



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

“Un Enfoque de Priorización de Requerimientos, a
partir de la Segmentación de las Preferencias de los
Stakeholders”.

Autor: Carlos Martín Azzolini

Directora

Dra. Isabel Passoni

Universidad Nacional de Mar del Plata

Co-Director

Mg. Pablo Thomas

Universidad Nacional de La Plata

*Tesis presentada para obtener el grado de Magister en Ingeniería de Software. Facultad de
Informática Universidad Nacional de La Plata.*

Noviembre de 2011

Resumen

La presente Investigación tiene como propósito general diseñar un modelo de priorización de requerimientos de software a partir de la segmentación de preferencias de los stakeholders. Para ello, se propone fusionar una serie de enfoques teóricos de la priorización de requerimientos, del tipo: Negociación-Cognitivo para luego integrarlos metodológicamente con algunos elementos propios de la Inteligencia Computacional (SOM de Kohonen), el Proceso de Jerarquía Analítica (AHP) y métricas de la Psicología Cognitiva (DF, VTV y RE). El Modelo propuesto, pretende establecer un Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU) que sea consensuado y validado por el total de stakeholders del caso experimental "Subsistema de Flujo de Trabajo" del Centro de Investigación en Procesos Básicos, Metodología y Educación (CIMEPB) de la Facultad de Psicología de la Universidad Nacional de Mar del Plata.

Abstract

This general research aims to design a model for prioritizing software requirements from the segmentation of preferences of stakeholders. We propose to merge a number of theoretical approaches to prioritization of requirements, such as: negotiation, Cognitive and then methodically integrate some elements of Computational Intelligence (Kohonen SOM), the Analytical Hierarchy Process (AHP) and Cognitive Psychology metrics (DF, VTV and RE). The proposed model seeks to establish a General Scheme for Priority deployable User Requirements (GSPDUR) that is agreed and validated by the stakeholders of "Workflow Subsystem" of the Centre for Research in Basic Processes, Methodology and Education (CIMEPB), Faculty of Psychology, National University of Mar del Plata.

Tabla de Contenidos

	Pagina
Resumen de la Tesis.....	02
Prologo.....	11
Introducción General.....	14
 Capítulo 1 “El Contexto de la Priorización de Requerimientos de Software y su justificación como temática de estudio”	
1.1 Introducción.....	17
1.2 Contexto Disciplinario de la Priorización de Requerimientos de Software.....	18
1.3 La Priorización de Requerimientos y su relación con el Problema de la Próxima versión (NRP).....	22
1.4 ¿Qué beneficios traerá aparejado la implementación de un Proceso de Priorización de Requerimientos?.....	25
1.5 Consideraciones Finales del Capítulo.....	26
 Capítulo 2: “Estado de Arte: “Priorización de Requerimientos de Software”	
2.1. Introducción.....	27
2.2 Aspectos de la Priorización de Requerimientos.....	28
2.3 Priorización de Requerimientos: Enfoque Cognitivo.....	31
2.4 Priorización de Requerimientos: Enfoque Negociación.....	35
2.5 Priorización de Requerimientos: Enfoque Métodos.....	46
2.6 Consideraciones Finales del Capítulo.....	52
 Capítulo 3: “Marco Teórico y Metodológico de la Investigación”	
3.1. Introducción.....	53
3.2 Método para el relevamiento de preferencias de implementación de los Requerimientos de Usuarios.....	54
3.3 Elementos de la Psicología Cognitiva aplicados en la segmentación de preferencias de implementación de Requerimientos de Usuarios.....	60
3.4 Técnicas para el Agrupamiento de preferencias de implementación de Requerimientos de Usuarios.....	67
3.5 Proceso de Negociación aplicado en Grupos de <i>stakeholders</i>	76

	Pagina
3.6 Elementos de la Teoría de Conjuntos Difusos.....	81
3.7 Consideraciones Finales del Capítulo.....	93

Capítulo 4: “Modelo de Priorización de Requerimientos de Usuario a partir de la Segmentación de Preferencias de los Stakeholders (MPRUSPS)”

4.1. Introducción.....	94
4.2 Relevamiento de los conceptos claves del dominio.....	97
4.3 Relevamiento de las Producciones Semánticas de los conceptos claves del Dominio.....	98
4.4 Análisis Métrico de las Producciones Semántica de los conceptos claves del Dominio.....	99
4.5 Relevamiento de las preferencias de implementación de los Requerimientos de Usuarios.....	100
4.6 Segmentación (agrupamiento) de las Preferencias de los Stakeholders.....	102
4.7 Identificación de los Esquemas Particulares de Prioridad Implementable de Requerimientos de Usuarios (EPPIRU).....	104
4.8 Determinación del Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU).....	106
4.9 Comparación entre el Esquema General (EGPIRU) y cada Esquema Particular (EPPIRU). Análisis de la Implementación de un Proceso de Negociación Grupal.....	108
4.10 Consideraciones Finales del Capítulo.....	116

Capítulo 5: “Experimentación: Ejemplo de aplicación del Modelo de Priorización de Requerimientos de Usuarios a partir de la Segmentación de Preferencias de los Stakeholders (MPRUSPS)”

5.1 Introducción.....	117
5.2 Descripción general del contexto de aplicación del MPRUSPS.....	118
5.3 Relevamiento de las preferencias de implementación de los Requerimientos de Usuarios.....	119
5.4 Segmentación (agrupamiento) de las Preferencias de los stakeholders.	121
5.5 Identificación de Esquemas Particulares de Prioridad Implementable de Requerimientos de Usuarios (EPPIRU).....	124

	Página
5.6 Determinación del Esquema General de Prioridad Implementable Requerimientos de Usuarios (EGPIRU).....	126
5.7 Comparación entre el Esquema General (EGPIRU) y cada Esquema Particular (EPPIRU). Análisis de la Implementación de un Proceso de Negociación Grupal.....	130
5.8 Análisis del Caso con un Método de Priorización alternativo: Karlsson (1996).....	133
5.9 Consideraciones finales del Capítulo.....	136

Capítulo 6: “Caso Experimental: Subsistema de Flujo de Trabajo del CIMEPB”

6.1. Introducción.....	138
6.2 Alcance del Flujo de Trabajo (Workflow).....	139
6.3 Descripción General.....	144
6.4 Especificación de los Requerimientos Funcionales del Subsistema de Flujo de Trabajo (SFT).....	148
6.5 Trazabilidad Enriquecida: Relación entre los Requerimientos de Usuarios y los Requerimientos del Sistema.....	150
6.6 Particularidades del Proceso de Elicitación y Validación de los Requerimientos Funcionales del SFT.....	155
6.7 Consideraciones Finales del Capítulo.....	157

Capítulo 7: “Resultados de la aplicación del MPRUSPS en el Caso Experimental: Subsistema de Flujo de Trabajo del CIMEPB”

7.1. Introducción.....	158
7.2 Etapas del MPRUSPS y Software a utilizar en el procesamiento de datos.....	159
7.3 Análisis de los Resultados.....	160
7.4 Consideraciones Finales del Capítulo.....	182

Capítulo 8: “Discusión general respecto de la implementación del MPRUSPS”

8.1. Introducción.....	183
8.2 Discusión sobre los aspectos metodológicos del MPRUSPS.....	184
8.3 Restricciones a considerar a la hora de aplicar el MPRUSPS.....	187
8.4 Consideraciones finales del Capitulo.....	189

Capítulo 9: “Conclusión, Recomendaciones y Futuros Trabajos”

9.1 Conclusión.....	190
9.2 Recomendaciones y Futuras Trabajos.....	193

Bibliografía Utilizada.....	196
------------------------------------	------------

Índice de Figuras

Página

Figura 1.1: Situación contextual de la Priorización de Requerimientos.....	19
Figura 1.2: “Área de Conocimiento: “Software Requirements”	20
Figura 2.1: “Fusión de la Teoría W y Modelo Espiral”	36
Figura 2.2: “Preferencia de los Requerimientos en base al Esfuerzo de Desarrollo”	37
Figura 2.3: “Análisis de Preferencia Multicriterio”	38
Figura 2.4: “Constelaciones de Negociación”	40
Figura 2.5: “Mapas de Constelaciones”	41
Figura 2.6: “Enfoque ASTN (A Socio-Technical Negotiation approach)”	43
Figura 2.7: “Pasos involucrados en el MDRPM”	50
Figura 2.8: “Pasos involucrados en AHP Engine”	51
Figura 3.1: “Implicancias de la emisión de un juicio de Preferencias”	63
Figura 3.2: “Mapa Bidimensional en SOM”	71
Figura 3.3: “Ejemplo de Matriz U”	73
Figura 3.4: “Modelo ABNP”	77
Figura 3.5 “Estructura del Proceso de Negociación”	79
Figura 3.6: “Ejemplos de Conjuntos Difusos Convexos”	83
Figura 3.7: “Función de Pertenencia tipo Triangular”	87
Figura 3.8: “Función de Pertenencia tipo traapezoidal”	88
Figura 3.9: “Función de Pertenencia tipo sigmoideal”	88
Figura 3.10: “Función de Pertenencia tipo gaussiana”	89
Figura 3.11: “Sistema de Inferencia Difuso”	90
Figura 4.1: “Arquitectura del MPRUSPS”	95
Figura 4.2 “Posibles grados de alineamiento del grupo tras la comparación entre su EPPIRU y el EGPIRU”	114
Figura 5.1 “Matriz unificada de distancia para el Caso propuesto en 5.3”	121
Figura 5.2 “Índice de Davies-Bouldin para el Caso propuesto en 5.3”	122
Figura 5.3 “Composición de los Grupos”	123
Figura 6.1: “Estructura del MoReq2: Procesos y Estándares”	143
Figura 6.2 “Jerarquía de Usuarios del SIA- CIMEPB- WEB”	146
Figura 6.3: “Casos de Uso del SFT”	147
Figura 6.4: “Trazabilidad Enriquecida del primer Requerimiento de Usuario”	151
Figura 6.5: “Trazabilidad Enriquecida del segundo Requerimiento de Usuario”	152
Figura 6.6: “Trazabilidad Enriquecida del tercer Requerimiento de Usuario”	153
Figura 6.7: “Trazabilidad Enriquecida del cuarto Requerimiento de Usuario”	154
Figura 7.1: “Determinación de los Conceptos Claves del Dominio”	162

	Página
Figura 7.2: “Producción Semántica de Software”	163
Figura 7.3: “Producción Semántica de Investigación”	164
Figura 7.4: “Producción Semántica de Trabajo Colaborativo”	165
Figura 7.5: “Producción Semántica de Interdisciplina”	166
Figura 7.6: “Matriz de Distancia Unificada”	169
Figura 7.7: “Identificación de grupos y rasgos característicos”	170
Figura 7.8: “Análisis de Grupos mediante el Índice Davies-Bouldin.....	172
Figura 7.9: “Composición de los Grupos”	173
Figura 7.10: “Funciones de distribución de preferencia de implementación: Grupo 3 versus Nivel General”	180

Índice de Tablas

	Página
Tabla 1: “Escala de Priorización de Requerimientos Ordinal-Conceptual”	46
Tabla 2: “Escala utilizada para la valorización de Requerimientos de Software”	57
Tabla 3: “Valores de Verdad”	81
Tabla 4: “Ejemplo del cuadro de Síntesis de las Producciones Semánticas de cada uno de los conceptos claves del dominio (SPSCCD)”	99
Tabla 5: “Ejemplo de la Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios (LFDPIRU).....	101
Tabla 6: “Ejemplo de la Tabla: Grupos y Composición (GyC)”	103
Tabla 7: “Ejemplo de Lista de Grupos con sus correspondientes Valores Medio de Preferencia de Implementación de Requerimientos de Usuarios (LGVMPRU)”	105
Tabla 8: “Ejemplo del cuadro de Síntesis de las Producciones Semánticas de los conceptos claves del dominio por Grupo (SPSCCDG)”	106
Tabla 9 “Ejemplo para la determinación de las preferencias acumuladas en el EGPIRU”	108
Tabla 10 “Ejemplo: cálculo del peso relativo de los primeros requerimientos de un EPPIRU”	109
Tabla 11 “Ejemplo del Grado de Coincidencia entre el EGPIRU y un EPPIRU”.....	109
Tabla 12: “Grado de Alineamiento del Grupo (GAG) en función del comportamiento de las variables de entradas”	113
Tabla 13: “Ejemplo de Lista de Grupos Clasificados según Categoría de Alineamiento (LGCCA)”	115
Tabla 14: “Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios (LFDPIRU).....	120
Tabla 15: “Preferencia Acumulada en tramos de Requerimientos”	124
Tabla 16: “EPPIRU asignado por Grupo”	125
Tabla 17: “Síntesis de las Producciones Semánticas de los conceptos claves del dominio por Grupo (SPSCCD) y en diferentes escenarios”	126
Tabla 18 “EGPIRU resultante en el Escenario 1”	127
Tabla 19: “EGPIRU resultante en el Escenario 2”	129
Tabla 20: “Lista de Grupos Clasificados según Categoría de Alineamiento (Escenario 1)”	130
Tabla 21: “Lista de Grupos Clasificados según Categoría de Alineamiento (Escenario 2)”	132

	Página
Tabla 22 “Orden de Prioridad de los Requerimientos de Usuarios según el modelo de Karlsson (1998)”	134
Tabla 23: “Lista de Grupos Clasificados según Categoría de Alineamiento (Método de Karlsson)”	135
Tabla 24: “Lista de Lista Tentativa de Conceptos claves del dominio. (LTCCD)”	160
Tabla 25: “Distribución de los votos para los Conceptos Claves del Dominio”	161
Tabla 26: “Síntesis de la Producción Semántica de los Conceptos claves del Dominio según total de stakeholders”	167
Tabla 27: “Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios”	168
Tabla 28: “Tabla de Grupos y Composición (GyC)”	174
Tabla 29: “Lista de Grupos con sus correspondientes Valores Medio de Preferencia de Implementación de Requerimientos de Usuarios (LGVMPRU)”	175
Tabla 30: “Síntesis de las Producciones Semánticas de los conceptos claves del Dominio por Grupo (SPSCCD)”	176
Tabla 31: “Lista de Grupos Clasificados según Categoría de Alineamiento”	178
Tabla 32: “Síntesis de datos del Grupo 3”	179
Tabla 33: “Experimentos de Priorización de Requerimientos de Software basados en AHP”	184

Prólogo

“Un Enfoque de Priorización de Requerimientos, a partir de la Segmentación de las Preferencias de los Stakeholders”

Carlos Martín Azzolini

La presente Investigación, se propone desarrollar un modelo de priorización de requerimientos, a través de la segmentación de las preferencias de los *stakeholders*. Para tal fin, se ha diseñado un Esquema General de Prioridad de Requerimientos de Software, que permite agregar las preferencias grupales de los *stakeholders*, a través de un mecanismo de ponderación de naturaleza socio-cognitiva. Asimismo y en base a dicho Esquema General, se determina el grado de alineamiento que un determinado grupo obtiene, tras comparar su Esquema Particular con el General. En el caso en que dicha comparación haya sido “Baja o Moderada” se propondrá una instancia de negociación focal, basada en la fusión de una serie de enfoques preexistentes en la materia. De esta manera, se espera que cada uno de los grupos de *stakeholders* haya podido formular un esquema particular de prioridad, cuyo grado de alineamiento con el Esquema General, sea “Aceptable”. **Como consecuencia de lo descrito, el modelo se postula como un enfoque experimental para abordar el problema que supone priorizar y negociar, los requerimientos de un determinado Proyecto de Software.**

Por su parte, dado que la formación de grado del autor no proviene del área Informática, sino de la Economía, el proceso de investigación que da origen a esta tesis tiene un enfoque transdisciplinar. En consecuencia, el autor ha debido sumar a su formación en las Teorías de: *Utilidad, Decisión y Negociación* (características del área: Microeconomía Aplicada) otros enfoques, tales como metodologías de Inteligencia Computacional y de la Psicología Cognitiva (especialmente Memoria Semántica), como requisito previo para abordar el Problema de la Priorización de Requerimientos según la concepción propuesta. Para ello, se ha establecido como miembro adscrito en Proyectos de Investigación del Área *Inteligencia Computacional a cargo de la Dra. Lucía Isabel Passoni en el Laboratorio de Bioingeniería de la Facultad de Ingeniería, Universidad Nacional de Mar del Plata*, y en forma concurrente, como miembro del *Centro de Investigación en Procesos Básicos, Metodología y Educación (Facultad de Psicología de la Universidad Nacional de Mar del Plata) a cargo del Dr. Jorge Vivas.*

En base a la experiencia citada, el autor ha podido desarrollar parte de la Metodología utilizada en la presente Investigación. Al efecto, puede consultarse: Pazgón;

Calcopietro; Azzolini ; Vivas (2010); Vivas, Azzolini, Vivas (2010); Vivas; Comesaña; Pazgon; Azzolini (2010).

Organización de la Tesis

A los efectos del desarrollo de la presente tesis, se ofrece a continuación una estructura organizativa de la misma. Asimismo y dada la naturaleza del trabajo, se requiere una organización basada en capítulos, en los cuales, se procura establecer un andamiaje de conocimientos articulados, cohesivos y consistentes. De esta manera, se espera servir tanto al propósito general, como a los diferentes objetivos particulares del presente plan de trabajo.

Capítulo 1: “El contexto de la Priorización de Requerimientos de Software y su justificación como temática de estudio”: El presente Capítulo, tiene como objetivo básico contextualizar la Priorización de Requerimientos, dentro del campo de conocimiento de la Ingeniería de Software (SWEBOK: *Software Engineering Body of Knowledge*). Luego, se pretende justificar su estudio a partir de su importancia e interacción con otros problemas de la Ingeniería de Requerimientos.

Capítulo 2: “Estado de Arte: Priorización de Requerimientos de Software”: En el presente Capítulo, se describirán aquellas líneas teóricas que han abordado el problema de la priorización de requerimientos de software a lo largo del tiempo. Luego, se pretende sistematizar las investigaciones citadas mediante su agrupación en Enfoques Teóricos.

Capítulo 3: “Marco Teórico y Metodológico de la Investigación”: En el presente Capítulo, se profundizarán aquellos enfoques de la Priorización de Requerimientos de Software, que mejor se adapten al problema y a los objetivos de investigación planteados en esta tesis. Asimismo, y desde una perspectiva estrictamente metodológica, se introducirán algunos aspectos de la Psicología Cognitiva, Redes Neuronales y la Teoría de Conjuntos Difusos.

Capítulo 4: “Modelo de Priorización de Requerimientos de Usuario a partir de la Segmentación de Preferencias de los Stakeholders (MPRUSPS)”: En el presente Capítulo, se formalizará cada una de las etapas que integran el Modelo de Priorización de Requerimientos de Usuarios, a partir de la segmentación de las preferencias de los *stakeholders* (MPRUSPS). Para ello, se propone integrar los aspectos teóricos y metodológicos oportunamente desarrollados (Capítulo 3).

Capítulo 5: “Experimentación: Ejemplo de aplicación del Modelo de Priorización de Requerimientos de Usuarios a partir de la Segmentación de Preferencias de los Stakeholders (MPRUSPS)”: En el presente Capítulo, se desarrollará un ejemplo típico de

aplicación del MPRUSPS (Capítulo 4). Para ello, se simulará un caso de experimentación en donde el MPRUSPS pueda ser aplicado. Asimismo se prevé una instancia de contraste mediante la aplicación del enfoque de Karlsson (1996) en el caso propuesto.

Capítulo 6: “Caso Experimental: Subsistema de Flujo de Trabajo del CIMEPB”: En este Capítulo, se establecerá la Especificación de los Requerimientos Funcionales del Subsistema de Flujo de Trabajo del Centro de Procesos Básicos en Metodología y Educación, órgano dependiente de la Facultad de Psicología de la Universidad Nacional de Mar del Plata. Asimismo, se mencionaran los aspectos generales inherentes a las Técnicas de Elicitación y Validación utilizadas para la elaboración de la presente Especificación, como así también, se dejará constancia de aquellos *stakeholders* que han participado en el presente proceso.

Capítulo 7: “Resultados de la aplicación del MPRUSPS en el Caso Experimental: Subsistema de Flujo de Trabajo del CIMEPB”: En el presente Capítulo, se analizarán los resultados obtenidos tras la aplicación del MPRUSPS en el caso de experimentación real: “Subsistema de Flujo de Trabajo del CIMEPB”.

Capítulo 8: “Discusión general respecto de la implementación del MPRUSPS”: En el presente Capítulo, y en base a los resultados obtenidos en los casos propuestos (Capítulos: 5 y 6), se debatirán los aspectos metodológicos del MPRUSPS. Luego, se evaluará la factibilidad de adoptarlo como método para priorizar y negociar requerimientos.

Capítulo 9: “Conclusión, Recomendaciones y Futuros Trabajos”: En el presente Capítulo, se detallarán los hallazgos más significativos en torno a la implementación del MPRUSPS. Asimismo, se recomendarán las futuras investigaciones que serán necesarias desarrollar, a partir de la experiencia capitalizada en esta investigación.

Bibliografía Utilizada: Como su nombre lo indica, esta sección final contendrá todas las citas bibliográficas que referencian los trabajos de investigación que han sido utilizados para la elaboración del presente trabajo.

Introducción General

La Priorización de Requerimientos de Software, implica la constitución de un proceso de decisión de naturaleza compleja. Este proceso es producto de la interacción de las *denominadas partes interesadas (stakeholders)*¹ que elaboran un juicio crítico y consensuado, con el fin de seleccionar, en primera instancia, el conjunto de requerimiento que deberían ser implementados a lo largo del ciclo de vida del software.

Por su parte, y tal cual lo manifiestan en Hadad (et. al. 2009), asignar prioridades en los requerimientos, conlleva necesariamente discernir cuestiones inherente a la dependencia de los mismos, la multiplicidad de intereses entre los *stakeholders*, las limitaciones de recursos, las necesidades de negocio, los costos de implementación, entre otros factores. En consecuencia, **La Priorización de Requerimientos**, ha sido reconocida como una de las actividades más importantes dentro del proceso de desarrollo del Software [Gomes y Pettersson (2007), Lethola y Kauppinen (2006), Ngo-The y Ruhe (2005), Moisiadis, (2002)]. En efecto, Ma (2009) considera que dicha actividad contribuye favorablemente en los procesos de decisión que son inherentes a la planificación, gestión y control del Desarrollo de Software.

En función de lo descrito anteriormente, se podrá indicar que la priorización de Requerimientos se relaciona con variables tales como: *Tiempo, Esfuerzo y Riesgo*. En consecuencia, la aplicación de un Enfoque de Priorización Requerimientos, traería aparejado cambios significativos en torno a las variables mencionadas, afectando así, el curso de un Proyecto de Desarrollo. Luego, en la mayoría de los Proyectos, no resulta factible implementar en forma conjunta la totalidad de los requerimientos disponibles dada las restricciones impuestas por las variables mencionadas. Con lo cual, en Lehtola (et. al., 2004) consideran que resultaría esencial decidir cuál subconjunto de los requerimientos especificados, deberían ser implementados en una próxima versión del Producto. Asimismo, en Wieggers (1999) se establece que la prioridad de los requerimientos, se constituye como una información crítica que facilita tal proceso.

En consecuencia, esta investigación indaga las corrientes teóricas que versan sobre la problemática de la priorización de requerimientos de software. Algunos enfoques parten de la idea central de formular una interacción continua y enriquecida por parte de los *stakeholders*, a los efectos de consensuar un esquema general de implementación de

¹ En adelante se utilizará el término en inglés *stakeholders* para referirse a las partes interesadas consideradas como el conjunto de personas que serán afectadas en forma directa o indirecta por el proyecto de software [Sommerville y Sawyer (1997)].

requerimientos de software [Boehm (et. al., 1995); Boehm e In (1996); Damian (et. al., 2008); In y Roy (2001); In (et. al. 2001); Ruhe (et. al, 2002); por citar algunos]. Otros en tanto, pretenden profundizar la faz cognitiva del proceso de priorización de requerimientos, toda vez que dicho proceso, se basa en decisiones y preferencias humanas [Martín (et. al, 2003); Martínez Carod y Cebrich (2007); Nuseibeh y Easterbrook (2000) entre otros]. Por su parte, existiría una tercera línea de investigación que promovería el uso de métodos analíticos a los efectos de la determinación de un esquema de priorización de requerimientos [Aho (et. al., 1983); Beck (2000); Brackett (1990); Leffingwell y Widring (2000); Saaty (1980; 1986; 1990; 2001) por ejemplo].

A partir del relevamiento y análisis de los antecedentes, se propone como **objetivo general diseñar un modelo de priorización de requerimientos de software, a partir de la segmentación de preferencias de los stakeholders**. Para ello, necesariamente se fusionarán una serie de enfoques preexistentes a los efectos de satisfacer el problema que implicaría la articulación e implementación de un **modelo** con tales características. Con lo cual, en primera instancia, será necesario disponer de un método de relevamiento de preferencias. En segundo lugar, se deberá especificar un criterio que permite segmentar las preferencias relevadas. Luego, y dado que las preferencias son relevadas a partir del juicio provisto por los diferentes *stakeholders*, se debería fundamentar cognitivamente el porqué de las mismas. Finalmente y ante el consecuente problema de disponer de un conjunto de preferencias segmentadas, se debería implementar algún mecanismo de negociación entre los *stakeholders*, a los efectos de constituir y consensuar un Esquema General de Prioridad implementable de Requerimientos de Usuarios (EGPIRU).

