16] Bucher, T., Klesse, M., Kurpjuweit, S., y Winter, R., "Situational Method Engineering", en *Situational method engineering: fundamentals and experiences*, Springer US, 2007, pp. 33-48.
- [17] Jafarinezhad, O., y Ramsin, R., "Development of Situational Requirements Engineering Processes: A Process Factory Approach", en *Proceedings of 36th IEEE International Conference on Computer Software and Applications*, 2012, pp.279-288.
- [18] Bakhat, K.A., Sarwar, A.A., Motla, Y.H.B., y Akhtar, M.C., "A Situational Requirement Engineering Model for an Agile Process", *Bahria University Journal of Information & Communication Technology*, 8, 1, 2015, pp. 21-26.
- [19] Macaulay, L., "Requirements capture as a cooperative activity", en *Proceedings of IEEE International Symposium on Requirements Engineering*, IEEE, 1993, pp. 174-181.
- [20] Leite, J.C.S.P., Doorn, J.H., Kaplan, G.N., Hadad, G.D.S., y Ridao, M.N., "Defining System Context using Scenarios", en *Perspectives on Software Requirements*, Kluwer Academic Publishers, EEUU, capítulo 8, 2004, pp.169-199.
- [21] Kaindl, H., "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, 2000, pp. 537-551.
- [22] Leffingwell, D., y Widrig, D., *Managing Software Requirements - A unified approach*, Addison-Wesley Object Technology Series, 2da. edición, 2003.
- [23] Antonelli, L., Rossi, G., Leite, J.C.S.P., y Oliveros, A., "Deriving requirements specifications from the application domain language captured by Language Extended Lexicon", en *Proceedings of XV Workshop on Requirements Engineering*. Buenos Aires, Argentina: Universidad Nacional de La Matanza, 2012.
- [24] Potts, C., "Using Schematic Scenarios to Understand User Needs", en *Proceedings of Symposium on Designing Interactive Systems: Processes, Practices and Techniques*, 1995, pp.247-256.
- [25] Leite, J.C.S.P., Hadad, G.D.S., Doorn, J.H., y Kaplan, G.N., "A Scenario Construction Process", *Requirements Engineering Journal*, 5, 1, 2000, pp.38-61.
- [26] Cockburn, A., *Writing Effective Use Cases*, Addison-Wesley Professional, 2000.
- [27] Seyff, N., Maiden, N., Karlsen, K., Lockerbie, J., Grünbacher, P., Graf, F., y Ncube, C., "Exploring how to use scenarios to discover requirements", *Requirements Engineering Journal*, 14, 2, 2009, pp. 91-111.
- [28] Maiden, N., y Rug, G. ACRE, "Selecting methods for requirements acquisition", *Software Engineering Journal*, 11, 3, 1996, pp. 183-192.
- [29] Hickey, A.M., y Davis, A.M., "Elicitation technique selection: how do experts do it?", en *Proceedings of 11th IEEE International Requirements Engineering Conference*, IEEE, 2003, pp. 169-178.
- [30] Coulin, C.R., *A Situational Approach and Intelligent Tool for Collaborative Requirements Elicitation*, University of Technology, Sydney, Australia, Tesis Doctoral, 2007.

- [31] Carrizo, D., Dieste, O., y Juristo, N., "Study of elicitation techniques adequacy", en *Proceedings of 11th Workshop on Requirements Engineering*, 2008, pp. 104-114.
- [32] Lauesen, S., *Software Requirements: Styles and Techniques*, Addison-Wesley, 2002.
- [33] Lobo, L.O., y Arthur, J.D., "An objectives-driven process for selecting methods to support requirements engineering activities", en *Proceedings of 29th Annual IEEE/NASA Software Engineering Workshop*, IEEE, 2005, pp. 118-130.
- [34] Alexander, I.F., y Beus-Dukic, L., *Discovering Requirements: How to Specify Products and Services*. Chichester, West Sussex, England: John Wiley & Sons, 2009.
- [35] Hadad, G. D., Doorn, J. H., y Kaplan, G. N., "Creating Software System Context Glossaries", en M. Khosrow-Pour (Ed.), *Encyclopedia of Information Science and Technology*, Second Edition, Hershey, PA: IGI Global, 2009, pp.789-794.
- [36] Hadad, G.D.S., *Uso de Escenarios en la Derivación de Software*, Universidad Nacional de La Plata, Argentina, Tesis Doctoral, 2008.
- [37] Kaplan, G. N., Doorn, J. H., y Gigante, N., "Evolución semántica de glosarios en los procesos de requisitos", en *Memorias del XVIII Congreso Argentino de Ciencias de la Computación*, Mar del Plata, Argentina, 2013.
- [38] Hadad, G.D.S., Doorn, J.H., y Kaplan, G.N., "Explicitar Requisitos del Software usando Escenarios", en *Proceedings of 12th Workshop on Requirements Engineering*, Universidad Técnica Federico Santa María, Valparaíso, Chile, Julio 2009, pp.63-74.
- [39] Hadad, G.D.S., y Doorn, J.H., "Introducing Variability in a Client-Oriented Requirements Engineering Process", en *Proceedings of ER@BR2013 - Requirements Engineering @ Brazil*, Pontificia Universidade Católica do Rio de Janeiro, ISSN: 1613-0073, Río de Janeiro, Brasil, Julio 2013, pp.8-13.
- [40] Hadad, G.D.S., Ledesma, V.A., y Doorn, J.H., "Proceso de Requisitos Adaptable a Factores Situacionales", en *Memorias de XVI Workshop de Investigadores en Ciencias de la Computación*, Universidad Nacional de Tierra del Fuego, Antártida e Islas del Atlántico Sur, Ushuaia, 2014, pp.448-452.
- [41] ul Haq, S., Raza, M., Zia, A., y Khan, M.N.A., "Issues in global software development: A critical review", *Journal of Software Engineering and Applications*, 4, 10, 2011, pp. 590-595.
- [42] Richardson, I., Casey, V., McCaffery, F., Burton, J., y Beecham, S., "A Process Framework for Global Software Engineering Teams", *Information and Software Technology*, 54, 11, 2012, pp. 1175-1191.
- [43] Mighetti, J.P., y Hadad, G.D.S., "Proceso de requisitos adaptado para mitigar amenazas en el Desarrollo Global de Software", en *Memorias de 3er Congreso Nacional de Ingeniería Informática / Sistemas de Información*, Universidad Tecnológica Nacional, Facultad Regional Buenos Aires, 2015.