El desarrollo de esta Tesis, se enmarca en un Proyecto de Investigación que se desarrolla en el Centro de Proceso Básicos, Metodología y Educación de la Facultad de Psicología de la Universidad Nacional de Mar del Plata, y en el Laboratorio de Inteligencia Artificial de la Facultad de Ingeniería (UNMdP). En este proyecto se propone, entre otras cuestiones, el desarrollo de un Producto de Software (Web), en el que resultaría esencial articular un esquema de priorización de requerimientos de usuarios, que permita conciliar los recursos disponibles afectados al ciclo de vida del software a implementar. De esta manera, y por medio del presente caso de estudio, se espera aportar un nuevo enfoque de priorización que contribuya con la Ingeniería en Requerimientos, y en general, con la Ingeniería de Software.

Objetivos Particulares

- Contextualizar el Problema de la Priorización de los Requerimientos de Software dentro del Cuerpo de Conocimiento de la Ingeniería de Software (SWEBOK: *Software Engineering Body of Knowledge*).
- Describir las líneas teóricas más importantes y actuales que versan sobre la problemática bajo estudio.
- Incluir aquel cuerpo teórico y metodológico que permita fundamentar conceptual y funcionalmente, un Modelo de Priorización de Requerimientos de Usuario a partir de la Segmentación de Preferencias de los *Stakeholders*.
- Recopilar un compendio de técnicas de Inteligencia Computacional a los efectos de contribuir metodológicamente a la constitución de la metodología consignada.

Capítulo 1

“El Contexto de la Priorización de Requerimientos de Software y su justificación como temática de estudio”

1.1 Introducción

Como se ha mencionado en la Introducción, la Priorización de Requerimientos de Software, implica la constitución de un proceso de decisión de naturaleza compleja. Este proceso es producto de la interacción de los *stakeholders* quienes elaboran un juicio crítico y consensado, con el fin de seleccionar, en primera instancia, el conjunto de requerimiento que deberían ser implementados a lo largo del ciclo de vida del software. Una vez logrado el consenso sobre el conjunto de requerimientos a implementar, se asignará prioridades a los mismos, lo que implicará la ponderación de un conjunto de factores, tales como: tiempos, costos, preferencias, entre otros.

De esta manera, el presente Capitulo tendrá como objetivo básico contextualizar el problema de la Priorización de los Requerimientos de Software. Luego, se procederá a fundamentar el porqué de su estudio, y finalmente, se indagará su relación con el problema de la “Próxima Versión”.

1.2 Contexto Disciplinario de la Priorización de Requerimientos de Software

Antes de justificar el problema que implica la priorización de requerimientos de software, se definirá en primera instancia, el concepto de requerimiento. Según el Glosario Estándar de la Terminología de la Ingeniería de Software de la IEEE (1990) un requerimiento es:

1. Una condición o capacidad que un usuario necesita para resolver un problema o alcanzar un objetivo.
2. Una condición o capacidad que debe poseer un sistema o los componentes de un sistema para satisfacer un contrato, estándar, especificación u otro documento formalmente impuesto.
3. Una representación documentada de una condición o capacidad como las descritas en (1) y (2).

Similarmente a lo planteado, la Guía SWEBOK (2004) define a un requerimiento de software como una propiedad que debería ser exhibida por el Software desarrollado o adaptado, a los efectos de resolver un problema en particular. Asimismo, su formulación surge a partir de una compleja interacción que involucra las necesidades de los *stakeholders*, su posicionamiento dentro de la Organización, y además, el entorno ambiental en donde el software se aplicará.

Siguiendo con la descripción de los requerimientos, además de las propiedades de comportamiento que expresan, adquieren otros atributos: *Prioridad, Esfuerzo y Riesgo*. De esta manera, se debe formular el grado de prioridad de los requerimientos; las restricciones impuestas por los recursos disponibles (esfuerzo) y las características particulares del dominio de aplicación (riesgo) [Moisiadis, (2002), SWEBOK (2004)].

Varios autores, como Berander (2004), Bergman (et. al., 2003), Davis (2003), Doerr (et. al., 2007), Nuseibeh y Easterbrook (2000) y Moisiadis (2002) establecen que uno de los desafíos más emblemáticos que debería superar la Ingeniería de Software, consiste en desarrollar sistemas que reflejen y satisfagan las necesidades de los *stakeholders*. Tal circunstancia, es fundamentada por Moisiadis (2002), cuando indica que una de las causas más frecuentes de retrasos y excesos en los costos presupuestados en los Proyectos, devendría porque los requerimientos más importantes no son elicitados, priorizados ni analizados en las etapas tempranas del proceso de desarrollo. En

consecuencia, las circunstancias descritas vulneran la calidad del Software [Bergman (et. al., 2003)], y por tanto, impactan en forma desfavorable en la satisfacción de los *stakeholders* [Berander, (2004)].

La Priorización de Requerimientos, ha sido reconocida como una de las actividades más importantes dentro del proceso de desarrollo de software [Gomes y Pettersson (2007), Lethola y Kauppinen (2006), Ngo-The y Ruhe (2005), Moisiadis, (2002)]. En efecto, Ma (2009) considera que dicha actividad contribuye favorablemente en los procesos de decisión que son inherentes a la planificación, gestión y control del Desarrollo de Sistemas.

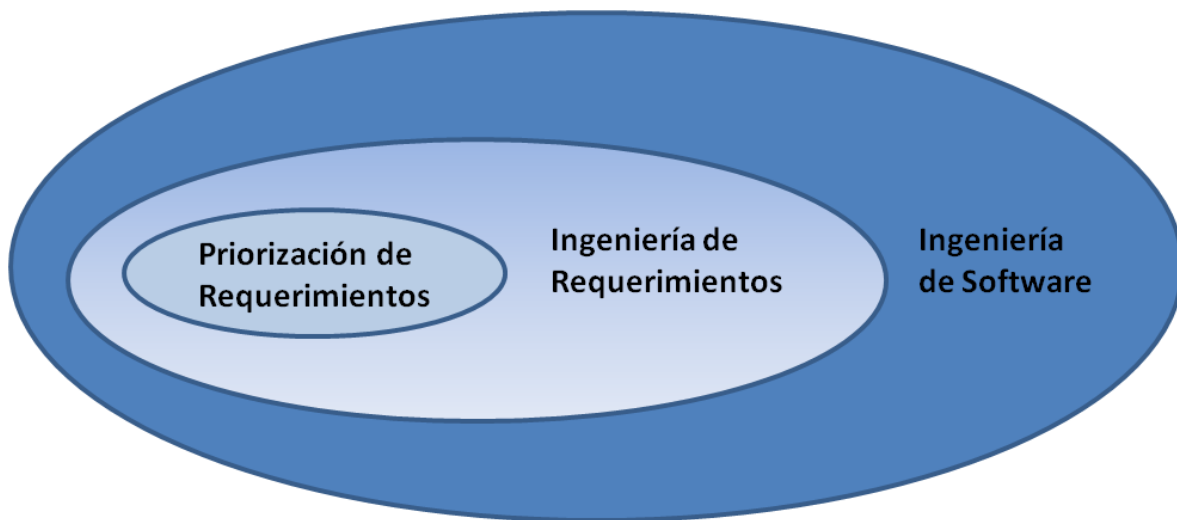


Figura 1.1: Situación contextual de la Priorización de Requerimientos: Fuente: Ma (2009). Traducción.

Como se aprecia en la Figura 1.1, el mencionado autor, propone una relación sistemática entre la Priorización de Requerimientos, la Ingeniería de Requerimientos y finalmente, la Ingeniería de Software. Es decir, explicita una relación de agregación en la cual, se establece una serie de dependencias intrínsecas entre las temáticas establecidas.

Por su parte, el cuerpo de conocimientos de la Ingeniería de Software [SWEBOK, (2004)] establece un marco formal y normativo de las Categorías de Conocimiento (KC) de la Ingeniería de Software (SE). Asimismo, y a los efectos de caratular el problema de Investigación, se propondrá una breve revisión del Área de Conocimiento: "Requerimientos de Software", recorriendo en consecuencia, sus diferentes tópicos de interés.

Área de Conocimiento: Requerimientos de Software (“Software Requirements”)

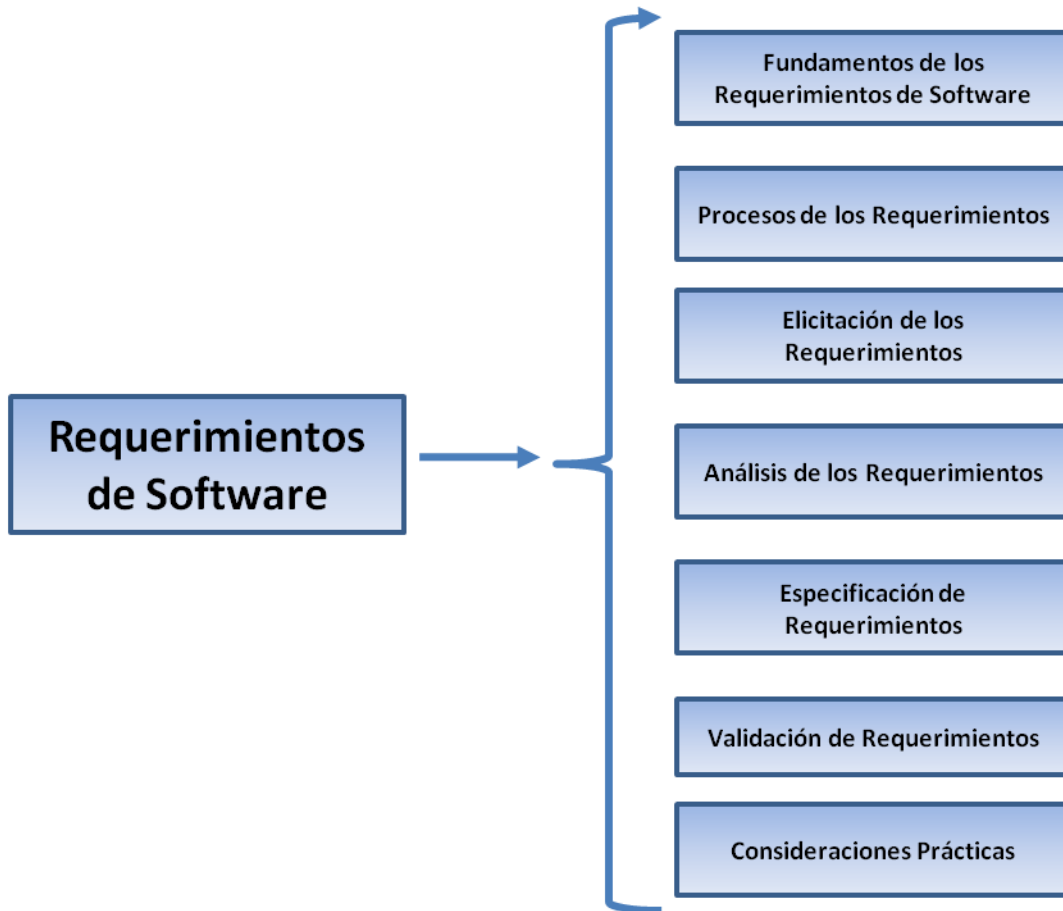


Figura 1.2: “Área de Conocimiento: “Requerimientos de Software”. Fuente: SWEBOK (2004).Traducción.

Como se indica en la Figura 1.2, la Guía SWEBOK (2004) especifica que al Área de Conocimiento: *Requerimientos de Software* (“*Software Requirements*”) le concierne todos aquellos tópicos inherentes a la Elicitación, Análisis, Especificación y Validación de los requerimientos de Software. Asimismo, dada la compleja naturaleza de la temática de

estudio, resulta necesario transcribir algunas enunciaciones de la Guía, a los efectos de comprender bajo qué perspectivas se podría abordar la “Priorización de Requerimientos de Software”. A continuación, se ofrece una breve descripción de los tópicos más importantes.

Procesos de los Requerimientos (“Requirements Process”) : es una sección que introduce los procesos de requerimientos de software. Asimismo, se compone de 5 sub-áreas y muestra de qué manera se relacionan las mismas con el resto de los procesos inherentes a la Ingeniería de Software [Davis, (1993)]. En efecto una de las sub-áreas, se denomina *Actores del Proceso (“Process Actors”)* la cual introduce los roles detentados por aquellas personas que participan en los procesos de requerimientos. Por su parte, el presente tópico pone de manifiesto la complejidad del proceso de requerimientos, dado el enfoque interdisciplinario que presupone la mediación efectiva entre los especialistas del dominio y los Ingenieros de Software. Asimismo, se reconoce la imposibilidad de satisfacer la totalidad de los requerimientos planteados, por lo cual, se exhorta al Ingeniero de Software a negociar con los *stakeholders* identificados.

Elicitación de los Requerimientos (“Requirements Elicitation”) es una sección que versa respecto al origen y adquisición de los requerimientos de software. Asimismo, dicha Área se divide en 2 sub-áreas denominadas: *Fuentes de los Requerimientos (“Requirements Source”)* y *Técnicas de Elicitación (“Elicitation Techniques”)*. En efecto, en la primera sub-área se establece el problema de la identificación de los *stakeholders* y sus puntos de vistas. Como se sabe, un *stakeholder* hace referencia a un actor que se ve influido por el proceso del Software. Luego, dependiendo del rol que adquiere el *stakeholder* en dicho proceso, será factible comprender el porqué de su punto de vista. A partir de allí, se constituyen los elementos críticos que el Ingeniero de Software debería considerar en el proceso de elicitación. Por tal motivo, el entendimiento de la cultura organizacional junto a la estructura política, podrían considerarse como aquellos factores claves que permitirán al especialista en requerimientos, comprender el dominio con mayor especificidad.

Análisis de Requerimientos (“Requirements Analysis”): es una sección que enuncia las problemáticas inherentes a la detección y resolución de conflictos entre requerimientos, los límites del software, su interacción con el entorno, así como también, de la forma en que los requerimientos de software surgen a partir de la interpretación de los requerimientos del sistema. Por su parte, el área de conocimiento se subdivide en 4 tópicos, denominados: *Clasificación de los Requerimientos (“Requirements Classification”)*, *Modelo Conceptual (“Conceptual Modeling”)*, *Diseño de Arquitectura y Designación de Requerimientos (“Architectural Design and Requirements Allocation”)* y *Negociación de*

Requerimientos (*Requirements Negotiation*). En la primera de ellas, surge una clasificación de requerimientos en función de su grado de prioridad. Indicando en consecuencia, que cuando mayor sea la prioridad de un requerimiento, más importante se vuelve para el cumplimiento de los objetivos globales especificados en el software. Por su parte, en la sub-área *Negociación de Requerimientos* (*Requirement Negotiation*), se especifican algunas particularidades inherentes a la problemática “conflicto” que pueden tenerse a lugar, cuando se analiza un conjunto de requerimientos. En efecto, tales pueden acontecer como consecuencia de una perspectiva diferente que los *stakeholders* pueden experimentar, en torno al conjunto de las características deseables que deben adquirir los requerimientos. Asimismo, y una vez más, se sugiere que el Ingeniero de Software promueva y facilite el consenso entre los *stakeholders* a fin de resolver tales circunstancias.

Se deduce del análisis del cuerpo de conocimiento de la Ingeniería de Software, que el proceso de Priorización de Requerimientos, implica un análisis temático de un subconjunto de las Unidades de Conocimiento del Área “Requerimientos de Software”. En consecuencia, surge la necesidad de identificar los Actores del Proceso de Requerimientos (“Requirements Process”), sus puntos de vista a partir de un análisis de la cultura y política organizacional (“Requirements Source” en “Requirements Elicitation”) para luego clasificar al conjunto de requerimientos en función de su grado de prioridad (“Requirements Classification” en “Requirements Analysis”) teniendo en consideración, la posibilidad de implementar un esquema de negociación que pueda resolver aquellos conflictos que puedan tenerse a lugar durante la ejecución del presente proceso (“Requirements Negotiation” en “Requirements Analysis”).

1.3 La Priorización de Requerimientos y su relación con el Problema de la Próxima Versión (NRP)

Como se ha mencionado, se considera a la priorización de requerimientos como una actividad crítica e importante dentro del proceso de desarrollo del software. Una de las cuestiones que subyace en este tópico en particular, hace referencia al momento en el cual debería ser factible su aplicación.

En efecto, Hatton (2008) sugiere que los desarrolladores deberían ser los encargados de escoger el momento oportuno para comenzar el proceso de priorización.

Sin embargo, el mismo autor, establece que si el proceso de priorización se ejecuta en etapas tempranas del desarrollo, se correría el riesgo de sesgar la estimación del tiempo del Proyecto y los recursos afectados al mismo, puesto que, ésta etapa se caracteriza por una gran volatilidad de los requerimientos. Si por el contrario, el proceso de priorización acontece en etapas tardías, se corre el riesgo de afectar una cantidad innecesaria de recursos del Proyecto, dado que existiría un cúmulo de requerimientos de bajo valor que han sido identificados en etapas más tempranas, con lo cual, aquellos de alto valor que deberían haber sido priorizados previamente, no han sido observados producto de la innecesaria acumulación secundaria o no prioritaria de información referente a los requerimientos.

Finalmente, y en base al análisis de una muestra representativa de Proyectos, el autor recomienda que los desarrolladores escojan el momento de aplicación de un Enfoque de Priorización, teniendo en consideración, una serie de factores tales como: el tiempo disponible, la metodología usada en el Proyecto, el grado de conocimiento que se tenga del conjunto de requerimientos y el grado de prioridad requerida.

En función de lo descrito anteriormente, se podrá indicar que la Priorización de Requerimientos se relaciona con variables tales como: Tiempo, Esfuerzo y Riesgo. En consecuencia, la aplicación de un Enfoque de Priorización Requerimientos, traería aparejado cambios significativos en torno a las variables mencionadas, afectando así, el curso de un Proyecto de Desarrollo.

Por su parte, y tal cual lo manifiesta Wiegers (1999) cuando las expectativas de los *stakeholders* son altas, los tiempos cortos y los recursos limitados, el Producto de Software debería enviarse con la funcionalidad esencial tan pronto como sea posible. En consecuencia, el alcance de cada versión debería ser limitada [Siddiqi (et. al. 1996)].

Como sucede comúnmente en la mayoría de los Proyecto de Software, no resulta factible implementar en forma conjunta la totalidad de los requerimientos disponibles dada las restricciones impuestas por las variables: Tiempo, Esfuerzo (recursos disponibles) y Riesgo [Firesmith, (2004)]. Con lo cual, en Lehtola (et. al., 2004) consideran que resulta esencial decidir cuál subconjunto de ellos, deberían ser implementados en una próxima versión del Producto. Considerando a Wiegers (1999), él establece que la prioridad de los requerimientos se constituye como una información crítica, puesto que, no sólo serviría a los efectos de ignorar parcialmente un subconjunto de requerimientos no prioritarios, sino que, también, dicha información ayudaría al Gerente del Proyecto a discernir sobre los conflictos acontecidos, planear escenarios de próximas versiones y calcular compensaciones de recursos.

Por su parte, “*El problema de la próxima versión*” (NRP: *Next Release Problem*) es uno de los tópicos más estudiados dentro de la Ingeniería de Software [Bagnall (et. al., 2001), Harman (2007), Harman (et. al., 2001)]. Según Durillo y Zhang (2009), se define como un proceso de decisión, en el cual, una compañía de software se propone determinar aquel conjunto de requerimientos que deberían implementarse en forma incremental, teniendo en consideración, aquella versión del producto que maximice la satisfacción de los clientes, y que a la vez, minimice los costos de tal implementación. Luego, y tal cual consideran los autores, maximizar la satisfacción de los clientes y minimizar el uso de los recursos disponibles, se constituye como un objetivo contrapuesto en sí mismo. Por tal motivo, coinciden en abordar el problema mediante una técnica de optimización multi-objetivo.

En otra perspectiva, autores como Jewer y McKay (2008), Aurum y Wohlin (2003) establecen que NRP es un proceso complejo de decisión, y como tal, muchas veces se resuelve gracias a la experiencia individual y el conocimiento tácito de los *stakeholders* del proceso. En Karlsson (et. al., 2004), justifican dicho fenómeno dada la inestabilidad para estimar tanto los esfuerzos de implementación, como la predicción del valor asignado por el cliente del producto. En consecuencia, las compañías de software establecen métodos *ad-hoc* para establecer su proceso de decisión en torno al problema que supone versionar el producto.

Asimismo, en Karlsson (et. al., 2004; et. al., 2007) consideran que el proceso de Priorización de Requerimientos sienta la base para solucionar el problema de la próxima versión. En efecto, manifiestan: “**...el éxito en el proceso de selección de aquel conjunto de requerimientos que serán implementados en una versión particular del producto (Next Release), depende fuertemente del éxito obtenido en el proceso de Priorización de Requerimientos**”. Por su parte, sí se considera que el objetivo último de cualquier Organización de Software, consiste en crear sistemas que satisfagan las expectativas de los clientes, entonces, quien es el responsable de tomar la decisión respecto de la planeación de las versiones del producto, debería conocer la prioridad relativa de los requerimientos de software peticionados [Wieggers (1999)].

En el transcurso de un Proyecto, el Gerente de Proyecto (“*Project Manager*”) debe tomar decisiones inherentes al plan de versiones del producto. Para ello, debe: estimar la disponibilidad de recursos, tiempos, hitos; resolver conflictos ante perspectivas diferenciales de los *stakeholders*; analizar el mercado; determinar la estrategia de producto; estimar costos; entre otras tareas. Luego, cada uno de estos factores debería estar presente durante la planeación de cada una de las versiones futuras del producto [Karlsson (et. al., 2007)].

1.4 ¿Qué beneficios traerá aparejado la implementación de un Proceso de Priorización de Requerimientos?

Como consecuencia de lo descripto, la Priorización de Requerimientos se constituye como una de las actividades claves dentro de los procesos que forman parte de la Ingeniería de Requerimientos. Asimismo, su correcta implementación, trae aparejado una serie de beneficios concretos que se materializan cuando resulte indispensable estimar el uso de los recursos disponibles, el tiempo comprometido y los costos incurridos en un Proyecto de Software. Luego, y con relación al “Problema de la Próxima Versión”, permite sentar la base para una mejor planeación del producto.

Siguiendo el razonamiento de Wiegers (1999), Berander (et. al, 2004) Perini (et. al., 2009), cuando consideran la prioridad relativa de un requerimiento como un elemento crítico dentro del proceso de estimación de las versiones del producto, se podrá inferir que la internalización de las preferencias de los clientes dentro de un enfoque de priorización, contribuye sin lugar a dudas, a incrementar el valor de negocio y la satisfacción de los *stakeholders* respecto de la versión o del producto final liberado.

Bajo esta perspectiva, se podrá asumir que la sumatoria de los requerimientos implementados en cada una de las versiones de un determinado Producto de Software, deberá ser igual al total de los Requerimientos contenidos en la Especificación que le dio origen. Debe destacarse, que los Requerimientos Priorizados son necesarios para llevar adelante el proceso de planificación de cada una de las versiones. De esta manera, se facilitará la estimación de recursos que deberían ser aplicados a lo largo del ciclo de vida del Producto. En consecuencia, la disponibilidad de los Requerimientos Priorizados posibilita que cada versión del Producto se encuadre dentro de los presupuestos establecidos, que cuente con el beneplácito de los *stakeholders*, y en función de las circunstancias alegadas, incremente su valor y calidad.

1.5 Consideraciones Finales del Capitulo

Como se pudo apreciar, la Priorización de Requerimientos de Software involucra explícita e implícitamente varias secciones del Área de Conocimiento “*Software Requirements*” (SWEBOOK, 2004). La complejidad de este proceso, viene dada por la necesidad de establecer un criterio de prioridad que sea comprendido por la totalidad de los *stakeholders*; recabar las preferencias por ellos asignadas; y resolver los eventuales conflictos que pueden tenerse a lugar, a través de la implementación de un proceso de negociación.

Por su parte, se ha podido delimitar el **“Problema de la Priorización de Requerimientos”** de otro problema íntimamente relacionado con él, al cual se lo denomina: **“Problema de la Próxima Versión (NRP)”**. En efecto, las cuestiones inherentes a la planificación del producto, factibilidad de implementación, entre otras, serán inherentes al problema NRP y no al problema de priorización de requerimientos.

Como consecuencia de lo anterior, se podrá concluir que la Priorización de Requerimientos de Software, hace referencia a una actividad crítica dentro de la Ingeniería de Requerimientos, a partir de la cual, se establece un proceso de decisión preferentemente consensuado, tendiente a establecer el orden de implementación de una serie de requerimientos de software, tomando en consideración, algún conjunto definido de variables objetivo (criterios), como ser: satisfacción de los *stakeholders*, costos, necesidades del cronograma, entre otras.

Capítulo 2

“Estado de Arte: “Priorización de Requerimientos de Software”

2.1 Introducción

Como se ha indicado, La Priorización de Requerimientos de Software, se constituye como una actividad crítica dentro de la Ingeniería de Requerimientos (Hatton, 2008; Ma, 2009). Asimismo, y entendido el fenómeno como un problema de decisión, resulta vital su determinación, puesto que determinará el ciclo de vida del producto de software.

Por su parte, y tal cual lo manifiestan en Hadad (et. al. 2009), asignar prioridades en los requerimientos, conlleva necesariamente discernir cuestiones inherente a la dependencia de los mismos, la multiplicidad de intereses entre los *stakeholders*, las limitaciones de recursos, las necesidades de negocio, los costos de implementación, entre otros factores. Luego, tal priorización resulta crítica, puesto que todo proyecto de software se encuentra limitado por un conjunto de restricciones comunes, como el tiempo, costo y afectación de recursos [Karlsoon, 1996; et. al. 1998; et. al. 2006; y Davis, 2003].

El presente capítulo, se articula en el desarrollo de las principales líneas teóricas, inherentes al Proceso de Priorización de Requerimientos. Dada la naturaleza del objeto de estudio, se describirán los principales enfoques que han sido propuestos a los efectos de abordarlo. En consecuencia, el Capítulo versará sobre los Aspectos de la Priorización (2.2), el Enfoque Cognitivo (2.3) de Negociación (2.4) y Métodos (2.5).

2.2 Aspectos de la Priorización de Requerimientos

Como se ha mencionado, la Priorización de Requerimientos de Software, implica la constitución de un proceso de decisión de naturaleza compleja. Asimismo, existen diferentes concepciones para el fenómeno en cuestión. Por ejemplo, Sommerville (1996) lo considera como una actividad, mediante la cual, los requerimientos de software más importante son descubiertos. Por su parte, Firesmith (2004) lo define como un proceso que determina el orden de implementación de los requerimientos de un sistema ó como aquel proceso que determina el orden de importancia de los requerimientos en función de la percepción de los *stakeholders*.

Un aspecto, puede considerarse como un atributo o propiedad inherente a un requerimiento en particular, en el cual, será susceptible basarse a los efectos de establecer un orden de importancia o prioridad, dentro de un proceso de priorización de requerimientos [Berander (2004) ; Ma (2009)]. Asimismo, Henry (et. al., 1993) y Hatton (2007) citado en Ma (2009) consideran los términos: “factores” y “criterios” como análogos al descripto.

En efecto, tales aspectos, criterios ó factores, han sido enumerados en la literatura con el objeto de abordar el problema de priorización, toda vez que su proceso de contemplación y constitución, subyace implícitamente de la interacción de los *stakeholders* del dominio. Luego, cuando los participantes emprenden un proceso de Priorización, se focalizan normalmente en los siguientes aspectos [Berander (2004, Ma (2009) Motupally (2008)]:

- Importancia: Se refiere a un criterio de priorización, por intermedio del cual, los *stakeholders* del proceso manifiestan para cada requerimiento en particular, su ordenación relativa respecto al presente aspecto. Sin embargo, el criterio “importancia” utilizado por un *stakeholder* en particular, puede identificarse por una serie de conceptos que no necesariamente resultan análogos a otro participante. En consecuencia, y tal cual lo manifiestan Berander (2004), Lehtola (et. al. 2004) y Ma (2009), el criterio debería ser previamente especificado para que los *stakeholders* puedan priorizar en forma homogénea, cada uno de los requerimientos disponibles.
- Penalidad: Se refiere a un criterio de priorización, por intermedio del cual, se establece la implicancia de no implementar un determinado requerimiento [Wieggers (1999)].

- Costo: Se refiere a un criterio de priorización, por intermedio del cual, se contabilizan las erogaciones monetarias incurridas a los efectos de implementar una serie de requerimientos candidatos [Karlsson y Ryan (1997)]. Asimismo, constituye un aspecto que refleja la complejidad y calidad de los requerimientos implementables [Motupally, (2008); Berander (2004)].
- Tiempo: Se refiere a un criterio de priorización, por intermedio del cual, los requerimientos implementables se valoran de acuerdo al tiempo necesario que se ha de incurrir, a los efectos de implementarlos en forma efectiva [Aurum y Wholin (2003)]. Asimismo, y tal cual manifiesta Motupally (2008), el tiempo incurrido depende de las características del proceso de desarrollo del software y de sus actividades concurrentes e independientes.
- Riesgo / Volatilidad: Se refieren una serie de criterios de priorización, por intermedio de los cuales, los requerimientos son valorables de acuerdo a su grado inherente de riesgo y probabilidad de cambio. Tales factores, inciden en el ciclo de desarrollo del producto, y sus manifestaciones más salientes implican retrasos en el cronograma, costos, presupuestos de recursos, entre otros. La información disponible de los requerimientos implementables, se constituye como el factor clave que podría mitigar los efectos del riesgo y la volatilidad [Motupally, (2008), Berander (2004), Ruhe (2003), Wiegers (1999)].
- Estrategia de Negocios: Se refiere a un criterio de priorización, por intermedio del cual, se valora la implementación del conjunto de requerimientos, en función de la probabilidad de inserción del producto, de la existencia de competidores en el mercado, entre otros factores. En sí mismo, se establece como un conjunto de sub-criterios que son ponderables por la entidad encargada de desarrollar el producto [Berander (2004), Firesmith (2004), Lethola (et. al. 2004), Ma (2009), Motupally, (2008)].
- Utilidad: Se refiere a un criterio de priorización, por intermedio del cual, se pondera el grado de satisfacción que la implementación de un determinado requerimiento, le reporta a un conjunto *stakeholders*. Considerando las preferencias, se intenta ordenar el conjunto de requerimientos disponibles de acuerdo a las expectativas reveladas por los participantes [Giesen y Völker (2002)].

Como consecuencia de lo descrito, a lo largo de estos últimos años, se han desarrollado una serie de metodologías de priorización de requerimientos, que ponderan los criterios mencionados, ya sea de manera explícita o implícita. A partir de allí, y según Lehtola y Kauppinen (2004) y Motupally (2008), los enfoques de priorización de requerimientos, pueden clasificarse en dos grandes categorías: “*Enfoques de Negociación y Métodos*”. El primero de ellos, se basa en la idea de un acuerdo continuo y enriquecido por parte de los diferentes *stakeholders*, los cuales, puntúan requerimientos, lo ordenan y luego comienzan tal proceso cediendo, abandonando y renegociando posiciones. Cercana a esta línea de pensamiento, se encuentran aquellos trabajos que ponen como centro de interés, los *aspectos socio-cognitivos* de los *stakeholders*. De esta manera, se pretende hallar aquel conjunto de requerimientos prioritarios a partir de los procesos de negociación, que surgen como consecuencia de las percepciones diferenciales de los sujetos en torno a la relevancia e interpretación de los requerimientos [Martín (et. al, 2003)].

Por su parte, los *Métodos* parten de la concepción que la prioridad de un requerimiento, surge a partir de la combinación de una serie de valores, los cuales, ponderan diferentes factores que integran la dimensión operacional del concepto prioridad. Luego, y desde un punto de vista procedimental, existen métodos que procuran un proceso de priorización, a partir de las características intrínsecas que subyacen de cada requerimiento en particular.

De esta manera, se propone una revisión bibliográfica de aquellas líneas de investigación, que abordan la problemática de priorización de requerimientos, desde las perspectivas propuestas. Es decir, el objetivo básico aquí, consta de describir las principales investigaciones desarrolladas en torno al problema de priorización de requerimientos, considerando las perspectivas de abordaje más frecuentes: *Negociación, Cognición y Métodos*.

2.3 Priorización de Requerimientos: Enfoque Cognitivo

Según Nuseibeh y Easterbrook (2000), en el contexto de implementación de un sistema de software, la Ingeniería de Requerimientos (RE) debe proveer un marco conceptual que tienda a explicar de qué manera los *stakeholders* se relacionan entre ellos, en cómo perciben y comprenden la realidad circundante, y cómo la sociología organizacional afecta sus acciones. Como consecuencia de ello, existe un importante componente filosófico dentro de la estructura conceptual de la Ingeniería de Requerimientos (RE). Es decir, debe proveer un marco metodológico aplicable a la interpretación y entendimiento de la terminología utilizada por los *stakeholders*, sus puntos de vista y objetivos. Luego, la RE debe preocuparse por la comprensión de las creencias de los *stakeholders* (epistemología), la cuestión de lo que es observable en el mundo (fenomenología), y finalmente, respecto de la cuestión de lo que objetivamente resultaría más certero (ontología).

Como consecuencia de lo descripto, la RE se nutre de las Ciencias Sociales y Cognitivas a los efectos de proveerse de un cuerpo de conocimiento común, tendiente a satisfacer las inquietudes fundamentales que subyacen de la interacción humana en el contexto de ejecución de los procesos básicos de la RE: Elicitación, Especificación y Validación de los Requerimientos [Loucopoulos, Karakostas (1995)]. Por su parte, la Psicología Cognitiva ofrece un marco conceptual, por intermedio del cual, se podría comprender el porqué del acontecimiento de ciertas dificultades, cuando las personas intentan describir sus necesidades [Posner, (1993)].

Tal cual proponen Nuseibeh y Easterbrook (2000), los expertos en el dominio del problema, poseen usualmente un gran caudal de conocimiento tácito que no es susceptible de introspección, por lo tanto, cuando los usuarios responden a las inquietudes planteadas por los Ingenieros en Requerimientos, es probable que dicha respuesta no coincida con su comportamiento efectivo. Por tal razón, Burg (1997) considera el uso de Técnicas y Herramientas Lingüísticas para analizar el patrón de comunicación organizacional durante la ejecución del proceso de elicitación de requerimientos de software.

Cuando los *stakeholders* o el colectivo social de involucrados, resultan indagados respecto de un concepto del dominio, durante el Proceso de Elicitación, es factible que experimenten una serie de reacciones intelectuales, fácticas y emocionales en torno al objeto en cuestión. En efecto, los atributos identificados para un concepto en particular, varían en forma continua en una escala definible, en la cual, en un extremo se encontrarán aquellos rasgos esenciales para la definición del significado (fuerte

reconocimiento social) y en el otro, aquellos atributos secundarios o accesorios al concepto central (bajo reconocimiento social) [Vivas, **Azzolini**, Vivas, (2010); Vivas, Pazgón, **Azzolini**, (2010)].

La mayoría de los procesos en Ingeniería de Software, tienen la característica de requerir un intensivo esfuerzo de comunicación entre un conjunto dado de sujetos. Luego, y atendiendo a la complejidad que implica su gestión eficaz, Martínez Carod y Cebrich (2007), proponen la aplicación de técnicas propias de la Ciencia Cognitiva, con el objeto de mitigar las disfunciones comunicativas que se tendrían a lugar, luego de una experiencia interactiva entre sujetos, cuyos aspectos psicosociales, difieren significativamente entre sí.

Asimismo, y tal cual lo manifiesta Martínez Carod (2007), en los procesos de elicitación de requerimientos, los *stakeholders* detentan diferentes expectativas con respecto al sistema a desarrollar. En efecto, las mismas se encuentran legitimadas por los intereses diferenciales de los sujetos intervinientes, sus conocimientos previos y necesidades puntuales. Así, los factores personales determinan la percepción de los *stakeholders*, con lo cual, se facultan interpretaciones diferenciales en torno a los requerimientos de un futuro sistema.

Enunciada la complejidad del fenómeno anterior, es deducible que un *stakeholder* asigne un valor preferencial a un requerimiento particular luego de su validación y especificación. En efecto, como se establece en Martín (et. al, 2003) y Martínez Carod y Cebrich (2005), la mayoría de los enfoques utilizados en la priorización, ignoran en gran parte los *aspectos socio-cognitivos* de los sujetos intervinientes.

En esta línea de pensamiento, Martínez Carod (2005) propone la elaboración de un modelo que incluye el *aspecto cognitivo* dentro del proceso de priorización. En efecto, las variables operacionales escogidas en su trabajo, son las siguientes: *Conocimiento del Requerimiento por parte del Participante*, *Categorización del Individuo* y *Valor Asignado*. La primera de ellas, alcanza 4 dimensiones “Sin Conocimiento”, “Poco Conocimiento”, “Conocimiento Suficiente”, “Experto”. La segunda, considera la Jerarquía del Individuo dentro de la Organización, o pondera un peso específico para aquellos *stakeholders* ajenos a la misma. Por último, la tercera variable tiene una escala numérica de tipo entera cuyo rango de variabilidad se extiende de -9 a 9. La misma, recoge la preferencia del sujeto en torno al Requerimiento en cuestión, considerando que cuando el valor es negativo significa que la implementación de dicho requerimiento influye de manera negativa en la determinación general del sistema.

Por último, la autora establece que la estrategia presentada, considera la diferencia de evaluación, respecto de las preferencias relevadas por los *stakeholders* a partir de la misma interpretación de los requerimientos. Asimismo, uno de los objetivos perseguidos, es la determinación de la menor cantidad de requerimientos que resultarían necesarios de implementar, a los efectos de satisfacer lo prioritario de un desarrollo.

Relacionado con la línea teórica precedente, se encuentra un trabajo elaborado por Nuseibeh (et. al., 2003), el cual establece que el desarrollo de los sistemas complejos conlleva necesariamente la interacción de un conjunto de *stakeholders*, los cuales, detentan perspectivas diferenciales respecto del problema a ser tratado, y en consecuencia, de la importancia que adquiere ciertos ítem o requerimientos para su resolución. En efecto, proponen la implementación de un *framework* denominado “*Punto de Vista*”, el cual proporciona una infraestructura para captar y organizar el conocimiento respecto del desarrollo del software, con el fin básico de establecer un objeto que contenga el conocimiento transversal de la notación, proceso y dominio del discurso, desde la perspectiva de un actor particular o grupo. De esta manera, se establece una colección estructurada de conocimiento, que administrada coordinadamente, permitirá establecer un nudo focal para el control de coherencia entre los “puntos de vista” provistos por los *stakeholders*.

El *framework* descrito, conlleva implícitamente un método de negociación a ser emprendido por parte de los *stakeholders* una vez reunido el conocimiento colectivo del dominio especificado. En efecto, y tal cual como se manifiesta en Kaiya (et. al, 2005), dicho método resulta efectivo para resolver el conflicto de ausencia de requerimiento. Según el mismo autor, se denomina a tal conflicto, cuando existen grupos de *stakeholders* que suponen la existencia de un requerimiento, mientras que otro grupo, no lo pondera. La solución podría devenir, sí el grupo que pondera tal requerimiento, revela las necesidades que fundamentan su implementación. De esta manera, el grupo que no lo ha considerado inicialmente, puede incorporarlo dentro de sus preferencias.

Si se considera el citado trabajo de Kaiya (et. al. 2005), aparte del conflicto de ausencia, los autores describen el problema de la inconsistencia y la discordancia como una serie de fenómenos, que son experimentados cuando 2 o más *stakeholders*, exhiben una percepción diferencial respecto a la constitución, evaluación e interpretación de los requerimientos de un sistema. Por su parte, el problema de la inconsistencia ha motivado el uso de diferentes métodos resolutivos. Por ejemplo, en Easterbrook y Chechik (2001) se propone que la especificación de requerimientos sea definida mediante una maquina de transición de estados, y que, las inconsistencias relevadas entre dos transiciones sean

formalizadas mediante la implementación de un sistema basado en lógica difusa. Asimismo, en Blanc (et. al., 2008) se propone la representación de un meta-modelado basado en operaciones de construcción de modelos de secuencias, en las cuales, se utilizan restricciones lógicas para definir reglas de inconsistencias.

Finalmente, y considerando el fenómeno de la discordancia, en Kaiya (et. al. 2005) se establece que puede presentarse en dos perspectivas. Una podría devenir de un esquema diferencial en la evaluación de un requerimiento, mientras que, la segunda alternativa, podría presentarse producto de una discordancia interpretativa. En ambos casos, la brecha de conocimiento entre los grupos intervinientes, representa el factor clave que fundamenta dicho fenómeno.

Como se ha podido apreciar, existe un desarrollo alternativo de líneas teóricas inherentes al Proceso de Priorización, las cuales se basan exclusivamente en los aspectos socio-cognitivos de los *stakeholders*. Luego, sí los aspectos de priorización adquieren una significación diferencial dependiendo del participante, entonces, el análisis de su proceso cognitivo fundamentará el porqué de tales diferencias.

En el Capítulo 1, cuando se han descrito las actividades comunes en la Negociación de los Requerimientos ("*Requirement Negotiation*"), se ha establecido que el Ingeniero de Software deberá promover y facilitar el consenso entre los *stakeholders* a fin de resolver los conflictos que pueden tenerse a lugar, cuando se analiza un conjunto de requerimientos [SWEBOOK, (2004)]. Luego, los enfoques desarrollados aquí, podrían proveer un marco conceptual que explicaría el porqué de la existencia de tales conflictos.

2.4 Priorización de Requerimientos: Enfoque Negociación

Según Lehtola y Kauppinen (2004), el enfoque de Negociación como alternativa de Priorización de Requerimientos, se basa en la idea fundamental de un acuerdo continuo y enriquecido por parte de los diferentes *stakeholders*. Asimismo, y dependiendo de cada enfoque en particular, el proceso en sí, conllevará ciclos de cesión, abandono y renegociación de posiciones entre los diferentes involucrados. Luego, y en forma implícita, tal proceso denota un esfuerzo de comunicación considerable. En efecto, y tal cual lo manifiestan Nuseibeh y Easterbrook (2000), la brecha comunicativa entre los grupos de *stakeholders*, son consecuencia de la diversidad de objetivos, intereses y prioridades de los mismos. Por su parte, en Damian (et. al. 2008) se sugiere que el proceso de negociación de requerimientos, será más efectivo a medida que los *stakeholders* compartan un entendimiento común del dominio, y además, promuevan una comunicación cuyo contenido sea semánticamente válido y aceptado por los diferentes sujetos involucrados.

Lograr una negociación efectiva de requerimientos, por parte de conjunto dado de *stakeholders*, quienes detentan diferentes roles, responsabilidades y prioridades, en una etapa temprana en el ciclo de vida del software, constituye un factor clave en el éxito del Proyecto [In y Roy (2001); Boehm (et. al., 1995); Boehm e In (1996); In (et. al. 2001). Al establecer una negociación colectiva, se intenta establecer una visión compartida respecto del Proyecto en cuestión.

De esta manera, y a lo largo de estos últimos años, se han desarrollado una serie de metodologías de negociación, tendientes a facilitar las acciones colaborativas de los *stakeholders*, con el fin de promover una visión global del proyecto. A modo de ejemplo, el trabajo de Boehm y Ross (1989) establece que los proyectos de software necesitan satisfacer en forma simultánea, una serie de objetivos que son relativos a los usuarios, clientes, el equipo de desarrollo, mantenimiento, entre otros involucrados. Así, elaboraron la Teoría “W” la cual se basa en la idea de elicitar argumentos de tipo “win” entre los *stakeholders*. Es decir, tareas y condiciones inicialmente óptimas para cada uno de ellos, que luego y por intermedio de un proceso iterativo de negociación, serán cedidas, fortalecidas o renegociadas entre la totalidad de los involucrados. Asimismo, en Boehm (et. al., 1995) se fusiona la Teoría “W” junto al modelo de espiral Boehm (1988).

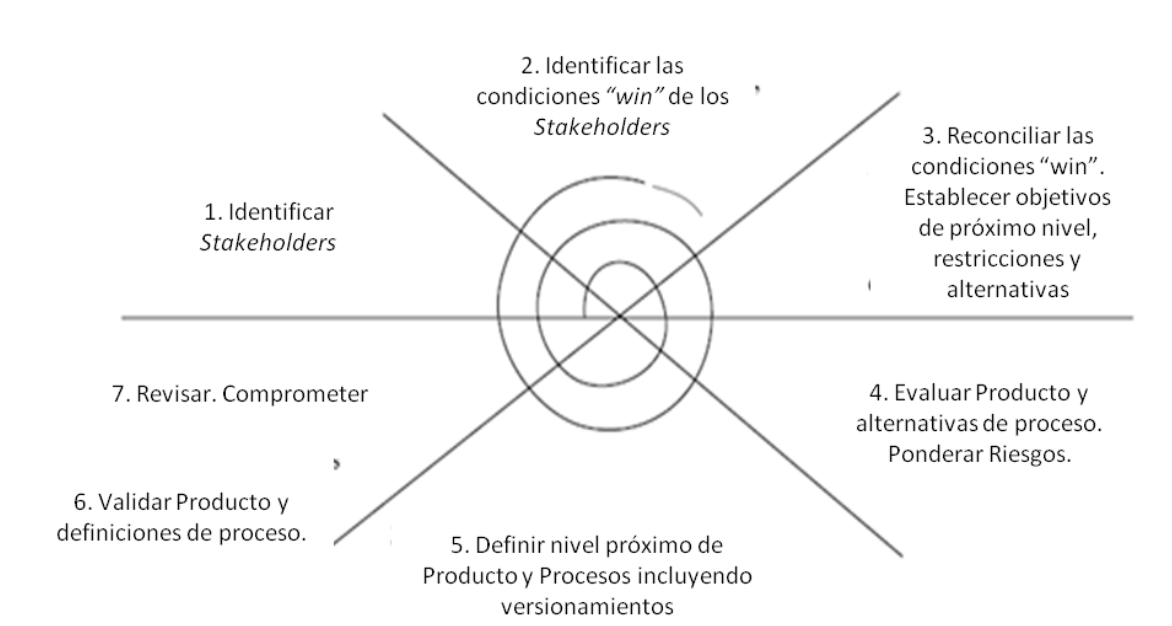


Figura 2.1: "Fusión de la Teoría W y Modelo Espiral". Fuente: Boehm (et. al., 1995). Traducido.

Como se aprecia en la Figura 2.1, la articulación de los enfoques permite construir escenarios factibles de negociación y renegociación en etapas tempranas del desarrollo del software. Los rangos distintivos de esta vinculación, podrían ser ejemplificados de la siguiente manera:

- Se establece un conjunto explícito de objetivos: identificación de *stakeholders* y de sus condiciones "win". Luego se establece un proceso de conciliación entre las partes interesadas para la definición y colaboración de alternativas de producto.
- Se incorpora de forma explícita las actividades de colaboración dentro del proceso del ciclo de vida del software. Considerando en efecto, la resolución de riesgos, versiones del producto, entre otros.

Por su parte, en Ruhe (et. al, 2002) extienden el modelo planteado en Boehm (et. al., 1995), estableciendo un enfoque cuantitativo a los efectos de promover una mejor racionalidad en las decisiones. Al igual que el método base, el presente considera la interacción como un proceso que permite aumentar el nivel de conocimiento de los *stakeholders* respecto de los requerimientos de un sistema. No obstante, implementa el método AHP (*Analytic Hierarchy Process*) a los efectos de cuantificar las preferencias de los involucrados. Luego, dicho resultado es recombinado por el método de estimación

GENSIM [Anderson y Evans, 1995], con el fin de establecer la viabilidad del subconjunto de requerimientos alternativos. De esta manera, el método presentado permite decidir respecto de los requisitos que son preferidos por los involucrados, teniendo en consideración, el esfuerzo de desarrollo y el valor empresarial aportado.

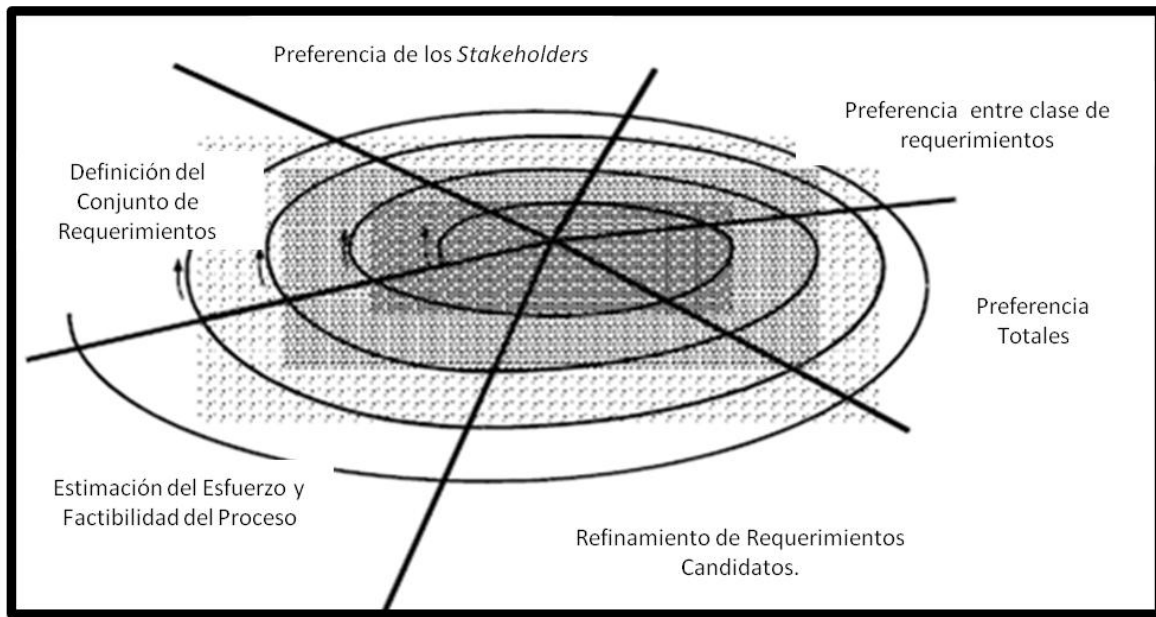


Figura 2.2: “Preferencia de los Requerimientos en base al Esfuerzo de Desarrollo” Fuente: Ruhe (et. al. 2002). Traducción.

En la Figura 2.2, el método en cuestión propone un esquema análogo al presentado por Boehm (et. al., 1995). Sin embargo, en Ruhe (et. al. 2002) se considera especialmente la viabilidad y la estimación del esfuerzo como criterio para el establecimiento de requerimientos alternativos.

Por su parte, un trabajo realizado en In (et. al., 2002) promueven una nueva extensión al modelo desarrollado por Boehm (et. al. 1995). En efecto, el modelo MPARN (*Multi-Criteria Preference Analysis for Semantic Requirements Negotiation*) permite elicitar condiciones “win” entre los participantes, a través de una serie de pasos:

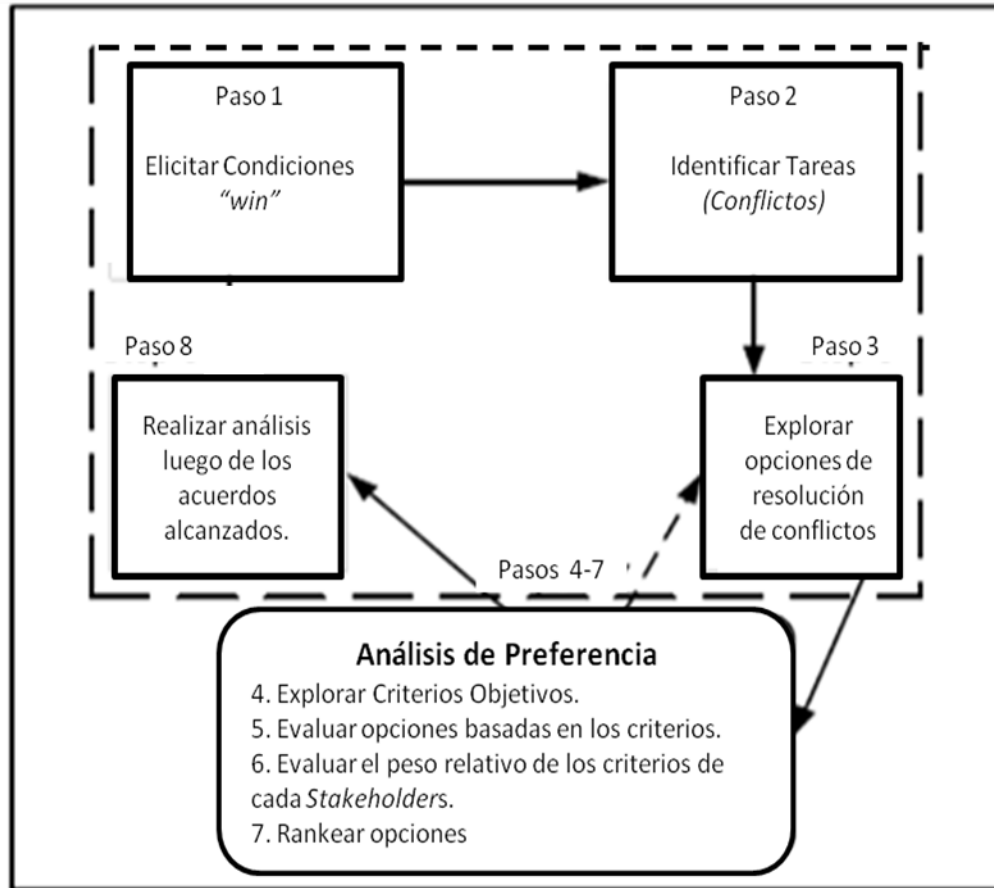


Figura 2.3: "Análisis de Preferencia Multicriterio para la negociación semántica de los requerimientos MPARN (Multi-Criteria Preference Analysis for Semantic Requirements Negotiation)". Fuente: In (et. al. 2002). Traducción.

Como se aprecia en la Figura 2.3, el modelo extendido pretende mejorar los niveles de cooperación entre los *stakeholders* a los efectos de satisfacer acuerdos desde una perspectiva más racional entre los participantes. En consecuencia, se intenta formalizar lo que en el modelo original, se establecía de manera *ad-hoc*. Los pasos del modelo, son:

Paso I: Cada *stakeholder* identifica sus condiciones *win*. Este paso proporciona las bases características de un proyecto ideal, según las percepciones de los interesados.

Paso II: Las listas establecidas por los *stakeholders*, son revisadas a los efectos de detectar posibles problemas o conflictos con el fin de categorizarlos.

Paso III: Los *stakeholders* pueden ganar comprensión mediante la revisión del Paso II. De esta manera, podrán establecer opciones de resolución para aquellos conflictos o problemas revelados.

Paso IV: Se comienza a identificar funciones de preferencias mediante la implementación de criterios de importancia.

Paso V: Una vez relevados los criterios de importancia, éstos se listan y los *stakeholders* los ponderan según su percepción.

Paso VI: Se obtienen los pesos relativos de los criterios relevados, según la evaluación de los involucrados.

Paso VII: Se ordenan los criterios por el orden de mérito asignado.

Paso VIII: Se trata de alcanzar la convergencia, mediante la implementación de una solución satisfactoria para la totalidad de los *stakeholders*.

Finalizados los pasos precedentes, se espera que la vinculación entre el modelo de negociación “*Win-Win*” y el MPARN otorgue un marco de discusión basado en un esquema de preferencia cuantificable. Luego, y de acuerdo con la evidencia otorgada por los autores, la propuesta descrita promete incrementar los niveles de cooperación y confianza entre los *stakeholders*, promoviendo así, un mejor proceso de negociación entre los requerimientos disponibles.

En correlato a las metodologías de negociación de requerimientos, un estudio empírico elaborado por Damian (et. al., 2008) establece que las discusiones asincrónicas de los *stakeholders*, son útiles en cuestiones relacionadas a la incertidumbre de los requerimientos. De esta manera, un proceso ulterior de negociación sincronizada, permitirá focalizarse en la eliminación de ambigüedades en los requerimientos. Nuevamente, y en base a los postulados señalados previamente, la implementación de un proceso iterativo entre los *stakeholders*, promovería un mejor conocimiento del dominio de la aplicación, y en consecuencia, la ambigüedad e incertidumbre asociada a la selección de requerimientos, quedaría disminuida.

Por su parte, un estudio elaborado por Fricker y Grünbacher (2008) propone la elaboración de un marco, el cual se basa en la idea de establecer una serie de taxonomías de negociación configuradas en constelaciones. La idea fundamental del presente, consiste en otorgar una serie de consejos inherentes a la negociación táctica y metodológica entre los diferentes *stakeholders*, los cuales detentan diferentes roles y por tanto, intereses contrapuestos dentro de una misma Organización.

Tal cual lo manifiestan los autores, existen dos formas básicas para promover un proceso de negociación. En primera instancia, un proceso de negociación podría llevarse a cabo por un Ingeniero de Requerimientos, el cual elicitaba posiciones y perspectivas entre

los diferentes involucrados, con el fin de documentarlas y explicitarlas en un modelo tentativo de requerimientos. Finalmente, dicho modelo se comunica con el objeto de resolver eventuales conflictos, promoviendo así, un acuerdo colectivo. En segundo lugar, podría considerarse que un proceso de negociación, podría llevarse a cabo por grupos específicos de *stakeholders*. Es decir, de acuerdo con la función que desempeña el sujeto dentro de la organización, será factible segmentarlos en grupos homogéneos, a los efectos de promover negociaciones focalizadas o distribuidas en torno a una especificación determinada (constelación).

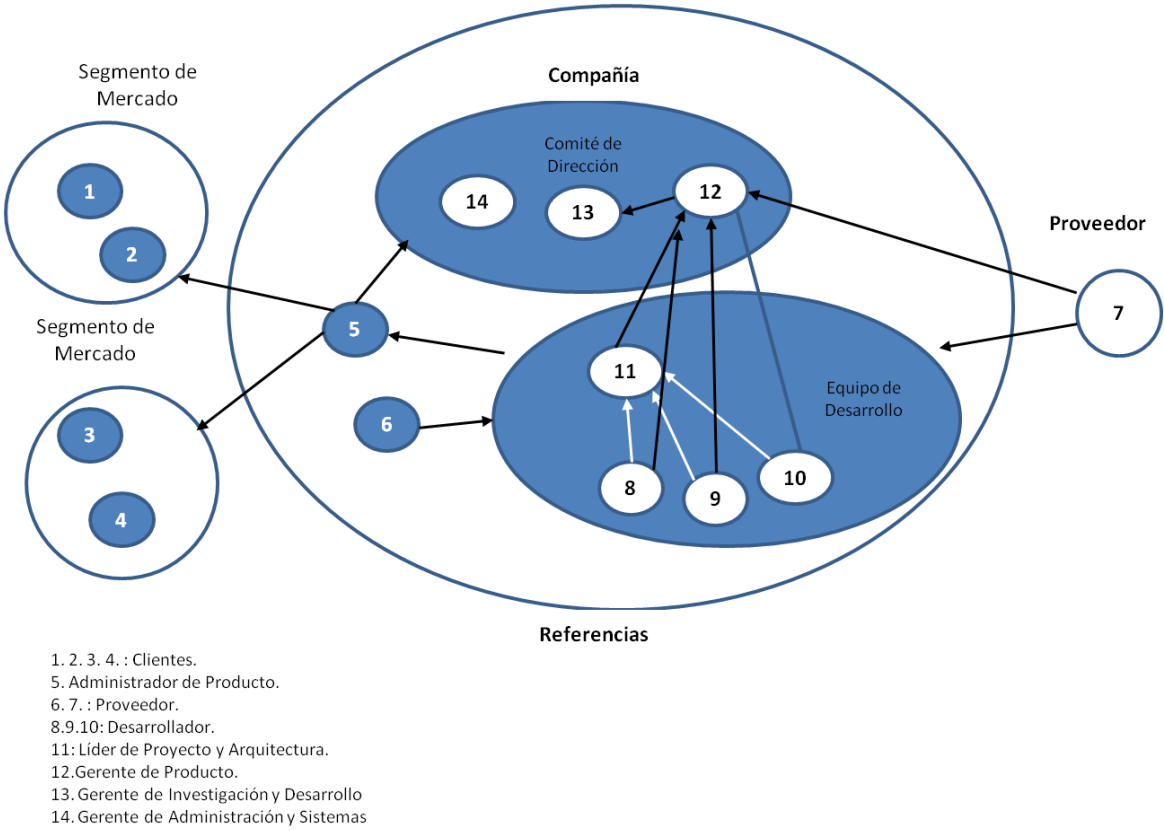


Figura 2.4: “Constelaciones de Negociación”. Fuente: Fricker y Grünbacher (2008). Traducción.

Como se aprecia en la Figura 2.4, Fricker y Grünbacher (2008) establecen un modelo de contrato por intermedio del cual, los diferentes grupos segmentados inician un proceso de negociación focalizado. De esta manera, las negociaciones se tornan sobre criterios o especificaciones particulares cuyos actores involucrados, detentan un conocimiento amplio y certero sobre el tópico que los reúne. Asimismo, el presente modelo trae aparejado la necesidad de satisfacer los siguientes objetivos:

- Conceptualizar correctamente la constelación de la negociación.
- Conocer las ventajas y limitaciones de la constelación configurada.
- Conocer las tácticas de negociación y los métodos adecuados para la constelación.
- Identificación de los actores que deben participar en la negociación.
- Seleccionar y continuar con el enfoque de negociación más adecuado.

De esta manera, el *framework* desarrollado por los autores, pretende responder tales inquietudes por medio de la aplicación de una serie de enfoques y métodos aplicados a cada una de las constelaciones configuradas. A continuación, y en base a la Figura anterior, los autores efectúan una serie de recomendaciones en función de las siguientes constelaciones (Figura 2.5):

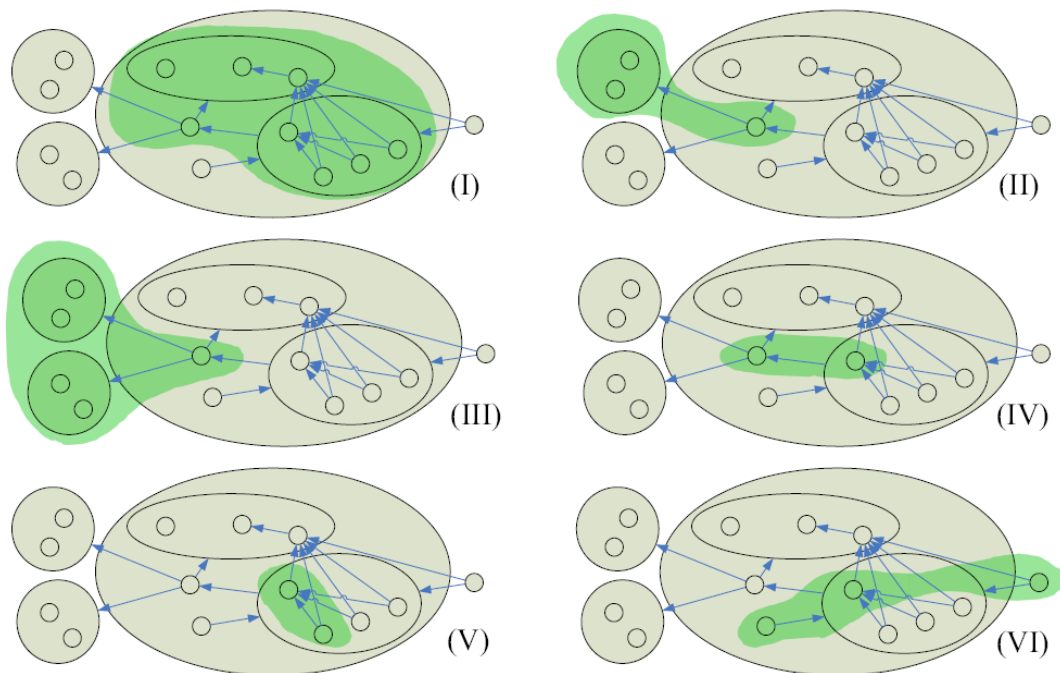


Figura 2.5: “Mapas de Constelaciones” Fuente: Fricker y Grünbacher (2008)

Constelación I: El líder del Proyecto, quien es miembro del equipo de Desarrollo, tiene la responsabilidad de establecer los lineamientos arquitectónicos.

Constelación II: El Líder del Producto es el responsable de comprender el mercado de la compañía e identificar los requerimientos que mejor satisfacen las necesidades de los clientes.

Constelación III: Alternativamente, el Líder del Producto puede identificar múltiples segmentos de mercado con grupos diferenciados de clientes. En esta situación, se constituye como el responsable que deberá implementar aquel enfoque de la Ingeniería de Requerimientos, que mejor se adapte a la variabilidad de las necesidades desprendidas en cada segmento en particular.

Constelación IV: Pasado un tiempo, el Líder del Producto ha producido una Especificación de Requerimientos de Software (SRS) en conjunto con el Líder de Proyecto y el Equipo Desarrollador. Luego, el Líder de Proyecto y el Líder de Producto establecen un marco de negociación para la propuesta de implementación y promueven un ciclo interactivo.

Constelación V: En el avance del proyecto, el Líder de Proyecto distribuye actividades y tareas entre los miembros del equipo de desarrollo, en función del acuerdo previo establecido respecto a la Arquitectura del Software (Constelación IV).

Constelación VI: Finalmente, y en función de las características del Producto, se establece un marco de negociación entre Proveedores y los Líderes de Proyecto y Producto. Teniendo en consideración, cuestiones tecnológicas, de costos y de cronograma.

Por su parte, Fricker y Grünbacher (2008) establecen que el presente enfoque, permitirá establecer cuáles serán los *stakeholders* claves en cada uno de los procesos y en función de ello, comprender el marco de negociación que se deberá establecer dado el rol y las características de los intervinientes afectados. En cuanto a las limitaciones del presente, establecen que las habilidades de negociación de los líderes, deberían ser especialmente incentivadas en ellos, a los efectos de satisfacer el cumplimiento de los objetivos propuestos transversalmente.

En una línea de pensamiento similar, el trabajo de Lu y Jing (2009) propone un enfoque de negociación socio-técnico (ASTN) a partir de la integración del *framework* STCP [*Socio-Technical Co-construction Process*, Lu (1999)] y el Modelo ABNP [*Argument-Based Negotiation Process*, Toulmin (1958); Chang et. al. (1995); Sillence et. al. (1999)]. Si se parte de la concepción que la construcción de un Software, implica un trabajo colaborativo por parte de un grupo de *stakeholders*, y que cada uno de ellos, detenta diferentes perfiles formativos, experticia e intereses, entonces, es fácil suponer que

existirá una variedad de percepciones respecto del objeto en cuestión, que serán necesarias de conciliar, a los efectos de cumplimentar las actividades planificadas en el ciclo de vida del software. De esta manera, los autores desarrollan el enfoque basado en una negociación socio-técnica, involucrando 3 fases (Figura 2.6):

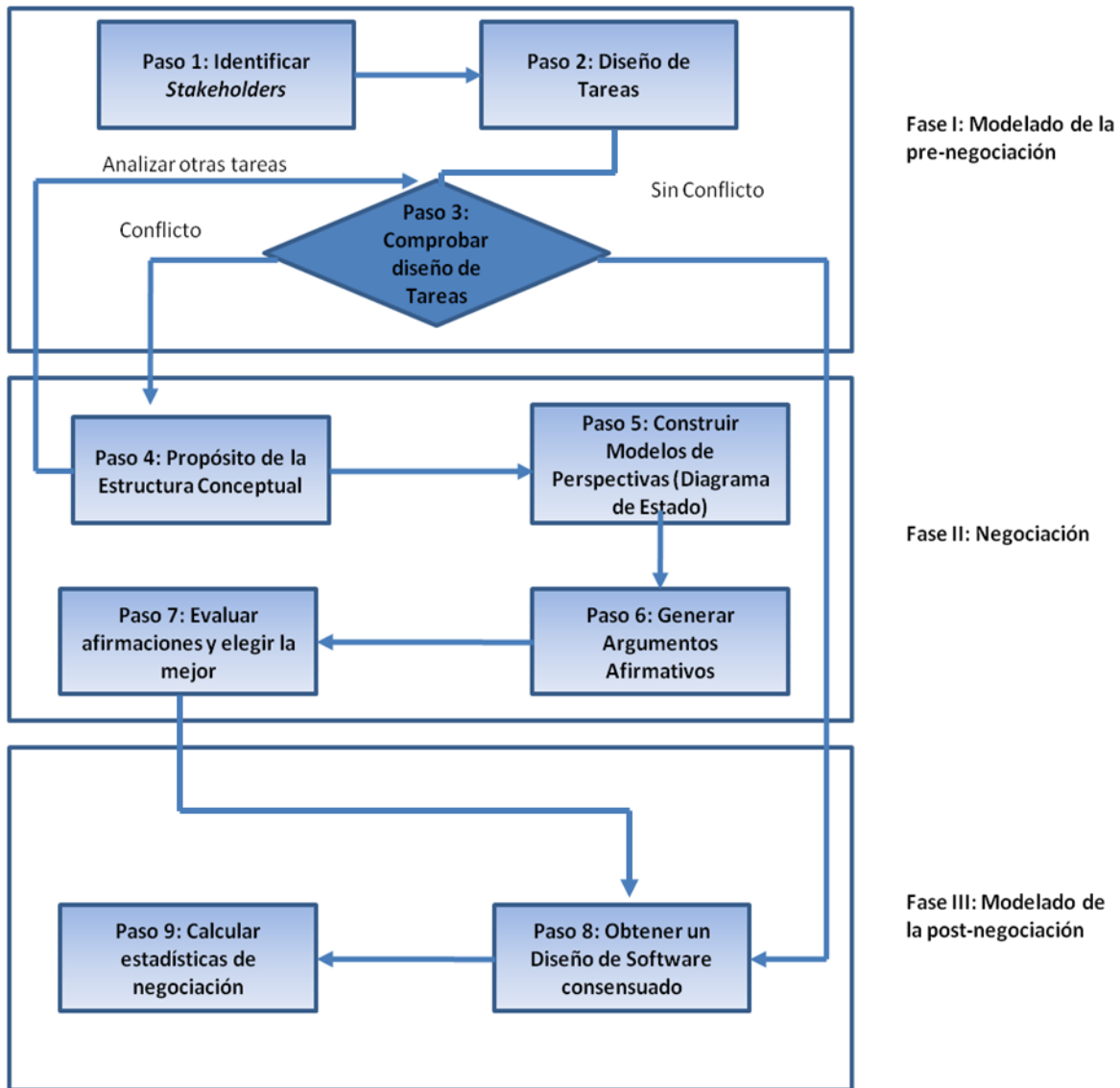


Figura 2.6 “Enfoque ASTN [A Socio-Technical Negotiation approach] ” .Fuente: Lu y Jing (2009). Traducción.

Fase I: Modelado de la Pre-Negociación: El objetivo de esta fase, consta de la identificación de conflictos potenciales, chequeando las diferencias entre las tareas propuestas. Luego, se intenta ayudar a los *stakeholders* a que organicen sus argumentos de cara a la negociación de conflictos. En los pasos I, II y III se aplica el *framework* STCP creado por Lu (1999).

Fase II: Negociación: Los participantes son guiados para negociar entre ellos. Al efecto, se parte de una estructura conceptual en donde se identifican perspectivas dinámicas a partir de la interacción de los participantes. Luego, se generan una serie de afirmaciones argumentativas que serán evaluadas y ponderadas en un orden de mérito. Para el presente proceso de negociación, se aplicará el Modelo ABNP [Chang (et al., 1995); Jennings (1998); Sillince (et al., 1999); Amgoud (et al., 2000); Avery et al., 2001; Kraus, 2001 y Rong et al., 2002)] hasta lograr un nivel de interacción que implique un mutuo consenso.

Fase III: Post-Negociación: En la presente, los *stakeholders* ya han resuelto todos los conflictos identificados y han aceptado en forma conjunta, el diseño del software a implementar. Es decir, han acordado el desarrollo de las actividades planificadas que son inherentes a la especificación de los requerimientos del producto y del diseño del mismo. Tales circunstancias, implican la elaboración de una estructura, en la cual, los contenidos vertidos se hallan socialmente compartidos gracias a la aplicación del *framework* ASTN. Dado que tal estructura, puede explicar en forma clara las especificaciones y diseño del producto, esta experiencia deberá ser capitalizada a los efectos del aprendizaje y reutilización del conocimiento en otros equipos de trabajo. En consecuencia, el último paso del *framework*, implica la elaboración y resumen de las estadísticas de colaboración. Por ejemplo, la cantidad de tiempo o interacciones que se tuvieron a lugar como consecuencia de la resolución de un conflicto dado, entre otras. Finalmente, la aplicación de tales estadísticas, se constituyen como un factor importante que permitirá estimar los esfuerzos dedicados en los procesos de negociación en tales tópicos.

Finalmente, los autores concluyen que el presente *framework* (ASTN) incorpora una dimensión social en los procesos técnicos de especificación y diseño, gracias a la interacción propuesta entre el *framework* STCP y el modelo ABNP. Asimismo, consideran que el *framework* propuesto se constituye como una práctica más comprensiva para el abordaje de las problemáticas inherentes a los procesos de negociación y resolución de conflictos.

Por su parte, Fricker (2009) propone un modelo basado en la Teoría de Redes y Negociación para analizar y diseñar el flujo de requerimientos a través de un ecosistema

de software. Precisamente define el concepto de “ecosistema”, como una red de *stakeholders* interdependientes que se encuentran en mercados distribuidos y son parte de grandes Proyectos de Software. Basándose en el cuerpo de conocimiento de las Redes de Computación, analiza las características de los nodos y arcos de la estructura ecosistémica incluyendo las propiedades de la comunicación de los requerimientos. Finalmente, el autor advierte la necesidad de comprender como la red puede ser modelada, cuando los *stakeholders* pertenecen a diferentes grupos, y además, no se encuentren organizados en una misma estructura jerárquica.

En correlato con las líneas precedentes, en Gaur (et. al., 2010) se reafirma que los *stakeholders* adquieren diferentes perspectivas, intereses y prioridades en la implementación de varios “agentes” (Paradigma Orientados a Agentes). Asimismo, los requerimientos elicitados a los *stakeholders* a menudo representan un conocimiento vago e impreciso del dominio. Con lo cual, proponen un modelo de decisión difuso denominado “Modelo de Hacedor de Decisión Multi-Persona” (MPDMM) a los efectos de integrar y negociar los requerimientos de varios *stakeholders*. El modelo en cuestión, se enfoca en el problema de priorizar y decidir cuáles agentes serían viables de implementar, ante la restricción que impone los límites de tiempo, presupuesto, entre otros.

2.5 Priorización de Requerimientos: Enfoque Métodos

Se han desarrollado a lo largo de estos últimos años, diversos métodos que permiten iniciar un proceso de priorización, a partir de la valorización particular o conjunta de una serie de requerimientos implementables. De esta manera, y según Motupally (2008) tales métodos pueden clasificarse en dos grandes grupos, según la escala de medida que utilizan para efectuar las ponderaciones de prioridad: *Métodos de Escala Ordinal*, y los *Métodos que usan la Escala de Ratios*. Cabe destacar que los métodos que a continuación se describen, son los más utilizados en el campo de la Priorización de los Requerimientos de Software.

2.5.1 *Métodos de Escala Ordinal*: Procuran asignar un requerimiento en particular a una categoría conceptual de prioridad. Se considera además, que el mismo grupo de requerimientos contenidos en la misma categoría conceptual, mantienen su mismo grado de prioridad. Dicho grupo puede estar compuesto de uno o más elementos. Los métodos más relevantes de la clasificación, son:

2.5.1.1 *“Asignación Numérica”*: Es un método de fácil uso creado por Brackett (1990) y recomendado posteriormente por el estándar 830-1998 IEEE. El mismo, se basa en clasificar a los requerimientos en tres categorías conceptuales: “Obligatorios”,

“Deseables”, “No esenciales”. De esta manera, se indaga a los *stakeholders* respecto de la “importancia percibida” en cada uno de los requerimientos especificados, utilizando la siguiente escala (Tabla 1):

Categoría de Prioridad	Escala Numérica
No esencial	1
Deseable	2
Obligatorio	3

Tabla 1: “Escala de Priorización de Requerimientos Ordinal-Conceptual”. Fuente: Motupally (2008) y Ma (2009)

Existen otros métodos que procuran una variación en la escala señalada en la Tabla 1, incorporando más grupos de prioridad como el caso de MoScow [citado en Ma (2009), Hatton (2007, 2008)]. Sin embargo, el método base no deja de presentar la misma metodología que “Asignación Numérica”, por lo tanto, el resto es considerable como extensiones básicas del mismo [Leffingwell y Widring (2000), Sommerville y Sawyer (1997)]. Por su parte, estudios empíricos demuestran que aproximadamente más del 80% de los *stakeholders* tienen asignar a los requerimientos, en la categoría “Obligatoria” y menos del 5% en la categoría “No esencial”, con lo cual, se pierde efectividad a los efectos de discernir los verdaderos requerimientos prioritarios [Berander (2004)].

2.5.1.2 Ordinal (“Ranking”): Como el método anterior, el presente procura una ordenación de los requerimientos pero sin agruparlos en categorías. Es decir, se ordenan de 1 a N considerando el nivel “1” como el requerimiento más importante, y el “N” como el de menor importancia [Ma (2009), Hatton (2008)]. Dada su naturaleza, el presente método procura que cada requerimiento pertenezca a una categoría de prioridad particular. En consecuencia, no permitirá establecer la diferencia relativa que existirá entre dos o más requerimientos, con lo cual, la priorización relativa resulta inexistente [Berander, (2004)].

2.5.1.3 Burbuja (“Bubble Sort”): El presente método, procura una ordenación de los requerimientos, tomándolos de a dos a la vez. Se le solicita a los *stakeholders* que procuren una enésima ordenación de los requerimientos, en caso que su orden de prioridad sea incorrecta según su percepción. Finaliza la ejecución del método, luego de

obtener una lista ordenada y consensuada de los requerimientos [Aho (et. al., 1983), Karlsson (et. al. 1998), Ma (2009)].

2.5.1.4 Árbol Binario de Búsqueda (“Binary search Tree”): Básicamente, el método procura una ordenación en árbol de los requerimientos. El mismo se compone de nodos jerarquizados que indican el nivel de prioridad de un determinado requerimiento. Se efectúan comparaciones entre los requerimientos, del tipo: “*más o menos importante*” a los efectos de cambiar la jerarquía de los nodos [Aho (et. al., 1983), Karlsson (et. al. 1998), Ma (2009)].

2.5.1.5 Planeamiento del Juego (“Planning Game”): Es un método de priorización propuesto por Beck (2000) y ha sido ampliamente utilizado. Inclusive, se constituye como uno de los métodos claves dentro de la Metodología XP (Extreme Programming) [Karlsson, et. al., 2004]. El mismo presenta una combinación de la técnica de asignación numérica y “*Ranking*”. Consta de asignar a los Requerimientos entre tres categorías conceptuales: (1) “Sin ellos, el Sistema no funcionaría” (2) “Son menos esenciales, pero al implementarlos se proveería de un significativo Valor de Negocio” (3) “Sería deseable implementarlos”. Luego de asignar los requerimientos en las tres categorías conceptuales provistas, se realiza un ranking simple en cada una de ellas [Ma, (2009)].

2.5.2 Métodos de Escala Ratio: Se compone de un conjunto de métodos que tienen como objetivo principal, especificar las diferencias relativas entre los requerimientos, tomando como criterio de comparación, la prioridad de cada uno de ellos [Ma (2009)]. La principal diferencia entre esta clase de método y la anterior, consiste en su capacidad de efectuar comparaciones del tipo “*cuanto más importante resulta un requerimiento respecto a otro*”. La escala usualmente se extiende del 0% al 100% [Berander (2004), Motupally (2008)]. Los métodos más representativos de la categoría, son:

2.5.2.1 Método de los 100 dólares (“Hundred Dollar Method”): El presente método, también denominado Voto Acumulado (“*Cumulative Voting*”) consiste en asignar una suma hipotética de dinero a cada *stakeholders*, para que cada uno de ellos procure una asignación prioritaria del “dinero” hacia el conjunto de requerimientos disponibles [Leffingwell y Widring (2000), Berander (2004), Ma (2009)]. En función de las asignaciones individuales, se tabulan las distribuciones en un diagrama de frecuencias relativas, indicando en consecuencia, el nivel de prioridad de cada uno de los requerimientos (en porcentaje). Uno de los problemas que se le han imputado a este método, se refiere a la extensión del conjunto de requerimientos que han de ser analizados, con lo cual, Regnell

(et. al. 2001) ha comprobado la eficacia de incrementar “la cantidad de dinero distribuible” a los efectos de poder valorar conjuntos de requerimientos más extensos.

2.5.2.2 Proceso de la Jerarquía Analítica (AHP: “Analytic Hierarchy Process”): El método AHP fue desarrollado por Saaty (1980; 1986; 1990; 2001) y se ha considerado como uno de los enfoques más utilizado cuando resulta necesario tomar una decisión, respecto a la selección y priorización de un determinado conjunto de alternativas, teniendo en consideración, el juicio provisto por un conjunto de sujetos [Kumar y Vaidya, (2006), Shiliang, Shibo (2004), Rodríguez Bello (2007)].

El método representa un enfoque de decisión multi-criterio, en el cual, cada factor de incidencia relativo a un proceso decisivo, se encuentra estructurado en una Jerarquía [Saaty (1990)]. Asimismo, la aplicación del método requiere la satisfacción previa del siguiente conjunto de actividades:

- Representar el Problema: Implica un estudio sistemático y minucioso del dominio de aplicación, sin llegar a considerar aspectos triviales que puedan afectar la sensibilidad en su interpretación.
- Considerar aspectos circundantes del ambiente externo del problema.
- Identificar tareas o atributos que contribuyen a la solución.
- Identificar los participantes (*stakeholders*) asociados al problema.

El fundamento del método, se basa en tres principios básicos: una estructura, una medida en escala de ratios y una síntesis [Saul (et. al., 2001), Rodríguez Bello (2007)]. A partir de estos elementos, un problema específico puede ser estructurado en una jerarquía de niveles, lo cual contribuye a una mejor comprensión humana respecto de la complejidad del mismo [Gass y Forman, (2001); Saaty, (1986)].

Finalmente, este método fue aplicado por primera vez en el área de Priorización de Requerimientos por Karlsson (1996). Sí bien fue ampliamente criticada la metodología, ya que la existencia de requerimientos dependientes y la inconsistencia en el juicio valorativo de los *stakeholders*, relativizan la robustez de los resultados ofrecidos por el método, se ha convertido en un enfoque que ha evolucionado como un componente metodológico clave dentro de la Priorización de Requerimientos de Software. Evidencia de

ello, puede apreciarse en el trabajo de Sadiq (et. al., 2010), Iqbal (et. al., 2010) entre otros².

2.5.2.3 Jerarquía AHP (“Hierarchy AHP”): Es un método introducido por Karlsson (et. al., 1998) para analizar Proyecto de Software de mediana a gran envergadura. La idea básica del método, consta en estructurar los requerimientos en una jerarquía, en la cual, se contemple el grado de generalización de los requerimientos. Es decir, requerimientos más generales, tendrán un nivel más alto en la Jerarquía que aquellos más específicos. Luego de establecer dicha Jerarquía, se aplicará el método AHP para realizar comparaciones entre pares en cada uno de los niveles jerárquicos establecidos [Davis, (1993)]. Este método, tiene la ventaja de reducir las comparaciones entre pares en forma significativa, dado que no se efectuaría sobre la totalidad de los requerimientos disponibles como en AHP (puro), sino que, por el contrario, se establecerían sobre las segmentaciones que se dan a lugar en cada uno de los niveles de la jerarquía. Su desventaja principal, es la dificultad que implicaría detectar la inconsistencia en los juicios personales [Ma (2009)].

2.5.2.4 Enfoque Costo-Valor (“Cost-Value Approach”): Es un enfoque introducido por Karlsson y Ryan (1997), en el cual, se pondera el grado de prioridad de un determinado requerimiento por su correspondiente costo de implementación. La metodología utilizada por los autores, se basa en el método AHP, a los efectos de realizar comparaciones por pares en los ítems Prioridad/Valor. Finalmente, y según los autores, la estrategia de seleccionar requerimientos prioritarios, consta de establecer aquel conjunto de requerimientos cuya prioridad relativa sea mayor, sujeta a la restricción del costo total de su respectiva implementación.

2.5.2.5 Árbol de expansión mínima (“Minimal Spanning Tree”): Es un método introducido en Karlsson (et. al. 1998) el cual presupone que los *stakeholders* no cometen errores transitivos en sus valorizaciones de prioridad. En consecuencia, se utiliza comparaciones por pares del orden $n - 1$. (Siendo n la cantidad de requerimientos). De esta manera, se reduce notoriamente los esfuerzos de comparación, tomando como eje de referencia, el método AHP. Sin embargo, el método en cuestión, tiene como desventaja principal su incapacidad en el reconocimiento de juicios inconsistentes.

² En la Sección 3.2.1 (Capítulo 3) se profundizará la descripción del método AHP y la reformulación propuesta por Karlsson (1996).

2.5.2.6 Modelo de Priorización de Requerimientos basado en el Mercado (“Market Driven Requirements Prioritization Model”): Es un enfoque desarrollado recientemente por Iqbal (et. al., 2010), el cual se encuentra diseñado para priorizar requerimientos en el contexto de desarrollo de software orientado al Mercado. Según los autores, la priorización de requerimientos “tradicional” se encontraba orientada a ponderar el valor de preferencia de los *stakeholders* del Proyecto de Software, mientras que, la presente metodología, abarca también a competidores del futuro Producto de Software, clientes diversos de la compañía, entre otros.

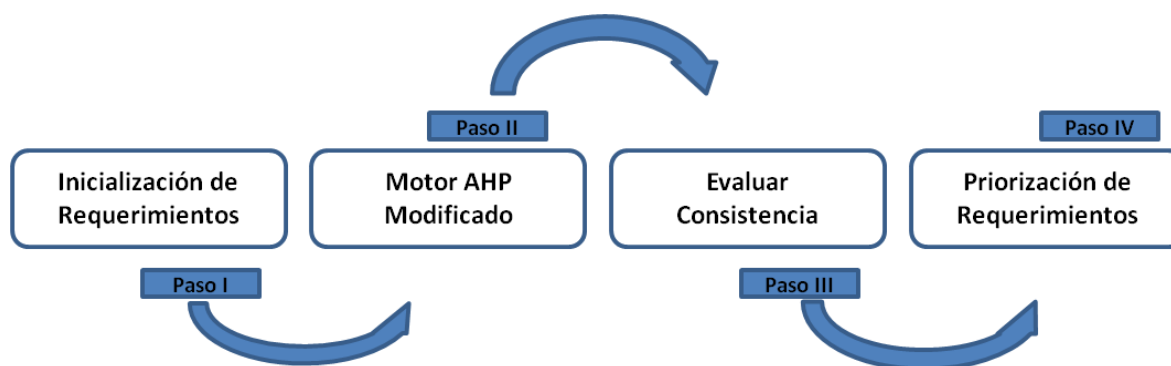


Figura 2.7: “Pasos involucrados en el MDRPM”. Fuente: Iqbal (et. al. 2010). Traducción

El modelo en cuestión (Figura 2.7), se basa inicialmente en 4 pasos. El primero de ellos, denominado “Inicialización de Requerimientos” comprende la etapa de asignar los Requerimientos del Sistema, dentro de las diferentes categorías conceptuales predefinidas por los autores. Las mismas, denominadas “*bin*”, totalizan 35 categorías factibles de clasificación, en las cuales, los mencionados requerimientos son asignados previo a un estudio cuantitativo y cualitativo al “*bin*” correspondiente. Por su parte, el paso 2, denominado “Motor de AHP”, implica un proceso de comparación de a pares, entre los requerimientos asignados a un “*bin*” en particular.

Referencias

1. Seleccionar el "bin" a priorizar.
2. Elegir los Requerimientos ha ser priorizados.
3. Colocar los Requerimientos candidatos en filas y columnas de una Matriz AHP (N x N).
4. Realizar las comparación por pares del punto 3.
5. Normalizar las filas y columnas para estimar el autovalor de la matriz.

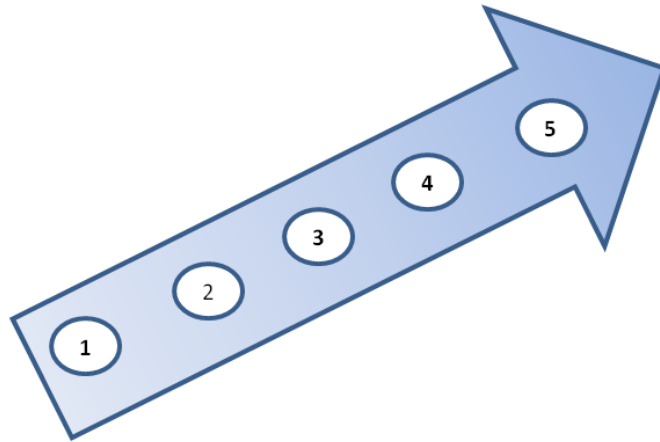


Figura 2.8: "Pasos involucrados en AHP Engine". Fuente: Iqbal (et. al. 2010). Traducción.

Como se puede apreciar en la Figura 2.8, de la totalidad de "bin" que se establecen, se selecciona cada uno en particular (1) y comienza a procesarse por el método AHP los requerimientos contenidos en él (2, 3, 4), luego se obtendrá la matriz normalizada indicando la distribución de las prioridades (5). El presente proceso, se repite tantas veces como cantidad de "bin" existan. La característica más saliente del proceso, es que la cantidad de comparaciones de a pares a efectuar, se reduce prácticamente al 50% comparado con el método original AHP. Finalizado el proceso anterior, se comienza el proceso de verificación del valor que adquiere el Ratio de Inconsistencia, verificando así, la incidencia que adquiere el error en el juicio valorativo de los *stakeholders*. Se toma como umbral de tolerancia, el nivel descrito por Saaty (1980):

$$CR \leq 0,1$$

El último paso del enfoque, denominado "Priorización de Requerimientos", implica un proceso de priorización de "bin". Es decir, en los pasos previos, se ha priorizado los requerimientos contenidos en cada uno de los "bin", obteniendo en consecuencia, una distribución de las preferencias de sus respectivos requerimientos. El último paso, implica un proceso de priorización a nivel de "bin", con lo cual, se obtendrá una matriz de priorización de "bin" y a su vez, cada "bin", tendrá un esquema de priorización de sus propios requerimientos. Finalmente, el presente enfoque de priorización, permite reducir significativamente la cantidad de comparaciones por pares. En consecuencia, y tal cual manifiestan los autores, el presente podría ser utilizado en proyectos de mediana a gran envergadura.

2.6 Consideraciones Finales del Capitulo

El presente Capitulo ha tenido como objetivo básico presentar las líneas de investigación más citadas y actuales, en torno al problema de priorización de requerimientos de software. En efecto, se han agrupado en tres grandes enfoques: **Cognitivo, de Negociación y Métodos**. Asimismo, y más allá de dicha clasificación conceptual, se ha constatado que las propuestas analizadas fusionan en forma implícita o explícita los enfoques mencionados.

Por ejemplo, los Enfoques orientados a la Negociación y aquellos con fuerte sesgo Metodológico, conllevan implícitamente un abordaje socio-cognitivo de los *stakeholders*. En efecto, en cada interacción propuesta en búsqueda del consenso o identificación de conflictos en torno a un esquema de priorización implementable de requerimientos de software, se relevan de cierta manera, las diversas posiciones, intereses y preferencias entre el total de involucrados (Enfoque Negociación). Asimismo, los Métodos descritos pretenden relevar explícitamente el perfil de preferencia de cada *stakeholder*, con lo cual, también se está indagando su aspecto cognitivo.

Capítulo 3

“Marco Teórico y Metodológico de la Investigación”

3.1 Introducción

El presente Capítulo tiene como objetivo básico reunir todos los elementos teóricos y metodológicos que serán utilizados en la presente Investigación. Partiendo de los diferentes enfoques de la Priorización de Requerimientos que se han citado en el Capítulo 2, se profundizarán aquellos que servirán como soporte teórico para la construcción del **“Modelo de Priorización de Requerimientos de Usuarios a partir de la segmentación de las Preferencias de los Stakeholders” (MPRUSPS)** que será desarrollado en el próximo Capítulo. Asimismo, se describirán los principales elementos metodológicos que serán utilizados al mismo efecto.

3.2 Método para el relevamiento de preferencias de implementación de los Requerimientos de Usuarios

Antes de describir la técnica utilizada para relevar las preferencias de implementación de requerimientos de usuarios, resulta necesario definir lo que se entiende en esta investigación por Priorización de Requerimientos. En efecto, se asumirá la definición aportada por Firesmith (2004), quien la define como: *“Proceso que determina el orden de implementación de los requerimientos de un sistema ó como aquel proceso que determina el orden de importancia de los requerimientos en función de la percepción de los stakeholders”*

Considerando esta definición, se podrá asumir que los *stakeholders* valorarán un requerimiento en particular en función de la **importancia** que le sugiere la implementación del mismo. Si bien varios autores, entre ellos: Berander (2004), Lehtola (et. al. 2004) y Ma (2009) han criticado este criterio por considerarlo un tanto abstracto y ambiguo, por el contrario, la presente investigación utilizará el criterio en cuestión. Se fundamenta tal decisión por dos motivos principales. En primer lugar, el método AHP que a continuación se describirá, se basa en este criterio en particular a los efectos de ponderar la preferencia de los *stakeholders*. Luego, la importancia revelada por los *stakeholders* será proporcionada por su desempeño socio-cognitivo. En consecuencia, el criterio de importancia será operado a través de aspectos numéricos y lingüísticos, con lo cual, se espera mitigar “el efecto ambiguo” que advierten los autores citados en relación al uso de este criterio en particular³.

En base al criterio de Priorización descripto, se procederá a describir el Método AHP como técnica aplicada para el relevamiento de las preferencias de los *stakeholders*. Se fundamenta tal decisión por tres motivos básicos. El primero de ellos, por la amplia utilización que ha tenido el Método AHP en el área de Priorización de Requerimientos de Software, sobre todo a partir de la reformulación propuesta por Karlsson (1996). Luego, y tal cual se manifiesta en Gass y Forman (2001) se estima que el método cuestión promueve una mejor comprensión del fenómeno a evaluar, gracias a la comparación de “a pares” que propone entre las alternativas disponibles. Por último, se pondera también la existencia de una métrica robusta (Índice de Inconsistencia) que permite apreciar el grado de coherencia experimentado en las decisiones de preferencia de un *stakeholder* en particular.

³ En la Sección 3.3 del presente profundizaremos el aspecto cognitivo-lingüístico.

3.2.1 Método del Proceso de la Jerarquía Analítica (AHP)

El enfoque AHP (*Analytic Hierarchy Process*) ha sido frecuentemente utilizado en la Ingeniería de Requerimientos en general y en la Priorización de Requerimientos en particular [Gaur, (et. al., 2010), Iqbal (2010), Karlsson (1996, 1997, 1998), Karlsson (2004,2006, 2007), Jung (1998), Motupally (2008), Perini (et. al., 2009), Perini y Sommarive (2007), Rodríguez Bello (2007), Sadiq (et. al, 2009), Sadiq (et. al, 2010), entre otros]. El instrumento en cuestión, se basa en tres componentes principales: una estructura, una métrica establecida en ratios, y una síntesis [Saul (et. al., 2001) y Rodríguez Bello (2007)]. Dicha herramienta, asiste en la toma de decisiones respecto a un problema en particular, el cual involucra una serie de alternativas que son ponderables en importancia a través del juicio valorativo que es asignado por un conjunto finito de participantes, quienes a su vez, adoptan múltiples criterios respecto del problema en cuestión. En este contexto en particular, el método ha demostrado su robustez y fiabilidad.

Sin embargo, y tratándose de decisiones humanas respecto a las preferencias entre diferentes alternativas, el problema de la consistencia, ha sido una cuestión inherente en este tipo de formulaciones. En el contexto de Priorización de Requerimientos de Software, la inconsistencia en el proceso de decisión puede deberse a una serie de cuestiones. Por ejemplo, tomar una decisión teniendo como límite el tiempo, disponer información incompleta [Kim y Ahn (1999)], tener diferente experticia en torno a una dimensión del problema [Li (1999)] son algunos factores que pueden influir en el proceso de decisión, al margen de los criterios que puedan existir entre los *stakeholders* a fin de abordarlo. En consecuencia, y tal cual lo manifiestan Wang y Chan (2007), disponer de un índice de consistencia convincente cuando el número de atributo o alternativas de un problema es incrementado, resulta muy dificultoso en el método AHP.

A los efectos de mitigar el problema de consistencia, se han desarrollado diferentes técnicas con el fin de interactuar con el método en cuestión. A modo de ejemplo, un estudio realizado por Li (et. al., 2009) propone un *framework* que integra elementos de la teoría de conjuntos aproximados junto a las propiedades operacionales de AHP. En efecto, basándose en los elementos metodológicos de la teoría: reducto, núcleo y región positiva, logran reducir un Sistema de Decisión, que luego servirá como insumo básico en la fase ulterior de procesamiento mediante AHP-Método de Escala. Uno de los beneficios que se desprende de este enfoque, consta de la optimización de factores desde una perspectiva tanto cualitativa como cuantitativa. Consecuentemente, la información a coleccionar es menor, el proceso de análisis insume menos tiempo, y

finalmente, se logra enfocar mejor a los usuarios respecto a los criterios claves que debe valorar ante un problema de decisión multi-criterio.

Asimismo, un estudio elaborado en Xie (et. al., 2008) determinan que los resultados provistos por el método AHP y por otros, como programación lineal o métodos propios de la Teoría de los Conjuntos Difusos, no se encontrarían exentos de los problemas de inconsistencia que pueden tenerse a lugar, producto de las circunstancias expuestas. En consecuencia, los autores proponen vincular el modelo VPRS de Ziarko (1993; 2002) [*Variable Precision Rough Set*] con el método AHP. El modelo VPRS permite un grado controlado de errores de clasificación. Cualquier regla de clasificación parcialmente incorrecta, ofrece una valiosa información sobre la tendencia, sí la mayoría de los datos disponibles a los cuales son aplicados tal regla, pueden ser correctamente clasificados. De esta manera, la vinculación propuesta pretende determinar el peso de los atributos de condición (integrado por factores) consignados por cada uno de los *stakeholders*, respecto a su juicio valorativo. El resultado preliminar del experimento, sugiere que dicha vinculación mitiga la incidencia de las circunstancias descritas (*ut supra*) y promueve el descubrimiento del o de los factores más importantes que promueven una determinada decisión.

Sin perjuicio de lo anterior, la presenta investigación utilizará el Método AHP para solucionar el problema que supone relevar las preferencias de implementación de una serie dada de requerimientos de usuarios. Sin embargo, se utilizará la reformulación propuesta por Karlsson (1996).

La reformulación propuesta por Karlsson (1996) pretende un esquema de valorización sin jerarquía de criterios, es decir, considera la valorización de un requerimiento por sobre otro en forma directa. Luego, se debe considerar que la cantidad de requerimientos de usuarios resultan irreductibles para la evaluación de las preferencias, con lo cual, no se podrá considerar la sugerencia realizada por Xie (et. al., 2008).

Por su parte, y tal cual se establecerá en el Enfoque Cognitivo (3.3), sí los criterios a considerar por parte de un *stakeholder* cuando éste emite un juicio de valorización, depende fundamentalmente de su Memoria Semántica, entonces, debería ser un método psico-cognitivo o conjunto de ellos, el/los responsable/s de elicitar dichos criterios. Con lo cual, todas aquellas técnicas aplicadas al AHP que no sean estrictamente psico-cognitivas no serán consideradas en este trabajo [como por ejemplo el *framework* elaborado por Li (et. al., 2009)].

En sí mismo, dicha reformulación supone el establecimiento de una serie de pasos previos que redundarán en la obtención del vector final de preferencia del *stakeholder*. Para ello, el primer paso consistirá en el establecimiento de una matriz de “n x n” requerimientos de usuarios [Ecuación (I)]:

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \quad (I)$$

En el segundo paso, se procederá a efectuar las comparaciones entre pares. Tal cual queda señalado en la Ecuación I, los términos de celda implican una comparación de a pares entre el requerimiento de orden i y su consecutivo (a_{ii+1}). Para llevar a cabo las comparaciones, se utiliza una escala de medida por ratios (Tabla 2). Su objeto básico, consta de medir la intensidad o importancia relativa que tiene un requerimiento con respecto a otro, de acuerdo a la percepción provista por un determinado *stakeholder* [Saaty, (1986); Brugha, (2004)].

<i>Intensidad de la Importancia</i>	<i>Definición</i>	<i>Explicación</i>
1	Importancia Igual.	Dos requerimientos contribuyen por igual al objetivo de satisfacción
3	Importancia Moderada.	La experiencia y el juicio favorecen levemente un requerimiento por sobre otro
5	Importancia Esencial o Fuerte.	La experiencia y el juicio favorecen fuertemente un requerimiento por sobre otro
7	Importancia Demostrada.	Un requerimiento se favorece fuertemente y su dominación es demostrada en la práctica.
9	Importancia Extrema.	La evidencia que favorece a un requerimiento por sobre otro está en el orden más alto posible de la afirmación
2,4,6,8	Importancia contenida en dos juicios adyacentes	Valores intermedio entre dos juicios adyacentes.

Tabla 2 "Escala utilizada para la valorización de Requerimientos de Software: Fuente: Karlsson (1996).

Asimismo, como es de suponer, todos los términos de la diagonal principal de la matriz (Ecuación I) serán 1, dado que se comparan pares de requerimientos iguales. Recabada la matriz de preferencia, el tercer paso consistirá en normalizar cada celda mediante la ponderación de la sumatoria de los términos que componen la columna de referencia. Es decir, dividiendo cada a_{ij} por la respectiva suma de la columna en donde la celda se halle. Finalmente, el cuarto paso, consistirá en estimar el vector de auto-valor que ponderado por el número de requerimientos a priorizar, proporcionará el vector de preferencia del *stakeholder* [Ecuación (II)]:

$$w = \frac{1}{n} \cdot \begin{pmatrix} w_{11} \\ \dots \\ w_{nn} \end{pmatrix} \quad (\text{II})$$

Con el fin de evaluar el grado de consistencia de las comparaciones efectuadas por el *stakeholder*, se procede a calcular en primera instancia el máximo auto-valor que adquiere la matriz de comparación, es decir, λ_{max} . De esta manera, se calculará el vector **p** [Ecuación (III)] mediante la multiplicación entre la matriz de comparación (*A*) y el vector de prioridad (*w*)

$$p = A \cdot w = \begin{pmatrix} p_{11} \\ \dots \\ p_{1n} \end{pmatrix} \quad (\text{III})$$

Una vez obtenido el vector **p**, se dividirá cada término de la columna por cada elemento del vector de prioridad. Se obtendrá en consecuencia el vector **t** [Ecuación (IV)]:

$$t = \begin{pmatrix} \frac{p_{11}}{w_{11}} \\ \dots \\ \frac{p_{1n}}{w_{nn}} \end{pmatrix} \quad (\text{IV})$$

Finalmente, el máximo auto-valor de la matriz de comparación (λ_{max}) surgirá como el promedio entre la sumatoria de los t_{1i} y el número total de requerimientos priorizados [Ecuación (V)]:

$$\lambda_{max} = \frac{\sum_{i=1}^n t_{1i}}{n} \quad (\text{V})$$

El Índice de Consistencia [Ecuación (VI)], relaciona el máximo auto-valor (*eigenvalue*) de la matriz de comparación (λ_{max}), junto con la cantidad de opciones evaluables. Sí el máximo auto-valor de la matriz de comparación, es igual a la cantidad de opciones evaluables, entonces, la consistencia será perfecta [Saaty, 1990]:

$$\text{Índice de Consistencia (CI)} = \frac{(\lambda_{max} - n)}{n - 1} \quad (VI)$$

Por su parte, el Índice Aleatorio [Ecuación (VII)] adquiere un valor estocástico que se encuentra especificado en función de la cantidad de opciones existentes que serán analizadas. En Saaty (1980), existe una tabla que tabula los valores de RI para un conjunto de 25 opciones.

$$\text{Índice Aleatorio (RI)} = f(n) \quad (VII)$$

Luego, el cociente entre (CI) y (RI) establece el Ratio de Inconsistencia [Ecuación (VIII)] obtenido luego del proceso de decisión. Según Saaty (1980, 1990), *cuando este índice sea menor o igual que 0,1, la inconsistencia exhibida será aceptable*. Luego, si este valor es significativamente mayor que el valor crítico establecido, entonces, sería prudente revisar los juicios vertidos.

$$\text{Ratio de Inconsistencia (CR)} = \frac{CI}{RI} \quad (VIII)$$

3.3 Elementos de la Psicología Cognitiva aplicados en la segmentación de preferencias de implementación de Requerimientos de Usuarios.

Se definirá a la Psicología Cognitiva como aquella rama especializada de la Psicología, cuyo objeto básico, consta del análisis de los procesos mentales implicados en el conocimiento. En consecuencia, su alcance como disciplina científica, comprende el estudio sistemático de la percepción, memoria y el aprendizaje hasta la formación de conceptos y el razonamiento lógico experimentado por los sujetos. En síntesis, se podrá considerar que la Psicología Cognitiva provee de un marco conceptual que ayudará a comprender las dificultades que las personas pueden experimentar cuando describen sus inquietudes [Posner, (1993)].

De acuerdo a las necesidades de la presente investigación, se estima como conveniente describir y analizar el concepto de memoria semántica. En efecto, y según Vivas (et. al., 2009) se podrá definir a la memoria semántica como aquella que permite acceder a los recuerdos de los significados de los conceptos, a la comprensión de esos recuerdos y a disponer de todo otro conocimiento basado en ideas sin tener necesidad de recuperar las experiencias específicas que las han originado. Asimismo puede ser pensada como la confluencia de un diccionario, una enciclopedia y un tesoro, todo ello en uno [Smith (1976) y Tulving (1972)].

Como consecuencia de lo anterior, el significado de un concepto parece no emerger directamente sólo de las propiedades intrínsecas del mismo, no se trata de un atributo que se debe descubrir como si fuese una propiedad inherente al objeto en cuestión, ni la labor del sujeto consiste en develar dicho significado. En lugar de ello, el significado atribuido a un ente, fenómeno o proceso parece emerger del peso relativo de todas y cada una de las relaciones fácticas, intelectuales o emocionales que el sujeto haya logrado establecer en su historia con el objeto de referencia [Murphy (2002); Rogers y McClelland (2004); McNamara (2005); Vivas (2009a)]. Confirmando esto último, en Peraita Adrados y Grasso (2010) se establece que en la representación conceptual que los sujetos poseen sobre las entidades del mundo que los rodea, existirán rasgos que tienen más relevancia o peso que otros, y que éstos a su vez, se manifestarán a partir del proceso de identificación de tales entidades (efecto distintivo).

Así, la comprensión de cualquier signo tiene un componente tanto interno como externo al sujeto. Es interno en tanto los procesos de percepción, abstracción, codificación, almacenamiento, evocación e interpretación son individuales y se estructuran en función de los otros signos existentes en la memoria semántica. Las

diferentes relaciones semánticas dependen del estado de la configuración disponible en el sujeto. Es externo en tanto que un signo no puede ser separado de su situación social sin perder su naturaleza semiótica. En un sentido amplio, se podrá indicar que el significado de un signo no es un hecho sino un proceso. En consecuencia, un signo siempre está orientado hacia alguien y, por tanto, es un elemento constitutivo de un proceso de negociación que le pertenece por igual al que lo emplea y quien va dirigido [Vivas (2009a); Vivas (2009b); Vivas y Ledesma (2009); Vivas (et. al., 2009)].

De esta modo, sí el significado de un símbolo no es una propiedad intrínseca, sino que emerge como resultado de un acto interpretativo, se podrá presumir que el significado puede no ser estático ni permanente, ya que las personas y las comunidades lo modifican según las circunstancias, necesidades, intereses, conveniencias, tendencias e ideologías. Es por ello que cada relación comunicativa entre dos hablantes implica siempre un acto de interpretación y negociación de significados [Vivas (et. al., 2009)].

Como consecuencia de ello, se podrá señalar que la emisión de una preferencia respecto de una entidad dada, dependerá fundamentalmente de la configuración que haya adquirido la red semántica del sujeto en torno a dicha entidad [Figuerola, Solís y González (1974)]. Por tal motivo, y para el propósito de ésta tesis, se podrá suponer que:

“Sí la emisión de un Juicio de Preferencia inherente a una Estructura Conceptual, depende de la configuración que haya adquirido la red semántica del sujeto en torno a dicha estructura, entonces, ante la existencia de un grupo de sujetos que emitan una serie de juicios de preferencias que sean similares entre sí, se podrá indicar que la configuración de las respectivas redes semánticas de los integrantes, se tornan equivalentes para la mencionada Estructura Conceptual”.

Luego, el supuesto anterior debe contener las siguientes restricciones:

- Los *stakeholders* pertenecen a la misma comunidad lingüística.
- No se denuncia comunidad léxica. Es decir, no existen enlaces basados en propiedades fonológicas.
- Las indagaciones son transversales en un momento dado.

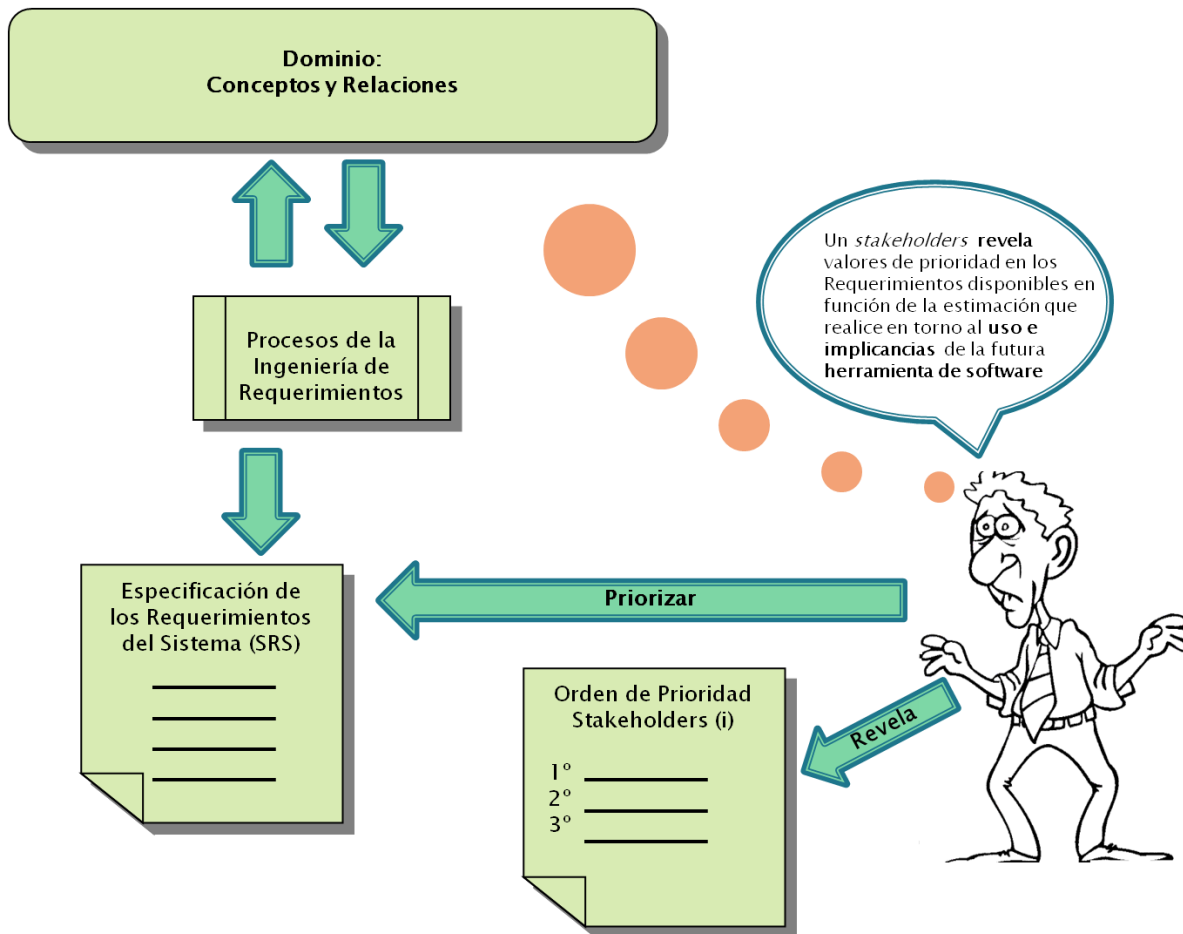


Figura 3.1: "Implicancias de la emisión de un juicio de Preferencias". Fuente: Elaboración Propia

Como se puede apreciar en la Figura 3.1, podrá considerarse que el Dominio de una aplicación se compone de una serie de conceptos y relaciones (**Estructura Conceptual**) que los diversos procesos de la Ingeniería de Requerimientos, tratarán de sintetizar a los efectos de constituir una Especificación de Requerimientos (SRS). Por su parte, cuando un *stakeholder* emite un juicio de preferencia con el objeto de ordenar en términos de importancia los requerimientos de dicha especificación, lo hará en función del grado de conocimiento que dicho sujeto detente respecto a los usos e implicancias que traerá aparejado la futura herramienta de software. En este sentido, el juicio de preferencia resulta funcional a la configuración que haya adquirido la red semántica del sujeto, en torno a la Estructura Conceptual que caracteriza al Dominio.

Bajo este razonamiento, ante la existencia de sujetos que emitan preferencias similares, se podrá sugerir que la configuración que hayan adquirido sus respectivas redes semánticas, se tornarán equivalentes para la mencionada Estructura Conceptual.

Finalmente, y a los efectos de operar este razonamiento, se utilizará en forma conjunta, los métodos: “Buscador de Definidoras” de Vivas (et. al., 2009), “Valor Total de las Valorizaciones” y “Razón de Exclusión” de Vivas, **Azzolini** y Vivas (2010). Tales métodos, han sido desarrollados por el Centro de Investigación en Procesos Básicos, Metodología y Educación (CIMEPB) de la Universidad Nacional de Mar del Plata.

3.3.1 “Buscador de Definidoras” (DF: Definition Finder)

Un concepto puede ser expresado o asociado por un conjunto de palabras. Asimismo, dichas palabras dependen fundamentalmente de una construcción social que las erige y las determina [Huapaya, Lizarralde y Arona (2009)]. Partiendo de este supuesto básico, el Grupo de Inteligencia Artificial (GIA) de la Facultad de Ingeniería (UNMdP), ha desarrollado una herramienta de Software denominada *Definition Finder* (Buscador de Definidoras) que es utilizado por el Centro de Metodología Básica (CIMEPB) a los efectos de establecer, como un concepto se determina a partir de un conjunto de Identificadoras provistas por un Colectivo Social.

El instrumento citado, ayuda a explorar las redes semánticas que se generan a partir de las ideas que un colectivo social, evoca cuando se presenta un concepto o juicio. En sí mismo, implica un procedimiento recursivo que reordena en una única tabla, el peso de la conexión de cada término evocado con referencia al concepto objetivo, en función del orden relativo con el que lo menciona o no un determinado *stakeholder* del grupo social.

Luego, el método en cuestión, asigna un peso relativo a cada una de las Definidoras elicidadas en un Colectivo Social dado, en función de la frecuencia de aparición del atributo y por el orden relativo asignado por cada uno de los sujetos que componen el mencionado colectivo. De esta manera, se obtiene para un concepto dado, un conjunto finito de definidoras que serán características para un determinado colectivo social [Ecuación (IX)]:

$$PRD_i = d_{Di,n} f(Fr_{Di,n}, O_{Din_j}) \quad (IX)$$

En donde:

PRD_i = Peso relativo de la Definidora de orden i asignado por el Colectivo Social.

d = Variable binaria. Adquiere valor "1" cuando la definidora de orden i se encuentra identificada por al menos dos sujetos del colectivo social "n". Luego, el valor "0" acontece cuando no se cumple la condición anterior.

Fr = Frecuencia relativa de aparición de la Definidora de orden i en el colectivo social n .

O = Orden de aparición que el sujeto de orden j del colectivo social n , le asigna a la Definidora de orden i .

3.3.2 Método: Valor Total de las Valorizaciones (VTV)

La VTV o Valor Total de las Valorizaciones, es un método creado por Vivas, **Azzolini** y Vivas (2010) el cual se puede definir como un estimador del campo semántico de un concepto particular. El presente Método, parte de la concepción que el significado atribuido a un ente, fenómeno o proceso, emerge del peso relativo de todas y cada una de las relaciones fácticas, intelectuales o emocionales que el sujeto haya logrado establecer en su historia con el objeto de referencia.

Por su parte, y desde una perspectiva social, la definición de un concepto emerge a partir de la identificación de un conjunto de definidoras que han sido propuestas por un determinado colectivo de sujetos. Luego, será factible ponderar el peso relativo que el conjunto de individuos indagados, le ha asignado a cada una de las definidoras que han identificado.

Para ello, se utiliza la sucesión de pesos relativos del *Definition Finder* para un concepto dado. En efecto, los resultados empíricos obtenidos, indican que las definidoras de un concepto dado, exhiben un comportamiento regular que es descriptible mediante la implementación de un Modelo Exponencial de ajuste.

Dicho hallazgo, permite distinguir el campo semántico de una lista de atributos, y al mismo tiempo, caracterizar el comportamiento de cada atributo en un concepto dado. De esta manera el método se propone estimar, por un lado, la relevancia semántica de un conjunto de definidoras, y por el otro, el efecto distintivo que se sucede en la secuencia de las mismas.

La relevancia semántica de un conjunto de definidoras, se define por el grado de contribución que un subconjunto de ellas, aporta al significado “base” de un determinado concepto [Sartori y Lombardi (2004); Sartori, (et., al., 2006)]. Luego, el efecto “relevancia semántica” queda aproximado por el valor que adquiere el estimador del parámetro β_0 del modelo VTV. Por su parte, el efecto distintivo en una serie de definidoras, hace referencia a una medida que va desde lo muy compartido hasta lo no- compartido en dicha sucesión [Devlin, (et. al., 1998); Garrard (et. al., 2001)]. Luego, el efecto “distintivo” queda recogido por el valor que adquiere el estimador del parámetro β_1 en el modelo VTV.

Por su parte, existe una relación inversa entre el VTV y el valor de β_1 . Es decir, cuanto mayor sea la proporción a la cual se degrada el peso asignado al conjunto de las definidoras ordenadas y sucesivas de un concepto en particular, tanto menor será la VTV del concepto dado. Por su parte, existirá una relación directa entre la VTV y β_0 . Es decir, cuanto mayor sea el peso parcial asignado por un Colectivo Social a las primeras definidoras de un concepto dado, tanto mayor será la VTV de éste último (dada la relevancia semántica aportada por las primeras definidoras del concepto). Finalmente, el modelo del Valor Total de las Valorizaciones [Ecuación (X)]:

$$VTV = \frac{\beta_0}{\beta_1} [(e^{\beta_1 n}) - (e^{\beta_1})] \quad (X)$$

En donde:

VTV = Se refiere al Valor Total de las Valorizaciones del Concepto analizado.

β_0 = Se refiere al peso que adquiere la “relevancia semántica” del concepto analizado.

β_1 = Se refiere al efecto distintivo que adquiere el concepto analizado.

n = se refiere a la cantidad de definidoras socialmente identificadas en el concepto analizado.

A nivel ilustrativo, si un Colectivo Social es indagado respecto a dos Conceptos A y B y resulta que la VTV del Concepto A es mayor que la correspondiente al Concepto B, se podrá indicar que el Concepto A, tiene un Campo Semántico mayor que el Concepto B. En este sentido, el Colectivo Social ha podido consensuar una mayor cantidad de Definidoras (“Efecto Definidoras Socialmente Identificadas” estimado por “n”) y/o ha establecido una mayor relevancia semántica (Efecto Relevancia Semántica) y/o se ha propiciado un efecto distintivo relativamente más bajo del Concepto A comparado con el Concepto B. En

síntesis, el Concepto A se encuentra mejor identificado que el Concepto B por parte del Colectivo social.

Relaciones:

- | | | | | | |
|----|---|-----------|---|---|-----|
| 1) | ↑ | β_0 | → | ↑ | VTV |
| 2) | ↑ | β_1 | → | ↓ | VTV |
| 3) | ↑ | n | → | ↑ | VTV |

En donde:

- 1) “Efecto Relevancia Semántica”
- 2) “Efecto Distintivo”
- 3) “Efecto Definidoras Socialmente Identificadas”

Por su parte, la variable VTV queda definida en el intervalo:

$$VTV \in [0; \infty+)$$

Sin embargo, diferentes experimentos han determinado un intervalo empírico como el siguiente [Vivas, **Azzolini** y Vivas (2010)]:

$$VTV \in [0; 6]$$

3.3.3 Método: Razón de Exclusión (RE)

Luego del análisis propuesto por los métodos: “**Definition Finder**” y **VTV**, resulta factible establecer una relación entre la cantidad de Definidoras que han sido elicitadas en un colectivo social, y aquellas que han sido aportadas en forma individual por los integrantes de dicho colectivo. Luego, la relación en cuestión se define como “Razón de Exclusión”[Ecuación (XI)]:

$$RE = \frac{\text{Cantidad de Definidoras Individualmente Identificadas}}{\text{Cantidad de Definidoras Socialmente Identificadas}} \quad (XI)$$

Se recuerda que el término “n” en la *Ecuación X* recoge el efecto que aporta la cantidad de definidoras socialmente establecidas e identificadas por un colectivo social. En este sentido, cuanto mayor sea dicho término, mayor será el campo semántico del concepto definido. Por su parte, la existencia de definidoras particulares que no sean compartidas por el Colectivo Social (peso relativo = 0), se consideran términos excluyentes que no promueven el incremento del campo semántico del concepto analizado. En consecuencia, y en términos de correlación, existirá una asociación inversa entre RE y VTV. Finalmente, la razón de exclusión, indicará la proporción de definidoras identificadas en forma individual por un determinado colectivo social.

Se operará este concepto, utilizando la sucesión de pesos relativos provistos por el *Definition Finder* para un concepto en particular. En efecto, dicha sucesión se compone de pesos iguales o mayores a 0. Con lo cual, la razón entre el número total de pesos neutros (0) y el número de pesos positivos (mayores a 0), permitirá calcular la Razón de Exclusión del concepto indagado. Por su parte, la variable RE queda definida en el intervalo:

$$RE \in [0; \infty+)$$

Sin embargo, diferentes experimentos han determinado un intervalo empírico como el siguiente [Vivas, **Azzolini** y Vivas (2010)]:

$$RE \in [0; 1,4]$$

3.4 Técnicas para el Agrupamiento de preferencias de implementación de Requerimientos de Usuarios.

Una vez aplicado el Método AHP (3.2), se dispondrá para cada uno de los *stakeholders*, de un vector que indique como se distribuye su preferencia en cada uno de los requerimientos de usuarios disponibles. Por aplicación del supuesto establecido en el Enfoque Cognitivo (3.3), se establece que ante la existencia de vectores similares en términos de la distribución de preferencias, se encontrarán sujetos cuyas redes semánticas se configuran de forma análoga a partir de la estructura conceptual dada. Bajo estas consideraciones, se necesitará implementar alguna técnica que permita aglomerar aquellas preferencias consideradas como similares. Luego, el objetivo consiste en detectar grupos de *stakeholders* cuyos niveles de preferencias sean internamente similares (preferencias intra-grupo) y externamente disímiles (preferencias inter-grupos).

En efecto, las técnicas de agrupamiento (*clustering*) tienen como objetivo la clasificación de objetos de acuerdo con sus similitudes, así como la organización de datos en grupo. En concreto, la propiedad que interesa es su facilidad para dividir un conjunto de datos en varios grupos, de forma que la similitud entre los elementos dentro de un grupo sea mayor comparada con otros grupos [Díez Ruano (2003)].

3.4.1 Técnicas de Agrupamientos (*Clustering*) más comunes

Agrupar las preferencias de los *stakeholders*, en función del grado de similitud exhibido en la asignación de importancia entre los diferentes requerimientos de usuarios disponibles, implica necesariamente escoger alguna técnica de agrupamiento. Según Álvarez García (2010), se podrá clasificar dichas técnicas en tres enfoques básicos:

- Clasificación o Reparto de clústeres: En la que se encontrarán aquellas técnicas que utilizan la metodológica Jerárquica o Particional para la configuración de clústeres. Por ejemplo, Dendrogramas, Algoritmo K-means, entre otras.
- Según la geométrica subyacente: En la que se encontrarán aquellas técnicas que permiten una representación visual de los grupos. Es decir que parten de la idea de una clasificación natural de los datos en donde resulta factible trasladarlos como puntos en el espacio (dos o tres dimensiones) de forma tal que se mantengan las

propiedades de clasificación. Son ejemplo de este enfoque: Mapa Auto-organizados (SOM), Escalonamiento Multidimensional (MDS), entre otros.

- Probabilísticos: Son aquellos que partes de la clasificación de los datos utilizando técnicas probabilísticas. En efecto, se busca modelar la colección de datos generando procesos aleatorios y asignando distribuciones específicas. Ejemplo de ellos son: Binario Multivariante, Multinomial (o Naive Bayes), Maximización de la Expectativa (EM), Mezcla de Múltiples Causas (MCMM), por citar algunos.

Respondiendo a esta clasificación, pero sin hacer una revisión exhaustiva, se comentarán brevemente las técnicas más comunes citadas en: Duda y Hart (1973); Maravall (1993); Díez Ruano (2003) y Duda (et. al., 2000). Por ejemplo:

3.4.1.1 Dendrogramas: Se refiere a una técnica de agrupamiento de tipo Jerárquica que organiza los datos en sub-categorías, que se van dividiendo en otras hasta llegar al nivel de detalle deseado. Este tipo de representación, permite apreciar claramente las relaciones de agrupación entre los datos e incluso entre grupos de ellos [Rousseeuw y Kaufman, (1990)]. Los pasos del método son:

1. Calcular las distancias entre todos los pares de objetos. Esto es lo mismo que asumir que cada objeto constituye un grupo: $\{C_1, \dots, C_N\}$.
2. Luego, se buscan los dos grupos más cercanos (C_i, C_j), éstos se juntan y constituyen uno solo C_{ij} .
3. Se repite el paso 2 hasta que no quedan pares de comparación.

La cuestión crítica de este método es la forma de “juntar” los grupos entre sí. En efecto, se utilizan básicamente tres enlaces básicos: simple, promediado y completo.

3.4.1.2 Algoritmo de K- medias (K-means): Se refiere a una técnica de *clustering* de tipo Participación. El nombre de este algoritmo establece que existen K clases diferentes, por lo que es necesario conocer a priori el número de clases existentes. El algoritmo ha sido ampliamente utilizado por su sencillez, eficiencia y robustez, siempre que se conozca el número de clases con exactitud. Los pasos del algoritmo, dado un conjunto Z de N elementos a clasificar, son:

1. Se escogen K elementos al azar que forman las clases iniciales, siendo ellos, como únicos elementos, los centroides (c_i) de las mismas.
2. Se asignan los restantes elementos a la clase A_i cuyo centro esté más cerca.
3. Se recalcula el centro de cada clase como la media de los elementos que componen la clase (z_j), buscando minimizar el índice de distancia intra-clase (J) [Ecuación (XII)]

$$J = \sum_{i=1}^K (\sum_{j, z_j \in A_i} \| z_j - c_i \|^2) \quad \text{XII}$$

4. Se vuelve al segundo paso hasta que la distribución sea estable (J no varíe) según una condición de finalización predeterminada.

Este algoritmo es extremadamente sensible al parámetro K, ya que un valor superior al número real de clases dará lugar a clases ficticias y un valor de K inferior producirá menos clases de las reales. Se podría detectar una mala elección del parámetro K analizando las dispersiones estadísticas de las clases formadas, aunque en realidad, esa dispersión también podría deberse a la naturaleza de las variables estudiadas.

3.4.3 Método de la Maximización de Expectativa (EM): Tal enfoque probabilístico es el más apropiado para la detección de patrones con un carácter estocástico. La idea básica de este tipo de modelo es que el vector de entrada o instancias x_1, x_2, \dots, x_m son observaciones provenientes de un conjunto K de distribuciones desconocidas: E_1, E_2, E_k . Se supone que la densidad de una observación x_k con respecto a E_i viene dada por $f_i(x_k, \Phi)$ para un conjunto desconocido de parámetros Φ . La probabilidad que x_k se distribuya como E_i se denota por τ_k^i .

Dado que por lo general se asume que cada punto pertenece a una sola distribución, se obtiene:

$$\sum_{i=1}^k \tau_k^i = 1$$

Finalmente, el objetivo del método consiste en estimar el valor de los parámetros Φ y τ que maximice (o minimice) la función de probabilidad $L(\Phi, \tau)$ [Ecuación (XIII)]

$$L(\theta, \tau) = \sum_{r=1}^n \ln\left(\sum_{i=1}^K \tau_k^i f_i(x_k|\theta)\right)$$

XIII

De esta manera, el método de la maximización de expectativa, permite dicha optimización a través de un proceso iterativo eficiente. A partir de allí, se establece los agrupamientos con sus respectivos datos [Fung (2001)].

3.4.2 Mapas Auto-Organizados de Kohonen (SOM)

Los modelos de Mapas Auto-organizativos (SOM) fueron introducidas por Kohonen (1982; 1990; 2001) y son un tipo especial de las redes neuronales artificiales de aprendizaje no supervisado y competitivo. El desarrollo de esta arquitectura está motivada por una característica notable del cerebro: muchas zonas especializadas se encuentran organizadas de tal forma que diferentes estímulos sensoriales están representados por mapas topográficamente ordenados [Haykin (1994)]. En particular, estímulos sensoriales como el táctil, visual y auditivo están mapeados en la corteza. En Kohonen (1990) se resume los dos elementos básicos de una red para lograr un ordenamiento espacial: concentración espacial de la actividad de la red neuronal en una o en un reducido grupo de unidades e incremento en la sensibilidad o selectividad de la neurona activada y su vecindad topológica al estímulo de entrada.

Arquitectura del SOM

Un modelo SOM está compuesto por dos capas de neuronas. La capa de entrada se encarga de recibir y transmitir a la capa de salida, la información procedente del exterior. La capa de salida, es la encargada de procesar la información y formar el mapa de rasgos. Normalmente, las neuronas de la capa de salida se organizan en forma de mapa bidimensional, como se aprecia en la siguiente Figura 3.2:

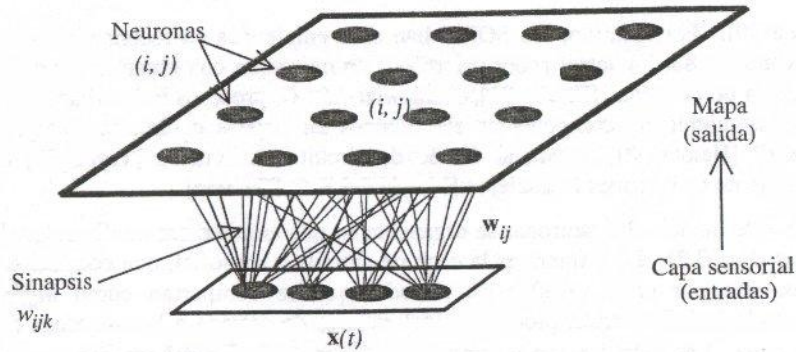


Figura 3.2: "Mapa Bidimensional en SOM". Fuente: Burgos (2003).

Las conexiones entre las dos capas que forman la red son siempre hacia delante, es decir, la información se propaga desde la capa de entrada hacia la capa de salida. Cada neurona de entrada i está conectada con cada una de las neuronas de salida j mediante un peso w_{ji} . De esta forma, las neuronas de salida tienen asociado un vector de pesos w_j llamado vector de referencia (o *codebook*), debido a que constituye el vector prototipo (o promedio) de la categoría representada por la neurona de salida j . Así, el SOM define una proyección desde un espacio de datos en alta dimensión a un mapa bidimensional de neuronas. Entre las neuronas de la capa de salida, puede decirse que existen conexiones laterales de excitación e inhibición implícitas, pues aunque no estén conectadas, cada una de estas neuronas va a tener cierta influencia sobre sus vecinas. Esto se consigue a través de un proceso de competición entre las neuronas y de la aplicación de una función denominada de vecindad, que produce la topología o estructura del mapa. Luego, las topologías más frecuentes son la rectangular y la hexagonal.

Por su parte, las neuronas adyacentes pertenecen a una vecindad N_j de la neurona j . La topología y el número de neuronas permanecen fijos desde el principio. El número de neuronas determina la suavidad de la proyección, lo cual influye en el ajuste y capacidad de generalización del SOM. Durante la fase de entrenamiento, el SOM forma una red elástica que se pliega dentro de la nube de datos originales. El algoritmo controla la red de modo que tiende a aproximar la densidad de los datos. Los vectores de referencia del *codebook* se acercan a las áreas donde la densidad de datos es alta. Eventualmente, unos pocos vectores del *codebook*, se encuentran en áreas donde existe baja densidad de datos.

Algoritmo del SOM

Paso 1: Entrenamiento: Presentado un vector x del conjunto de datos, se calcula su distancia (medida de similitud) a los vectores del *codebook*, usando por ejemplo la distancia euclídea: [Ecuación (XIV)]:

$$\|x - m_c\| = \min_j \{\|x - m_j\|\} \quad \text{XIV}$$

En donde:

x : es el patrón presentado.

m_c : es el vector de los *codebook* identificado por la BMU.

m_j : se refiere al conjunto de todos los *codebook*.

Paso 2. Una vez que se ha encontrado el vector más próximo o BMU (*best matching unit*) el resto de vectores del *codebook* es actualizado. El BMU y sus vecinos (en sentido topológico) se mueven cerca del vector x en el espacio de datos. La magnitud de dicha atracción está regida por la tasa de aprendizaje. Mientras se va produciendo el proceso de actualización y nuevos vectores se asignan al mapa, la tasa de aprendizaje tiende gradualmente a cero.

Sea $X_t \in R^p$, $i = 1..n$ un conjunto de datos de dimensión p y $V_j \in R^p$, $j = 1..c$, un conjunto de vectores de diccionarios, la regla de actualización de Kohonen será la siguiente: [Ecuación (XV)]:

$$V_{j,t} = V_{j,t-1} + \alpha_t h_{r,t} (X_t - V_{j,t-1}) \quad \text{XV}$$

Donde α_t representa el factor de aprendizaje, el cual es definido como una función decreciente que controla la magnitud de los cambios en cada interacción t y $h_{r,t}$ es una función que controla el tamaño de la vecindad de los nodos a ser actualizados durante el entrenamiento. Luego, ambos parámetros α_t y $h_{r,t}$ decrecen monótonamente durante el entrenamiento con el objetivo de lograr la convergencia [Pascual Montano (2002)].

Los pasos 1 y 2 se van repitiendo hasta que el entrenamiento termina. El número de pasos de entrenamiento se debe fijar antes a priori, para calcular la tasa de convergencia de la función de vecindad y de la tasa de aprendizaje. Una vez terminado el entrenamiento, el mapa se ordena en sentido topológico: “n” vectores topológicamente próximos se aplican en “n” neuronas adyacentes o incluso en la misma neurona.

Visualización del SOM

La matriz unificada de distancias, o Matriz U, es el método más común para mostrar la configuración topológica que adquiere un mapa auto-organizado. Asimismo, cada elemento del mapa representa una neurona en particular.

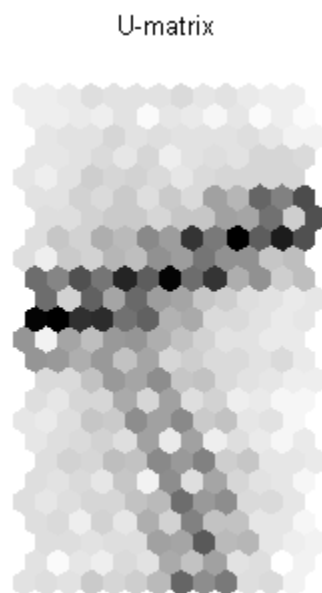


Figura 3.3: “Ejemplo de Matriz U”. Fuente: Elaboración Propia en base a Salida de Matlab.

Cuando se genera la Matriz U, se calcula a su vez una matriz de distancias entre los vectores de referencia de neuronas adyacentes en el mapa bidimensional. Después se selecciona algún tipo de representación gráfica, como por ejemplo, una escala de colores. Como se aprecia en la Figura 3.3, el mapa queda representado mediante una escala de grises. Las regiones o grupos quedan determinados por tonos aproximadamente iguales de grises y quedarán circunscriptos o delimitados por tonos más oscuros. En la misma Figura, puede observarse claramente la existencia de 3 regiones. Dichos grupos quedarían localizados en las secciones: superior, centro -inferior a la izquierda y centro- inferior a la derecha.

*Por motivos de la presente investigación, se utilizará este último método como técnica para agrupar las preferencias de los stakeholders. Se fundamenta tal elección por dos motivos fundamentales. En primer lugar, resulta necesario un método que permita, **sin información previa**, determinar a través de un conjunto reducido de datos de entrenamiento, una tipología (grupo) de stakeholders, en función del patrón de preferencia exhibido por ellos. Con lo cual, todos aquellos métodos como el K-means que necesitan de la definición a priori del número de grupos, quedarán descartados. Asimismo, se cree conveniente contar con la posibilidad de disponer de un recurso visual como la geometría subyacente de la clasificación resultante, a los efectos de corroborar como ha sido el proceso de imputación de los stakeholders en cada uno de los grupos o tipos hallados. Con lo cual, se descartarán todos aquellos métodos que no dispongan de tal característica.*

3.4.3 Índice Davies- Bouldin

Una vez procedida la clasificación (o segmentación) de las preferencias de los *stakeholders*, se deberá disponer de una métrica que permita determinar la bondad de las clasificaciones efectuadas. Para ello, se utilizará el método ofrecido por Davies-Bouldin (DB) por considerarlo como uno de los más comunes utilizados al efecto [Díez Ruano (2003)].

El Índice de Davies – Bouldin (DB) fue creado por Davies y Bouldin (1979) y constituye un método que permite medir la relación entre la dispersión de los datos y el centro de la clase o centro de clúster de las diferentes clases. Según Pinzón Morales (et. al., 2009) el presente es utilizado generalmente como algoritmo de penalización de separabilidad entre las clases. Por su parte, cuando en la muestra de datos existen valores atípicos y además los grupos se superponen, el índice DB, se postula como el más adecuado en términos de robustez y del costo computacional de procesamiento [Vesanto y Alhoniemi (2000)]. Este índice, requiere del cálculo de la similitud entre grupos: [Ecuación (XVI)]:

$$R_{i,j} = \frac{D_{i,i} + D_{j,j}}{D_{i,j}} \quad \text{XVI}$$

Donde: $D_{i,i}$ y $D_{j,j}$ son las dispersiones de los i -ésimo y j -ésimo clúster dadas respectivamente por : [Ecuación (XVII)]:

$$D_{i,i} = \frac{1}{N_i} [\sum_k \| x(k) - m \|^2]^{\frac{1}{2}} \quad \text{XVII}$$

Donde “ m ” representa la media del respectivo grupo y N_i es el número de datos. Luego, D_{ij} representará la distancia entre los centros estadísticos de los clústeres: [Ecuación (XVIII)]:

$$D_{i,j} = \| m_i - m_j \| \quad \text{XVIII}$$

Luego, el índice de DB es determinado: [Ecuación (XIX)]:

$$DB = \frac{1}{C} \sum_{k=1}^C \max_{i \neq j} (R_{i,j}) \quad \text{XIX}$$

Donde “ C ” representa el número de grupos. Asimismo, pequeños valores de DB indicarán que los grupos hallados se encuentran compactados y que los centros de cada grupo, se encontrarán alejados entre sí. Luego, la configuración de clústeres que minimiza DB, indicará el valor óptimo de clústeres [Bolshakova y Azuaje (2002)]. Finalmente, el presente índice se utilizará a los efectos de validar el número de clústeres establecidos a partir del procesamiento de las preferencias propuesto por la red SOM.

3.5 Proceso de Negociación aplicado a Grupos de *stakeholders*

Como se sabe, el producto básico que se obtiene a partir de la implementación de un proceso de priorización de requerimientos, consta precisamente en determinar una lista ordenada de requerimientos que se deberían implementar a lo largo del ciclo de vida del software. En este trabajo, se denomina a este producto como el *Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU)*. A partir de dicho esquema, resulta necesario constatar si los *stakeholders* se encuentran conformes o no con el mismo. En caso de registrarse conflictos (no conformidad) sería necesario implementar un proceso de negociación a los efectos de conciliar las diferencias halladas.

Asimismo, uno de los factores críticos en el éxito de un Proyecto de Software, consta precisamente en lograr una efectiva negociación de los requerimientos implementables, en una etapa relativamente temprana del ciclo de vida del software [In y Roy (2001); Boehm (et. al., 1995); Boehm y In (1996) In (et. al. 2001)]. Sucede que, al establecer una negociación colectiva, se intenta establecer una visión compartida respecto del Proyecto en cuestión y de los conceptos que éste gestiona.

A fin de establecer dicho proceso, se considerará en primer lugar la idea propuesta por Fricker y Grünbacher (2008), la cual se basa en aplicar tácticas de negociación focalizadas en constelaciones específicas. *En la presente investigación, las configuraciones no serán determinadas por los roles que adquieren los stakeholders en el torno organizativo, sino más bien, por su afinidad en el patrón de preferencia de requerimientos de usuarios implementables (grupos de stakeholders).*

El tal sentido, se destaca el pensamiento de Damian (et. al. 2008) el cual sugiere que el proceso de negociación de requerimientos, será más efectivo a medida que los *stakeholders* compartan un entendimiento común del dominio, y además, promuevan una comunicación cuyo contenido sea semánticamente válido y aceptado por los diferentes sujetos involucrados.

Respecto del funcionamiento del proceso de negociación, se partirá de la idea de Lehtola y Kauppinen (2004), en cuanto se pretende establecer un acuerdo continuo y enriquecido entre los *stakeholders*, a través de la ejecución de una serie de ciclos de cesión, abandono y renegociación de posiciones entre los diferentes involucrados que serán pasibles de un proceso de negociación. A tal efecto se aplica el modelo ABNP.

3.5.1 El Modelo ABNP

El Modelo ABNP (*Argument Based Negotiation Process*), se refiere a un proceso de negociación que se establece a partir de la interacción de los argumentos brindados por una serie *stakeholders* en torno a una estructura conceptual dada. Diversos autores, entre ellos: Chang (et al., 1995); Jennings (1998); Sillince (et al., 1999); Amgoud (et al., 2000); Avery et al., 2001; Kraus, 2001 y Rong et al., 2002), parten del esquema de argumentos de Toulmin (1958) a los efectos de guiar el diseño, generación e intercambio de afirmaciones que permitan justificar una determinada posición objetivo.

Cuando existen percepciones diferentes respecto de la interpretación de una determinada estructura conceptual, será factible implementar un proceso de negociación basado en argumentos (Modelo ABNP). En primera instancia, cada *stakeholder* definirá una posición que será relativa a su interpretación respecto de una estructura conceptual dada. Asimismo, se apoyará en un soporte de datos que considere oportuno. Luego, se inaugurará un ciclo iterativo en el cual, cada *stakeholder*, compartirá sus argumentos con otros, dando a lugar un proceso de cese, re-afirmación y refutación de las respectivas posturas iniciales [Lu y Jing (2009); Sierra (et. al., 1998)]. El ciclo iterativo se dará por finalizado, una vez que se haya logrado un consenso grupal respecto de la interpretación de una determinada estructura conceptual dada.

Proceso de Argumentación en el Modelo ABNP

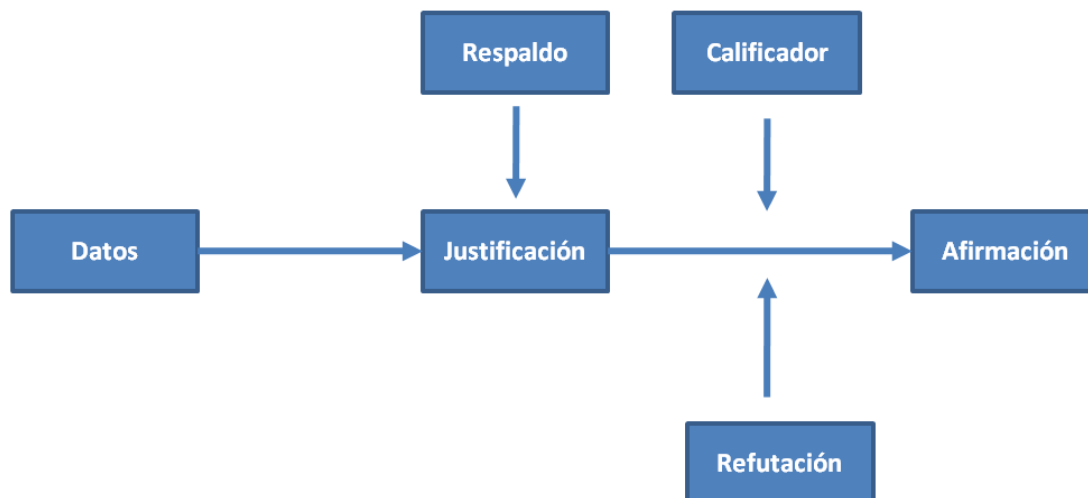


Figura 3.4: "Modelo ABNP". Fuente: Lu y Jing (2009).

En la Figura 3.4, se establece la estructura de un argumento según la concepción de Toulmin (1958). Luego, el modelo ABNP parte de esta visión a los efectos de proporcionar un lenguaje simbólico que permita sostener un proceso de negociación basado en argumentos.

En este contexto, una **Afirmación** se refiere a una aseveración respecto del estado actual de una determinada posición que el *stakeholder* se compromete a defender. Una estructura de **Datos** hace mención a un subconjunto definido de evidencia que soporta una determinada **Afirmación**. La **Justificación** permite legitimar el uso de una estructura de **Datos** definida, a los fines valorativos de una **Afirmación**. El **Respaldo** se refiere a un recurso de autoridad que es utilizado para apoyar y dotar de credibilidad una determinada **Justificación**. Un **Calificador** es quien juzga el grado de fuerza o certeza que una determinada **Afirmación** posee. Por último, la **Refutación** indica las circunstancias y/o excepciones en virtud de las cuales, una determinada **Afirmación** puede fallar o quedar deslegitimada. Por tal motivo, se deberá anticipar tal circunstancia a los efectos de reforzar una determinada **Afirmación** [Lu y Jing (2009)].

3.5.2 Estructura del Proceso de Negociación

El Proceso de Negociación que será aplicado en esta Investigación, se basa en las propuestas desarrolladas por In (et. al. 2002) y Lu y Jing (2009). Como se ha indicado en el Capítulo 2, los enfoques citados parten de la necesidad de establecer una etapa previa a la negociación, en la cual, se elicitan las posiciones asumidas por los diversos *stakeholders* en torno a una determinada situación, acción o circunstancia.

A partir de tal suceso, será factible establecer un esquema de negociación que permita establecer un consenso dinámicamente estable entre los participantes del proceso. A continuación, se ofrece la Estructura del Proceso de Negociación:

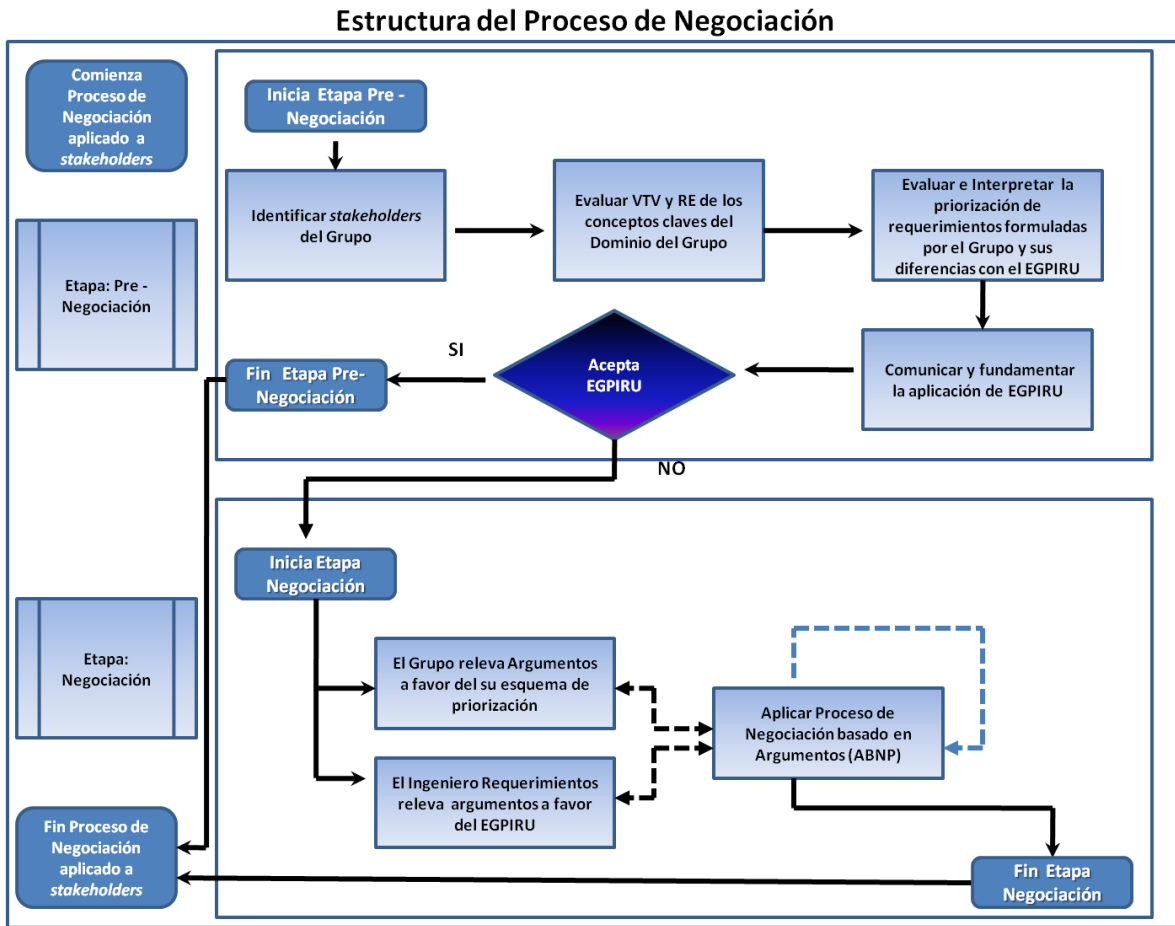


Figura 3.5 “Estructura del Proceso de Negociación”. Fuente: Elaboración Propia.

Como se puede apreciar en la Figura 3.5, el Proceso de Negociación se compone de dos etapas: “Pre- Negociación” y “Negociación”. La primera de ellas, tendrá como objetivo básico constatar las diferencias halladas entre el esquema de priorización establecido por un determinado grupo y que el acontece a nivel general (EGPIRU). Luego del análisis e interpretación de tales diferencias, el Ingeniero en Requerimientos comunicará y fundamentará la aplicación del EGPIRU.

En caso que el grupo en cuestión acepte el EGPIRU, el Proceso de Negociación se dará por finalizado, caso contrario, se inaugurará la Etapa de Negociación. Esta última etapa se caracteriza por una confrontación de argumentos a favor y en contra de la implementación del EGPIRU. Dichos argumentos deben basarse en el Modelo ABNP, considerando:

- a. La *Estructura de Datos* del Modelo ABNP: hace mención al esquema de priorización de requerimientos ofrecidos por el Grupo analizado (EPPIRU)
- b. La *Afirmación* en el Modelo ABNP: hace mención a los argumentos exhibidos a favor de la implementación del esquema de priorización (establecido por los miembros del Grupo)
- c. La *Justificación* en el Modelo ABNP: hace mención a la estructura discursiva establecida por los miembros del Grupo.
- d. El *Respaldo* en el Modelo ABNP hace mención aquellos recursos que los miembros del Grupo citen a los efectos de la defensa de su esquema de priorización.
- e. La *Refutación* en el Modelo ABNP: hace mención a todos aquellos contra-argumentos que el Ingeniero en Requerimientos debería interponer al Grupo, en caso de implementar el esquema de priorización propuestos por ellos.
- f.
El rol de *Calificador* en el Modelo ABNP: lo desempeñará el Ingeniero en Requerimientos dado que detenta la capacidad de valorar los argumentos exhibidos por los miembros del Grupo.

3.6 Elementos de la Teoría de Conjuntos Difusos.

En el mundo de la realidad objetiva no existe la contradicción absoluta; realmente, es parte de un hecho completamente humano y se da sólo en nuestro mundo verbal-conceptual. Asimismo, no existen separaciones nítidas entre las diferentes partes y estados de un sistema ni entre las fases o etapas de los procesos. Esto introduce incertidumbres en el conocimiento de la realidad que han sido siempre asumidas por nuestros modos verbales (mucho, poco, a veces...). Con lo cual, se puede apreciar que los elementos que construyen el pensamiento humano, no son del tipo numérico, sino que más bien, responden a etiquetas lingüísticas.

En este sentido, la Lógica Difusa permite una estructura de trabajo adecuada a los efectos de modelar conceptos inciertos, además de implementar sentencias de lenguaje natural. La modelización de la vaguedad, se logra a través de las variables lingüísticas, lo que permite aprovechar el conocimiento de los expertos [Mallo et. al., (2010)]. Dichas variables lingüísticas tienen su fundamento en escalas del tipo:

Valor de Verdad	Categoría
0	Falso
0.1	Casi Falso
0.2	Bastante Falso
0.3	Algo Falso
0.4	Más Falso que Verdadero
0.5	Tan Verdadero como Falso
0.6	Más Verdadero que Falso
0.7	Algo Verdadero
0.8	Bastante Verdadero
0.9	Casi Verdadero
1	Verdadero

Tabla 3 "Valores de Verdad". Fuente: Mallo (et.al., 2010).

Como se aprecia en la Tabla 3, podría modelarse diferentes grados de verdad de una determinada sentencia, a través de los predicados de la variable analizada. En tal sentido, dicha variable se encontraría expresada en valores de intervalo [0,1]. De esta manera, un valor “1” representaría la verdad absoluta de una sentencia, mientras que el valor “0” la no-verdad absoluta (o la negación de la verdad absoluta).

La Lógica Difusa puede definirse como aquella rama de la Inteligencia Computacional, que permite, entre otras cuestiones, modelar información vaga y de difícil especificación, cuando se la quiere utilizar objetivamente con un fin específico. De esta manera, se convierte en una herramienta para modelar el conocimiento o la comprensión de conceptos [Espín Andrade (et. al. 2004)]. Su flexibilidad la hace apropiada para los sistemas de asistencia en la toma de decisiones, ya que su capacidad para elaborar modelos lingüísticos, la hace muy útil para resolver problemas reales brindando un esquema adecuado para mejorar la comunicación de los que deben tomar decisiones y los expertos que ayudan a clasificarlas mediante sus conocimientos [Mallo, (et. al., 2010)].

Luego, la Lógica Difusa (*Fuzzy Logic*) permite representar el conocimiento común, que es mayoritariamente del tipo lingüístico o cualitativo, en un lenguaje matemático a través de la Teoría de Conjuntos Difusos y las funciones características que son asociadas a ellos. Mientras que la Teoría de Conjuntos Tradicional define ser miembro de un conjunto como un predicado booleano (pertenece o no pertenece a un determinado conjunto), la Teoría de Conjunto Difusos permite representar el ser miembro de un conjunto como una distribución de posibilidades. Por ejemplo, *un conjunto clásico A en el Universo del Discurso U, puede ser definido en términos de la función de pertenencia $\mu_A (X)$ [Ecuación (XX)]:*

$$\mu_A (X) = \begin{cases} 1 & \text{sí } X \in A \\ 0 & \text{sí } X \notin A \end{cases} \quad \text{XX}$$

El conjunto A es matemáticamente equivalente a su función de pertenencia $\mu_A (X)$, ya que conocer $\mu_A (X)$ es lo mismo que conocer A. Por su parte, *un conjunto difuso A en el Universo del Discurso U, se caracteriza por una función de pertenencia $\mu_A (X)$ que toma valores en el intervalo [0.1] y que puede ser representado por un conjunto de pares ordenados de elementos X con su correspondiente valor de pertenencia al conjunto: [Ecuación (XXI)]:*

$$A: \{(x, \mu_a(X)) \rightarrow x \in U\} \quad \text{xxi}$$

Este tipo de razonamiento fue formalizado por primera vez por Lotfi Zadeh en 1965, a través de la Teoría de Conjuntos Difusos [Zadeh (1965)]. Señaló que en muchos casos las clases de objetos que se encuentran en el mundo físico real no tienen criterios de membresía precisamente definidos. Postuló entonces los Conjuntos Difusos, como clases de objetos con grados de pertenencia continuos.

3.6.1 Propiedades de los Conjuntos Difusos:

Al igual que en la teoría de conjuntos tradicional, a los conjuntos difusos se les asocian ciertas propiedades. Los conjuntos difusos que generalmente se utilizan en aplicaciones prácticas son convexos, es decir,

$$\forall a, b \in U; \lambda \in [0,1]: \mu_a [\lambda a + (1 - \lambda)b] \geq \min[\mu_a (a), \mu_a(b)]$$

Existen muchas formas de construir un conjunto difuso convexo, como por ejemplo los que se observan en la Figura 3.6:

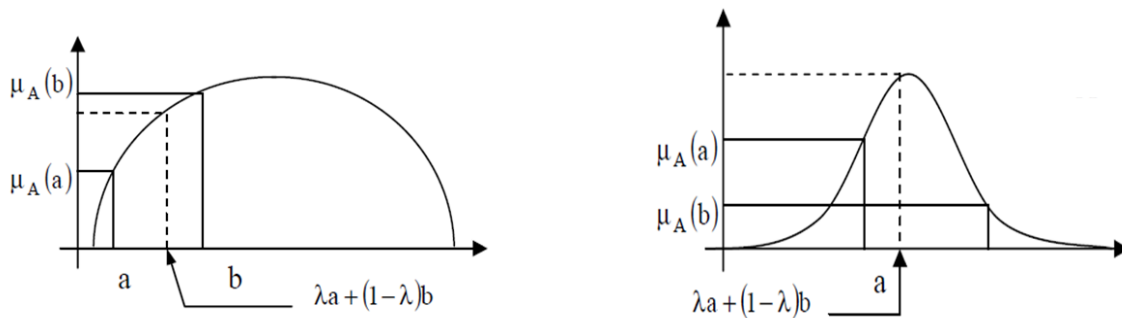


Figura 3.6: "Ejemplos de Conjuntos Difusos Convexos". Fuente: Elaboración Propia.

Núcleo, Soporte y Altura:

En los conjuntos difusos se distinguen el núcleo, que es el conjunto de elementos cuyo grado de pertenencia es igual a 1 (es decir, el rango en que la función de pertenencia normalizada vale 1):

$$\text{Nucleo } (A): \max \{x \in X \mid \mu_a(x) = 1\}$$

El soporte, que es el conjunto de elementos con grado de pertenencia no nulo:

$$\text{Soporte } (A) : \{x \mid \mu_a(x) > 0, x \in X\}$$

Y la Altura, que es el grado de pertenencia más grande de los elementos del conjunto:

$$\text{Altura } (A): \max \{h \mid h = \mu_a(x), x \in X\}$$

Cuantificadores Difusos:

Otra propiedad de los conjuntos difusos es que permiten el uso de cuantificadores difusos. Se usan para medir (o cuantificar) la cantidad o la proporción de objetos o elementos que cumplen o satisfacen cierta condición. En lógica clásica existen dos muy importantes:

- \forall (todo): Se refiere a todos los elementos u objetos.
- \exists (existe): Se refiere al menos a uno de los elementos u objetos.

En Lógica Difusa, existen los Cuantificadores Difusos que se representan como conjuntos difusos con dominio subyacente en los números reales (Zadeh, 1983). El Dominio Subyacente está limitado dependiendo del tipo de cuantificador [Kacprzyk, et. al. (1992); Yager (1983) ; Zadeh, 1983]):

- Cuantificadores Difusos Absolutos: $Q_{\text{abs}}: \mathbb{R}^+ \rightarrow [0,1]$.
 - Se refieren a una única cantidad determinada para medir si esa cantidad son “muchos”, “pocos”, “muchísimos”. Luego, para evaluar la verdad de un cuantificador absoluto se necesita una única cantidad.
- Cuantificadores Difusos Relativos: $Q_{\text{rel}}: [0,1] \rightarrow [0,1]$.
 - Se refieren a una proporción de elementos respecto del total de los que existen. Por ejemplo: “la mayoría”, “la minoría”, “casi todos”, “casi ninguno”, “aproximadamente la mitad”. Luego, para evaluar la verdad se necesitarían 2 cantidades: Los elementos que cumplen la condición y el total de elementos existentes.

Cardinalidad:

Formalmente se define la cardinalidad escalar A de un conjunto A en U como:

$$|A| = \sum_{x \in U} \mu_A(x)$$

Se define la cardinalidad difusa, como un número difuso. Por último, se utiliza la expresión $|A|$ para indicar que se están recorriendo todos los elementos de un conjunto.

Medida de Difusidad:

Existirán conjuntos más o menos difusos, o dicho de otra forma, conjuntos más o menos definidos. Esta propiedad permite establecer una medida de lo difuso (*fuzziness*). En general esta medida dependerá de la aplicación de que se trate. Una medida de lo difuso puede representarse en forma matemática como un mapeo “d” que satisface las condiciones siguientes:

- $d(A) = 0 \leftrightarrow A$ es un conjunto ordinario de U.
- $d(A)$ es máximo $\leftrightarrow \mu_A(x) = \frac{1}{2} \forall x \in U$
- $d(A^*) \leq d(A)$, en que A^* es cualquier versión más definida de A.
- $d(A^*) = d(A)$, es decir que A^* es tan difuso como A.

3.6.2 Operaciones en Conjuntos Difusos

Las operaciones lógicas que se pueden establecer entre conjuntos difusos son: la intersección, la unión y el complemento, igual que las utilizadas en la Lógica clásica. Mientras que el resultado de operar dos conjuntos nítidos es un nuevo conjunto nítido, las mismas operaciones con conjuntos difusos darán como resultado otros conjuntos difusos. Dado que la lógica difusa es una extensión de la clásica, las operaciones para intersectar o unir conjuntos difusos, se realiza de la misma manera. En este sentido, cualquier operación que cumpla las restricciones de una T-Norma puede ser usada para intersectar, igual que cualquier S-Norma puede ser usada para unir conjuntos difusos. Las T-Normas especifican un conjunto de condiciones que deben reunir aquellas operaciones que

deseen ser usadas para intersectar conjuntos, mientras que las S-Normas hacen lo propio para las uniones.

Igualdad: $A = B \Leftrightarrow \mu_A(X) = \mu_B(X) \quad \forall x \in X.$

Inclusión: $A \subseteq B \Leftrightarrow \mu_A(X) \leq \mu_B(X) \quad \forall x \in X.$

Unión: $\mu_{A \cup B}(X) = \max\{\mu_A(X) | \mu_B(X)\}$

Intersección: $\mu_{A \cap B}(X) = \min\{\mu_A(X) | \mu_B(X)\}$

Complemento: $\mu_{\bar{A}}(X) = 1 - \mu_A(X)$

Alfa- Corte $A_\alpha = \{x | \mu_A(x) > \alpha, x \in X\}$

T-Norma: Generalizada del concepto de Intersección:

$T: [0,1] \times [0,1] \rightarrow [0,1]$

$\mu_{A \cap B}(X) = T\{\mu_A(X) | \mu_B(X)\}$

- Conmutativa $T(a,b) = T(b,a)$
- Asociativa $T[a,T(b,c)] = T[T(a,b),c]$
- Monotonía $T(a,b) \geq T(c,d)$, si $a \geq c$ y $b \geq d$
- Condiciones frontera $T(a,1) = a$

S-Norma: Generalizada del concepto de Unión:

$S: [0,1] \times [0,1] \rightarrow [0,1]$

$\mu_{A \cup B}(X) = S\{\mu_A(X) | \mu_B(X)\}$

- Conmutativa: $S(a,b) = S(b,a)$
- Asociativa: $S[a,S(b,c)] = S[S(a,b),c]$
- Monotonía: $S(a,b) \geq S(c,d)$, si $a \geq c$ y $b \geq d$
- Condiciones frontera: $S(a,0) = a$

3.6.3 Tipos comunes de Funciones de Pertenencia

La función de pertenencia $\mu_A(X)$ proporciona una medida del grado de similitud del elemento U con el conjunto difuso. No obstante, su determinación resulta un tanto subjetiva ya que depende de los conceptos que se quieran representar con ella. Es decir, que pueden adquirir distintas formas y en ocasiones se pueden seleccionar con cierto grado de libertad por parte del diseñador, lo que en la práctica se traduce a introducir cierto conocimiento experto [Suarez Fernández (2007)]. En consecuencia, la forma de las

funciones de pertenencia utilizadas en los modelos resulta variada. Pueden ser funciones conocidas: triangulares, trapezoidales, campanas de Gauss (simples o compuestas), polinomiales, sigmoidales, entre otras, e incluso pueden ser funciones específicas con forma personalizada para una determinada variable [Jang, (1997)]. Sin embargo, y tal cual establece Straszeka (2000), generalmente se tenderá a escoger funciones de pertenencia definidas de la forma paramétrica.

- 1.Función Triangular: Definida mediante el límite inferior **a**, el superior **c** y el valor modal **b**, tal que **a<b<c**. Asimismo vale aclarar que la función no tiene porqué ser simétrica [Ecuación (XXII)]:

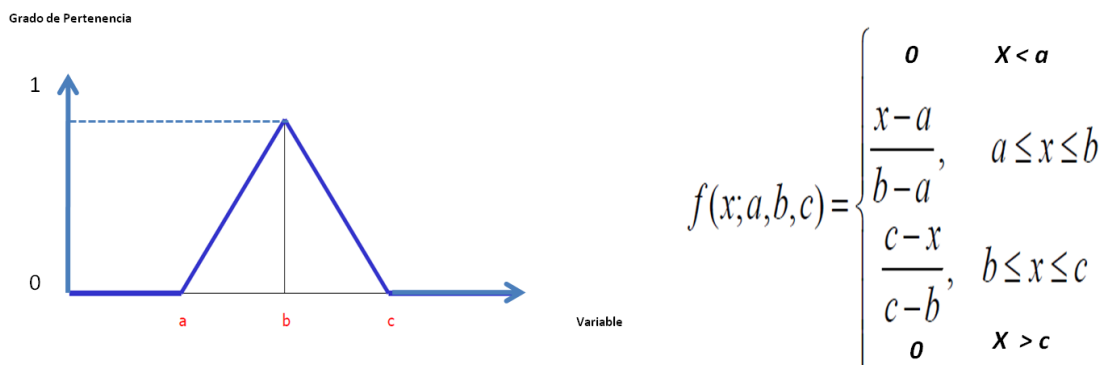


Figura 3.7: "Función de Pertenencia tipo Triangular". Fuente: Elaboración Propia.

Como se puede apreciar en la Figura 3.7, se ejemplifica una función de pertenencia de tipo Triangular [Ecuación (XXII)]. Dependiendo del valor que adquiera los parámetros consignados en la función, el valor de pertenencia oscilará en el rango [0,1].

- 2.Función Trapezoidal: Definida por sus límites inferior **a**, superior **d**, y los límites de soporte inferior **b** y superior **c**, tal que **a<b<c<d**. [Ecuación (XXIII)]:

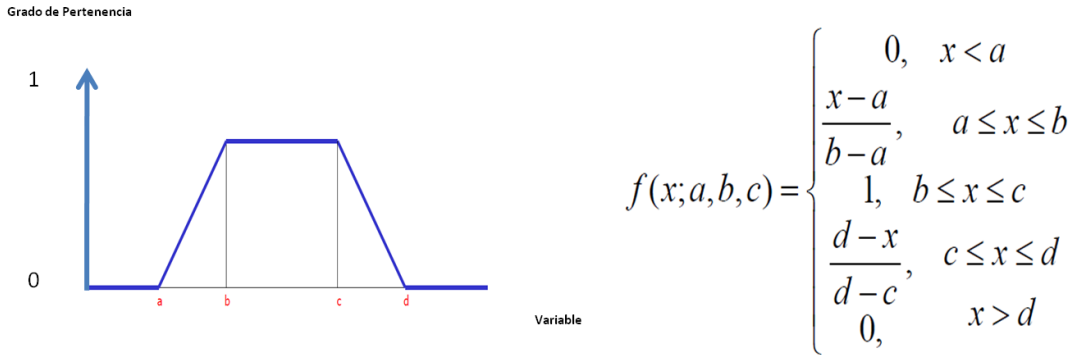


Figura 3.8: "Función de Pertenencia tipo trapezoidal". Fuente: Elaboración Propia.

En la Figura 3.8 se ejemplifica una función de pertenencia de tipo Trapezoidal [Ecuación (XXIII)]. Dependiendo del valor que adquiera los parámetros consignados en la función, el valor de pertenencia oscilará en el rango [0,1]. Se utiliza habitualmente en sistemas difusos sencillos, pues permite definir un conjunto difuso con pocos datos [Arbolera y Chérrez (2006)]. Por su parte, si b y c son iguales, entonces se tratará de una función de pertenencia de tipo triangular.

- **3.Función Sigmoidal:** Definida por sus límites inferior **a**, superior **b** y el valor **m** o punto de inflexión, tales que **a < m < b** [Ecuación (XXIV)]. El crecimiento será más lento cuanto mayor sea la distancia **a-b**. Para el caso concreto de $m=(a+b)/2$, que es lo usual, se obtiene la siguiente gráfica:

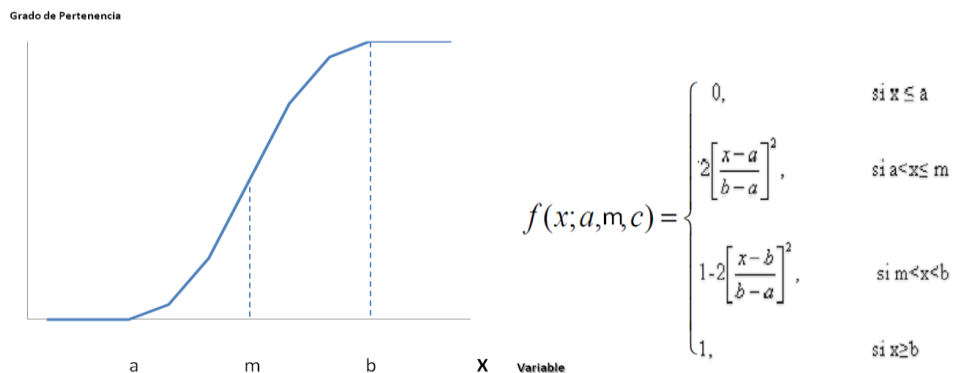


Figura 3.9: "Función de Pertenencia tipo sigmoide". Fuente: Elaboración Propia.

En la Figura 3.9 se puede apreciar el comportamiento que adquiere la función de pertenencia de tipo sigmoïdal [Ecuación (XXIV)]. Se caracteriza por tener un valor de inclusión distinto de 0 para un rango de valores por encima de **a**, siendo 0 para valores menores de **a** y 1 para valores mayores de **b**. Por su parte, su punto de inflexión característico se halla usualmente en **m**, y adquiere un comportamiento de tipo cuadrático (suave) entre **a** y **b** [Arbolera y Chérrez (2006)].

- **4.Función Gaussiana:** Dicha función de pertenencia adquiere una distribución de tipo campana. Como parámetro principal, se define por un valor modal **m**. Asimismo, puede tener una forma leptocúrtica (puntiaguda) o platocúrtica (tipo plato) dependiendo del valor de dispersión de los datos (curtosis) [Ecuación (XXV)].

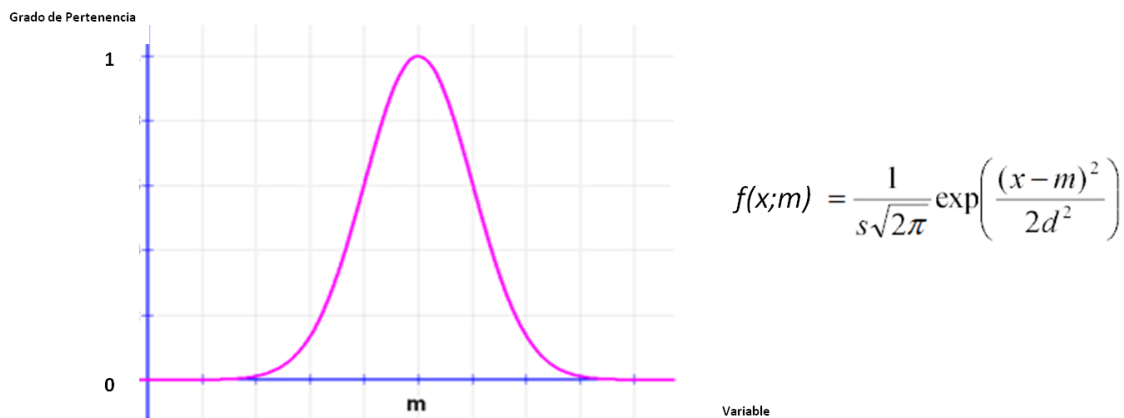


Figura 3.10: “Función de Pertenencia tipo gaussiana”. Fuente: Elaboración Propia.

En la Figura 3.10, se puede apreciar el comportamiento característico de la función gaussiana [Ecuación (XXV)]. Este tipo de función permite modelar fácilmente diversos modificadores, sin embargo, su cálculo resulta complejo [Suarez Fernández (2007)].

3.6.4 Sistemas de Inferencia Difusos

Los sistemas de inferencia difusos son algoritmos que se basan en la lógica difusa y el razonamiento aproximado. En función de la aplicación a la que se destine el sistema de inferencia, éste puede denominarse de distintas formas: modelo difuso, sistema experto difuso, controlador difuso, memoria asociativa difusa, entre otros [Barragán Piña (2009)]. Por su parte, dichos sistemas se componen de dos elementos básicos: El Motor de Inferencia y la Base de Conocimiento. Sin embargo, y tal cual lo menciona Mendel (1995), dependiendo del contexto de aplicación del Sistema de Inferencia Difuso, puede ser necesario la incorporación de un conjunto de *borrosificadores* (*fuzzifier*) y *desborrosificadores* (*defuzzifier*) a los efectos de conectar dicho sistema con el mundo real.

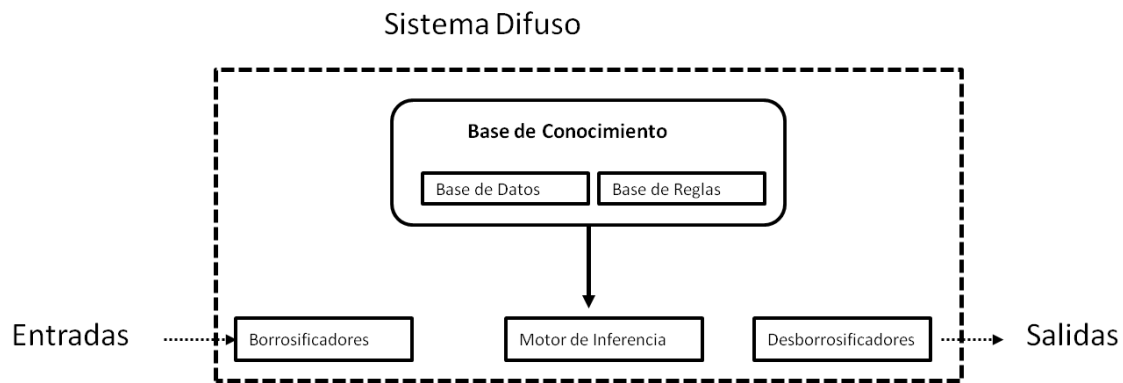


Figura 3.11 "Sistema de Inferencia Difuso". Fuente: Extraído de Barragán Piña (2009).

Como se puede apreciar en la Figura 3.11 se muestran los elementos básicos que pueden encontrarse en un Sistema Difuso. A continuación se describirán sintéticamente cada uno de ellos.

3.6.4.1 Base de Conocimiento

La base de conocimiento (*knowledge base*) almacena relaciones entre las entradas y salidas del sistema. Basándose en este conocimiento y en el motor de inferencia, el sistema es capaz de mapear las relaciones entrada-salidas. Este elemento puede segregarse en dos partes: la base de datos y la base de reglas. La base de datos (*data base*) contiene la definición de las variables lingüísticas empleadas en las reglas, es decir, las funciones de pertenencia que definen cada una de las etiquetas lingüísticas.

La base de reglas (*rule base*) contiene la colección de reglas lingüísticas del sistema. Luego, se define a una regla lingüística como una sentencia condicional que tiene la estructura SI-ENTONCES (IF-THEN). Por ejemplo:

Si la “Velocidad” es Alta y “Congestión en el Transito” es Elevada entonces “la posibilidad de Colisión” es Alta

La premisa de la regla, es decir, la condición, se conoce como antecedente, mientras que la consecuencia se conoce como consecuente. En el ejemplo anterior, “Si la *Velocidad es alta* y la *Congestión en el Transito es elevada*, se corresponde con el antecedente de la regla. Luego, queda determinado por la conjunción lógica de dos variables *fuzzificadas*. Las variables lingüísticas de entrada son: Velocidad y Congestión en el Transito, etiquetadas lingüísticamente en esta regla como “alta” y “elevada”, respectivamente. “*La posibilidad de colisión es alta*” es el consecuente de la regla. Finalmente, el consecuente de una regla puede ser un conjunto difuso, un punto difuso, o una función dependiente de las entradas.

El operador que relaciona la premisa con la consecuencia se conoce como operador de implicación, y se representa lingüísticamente con el adverbio “*entonces*” (*then*). Existen muchas maneras de definir el concepto de implicación. Basándose en cada una de ellas se pueden generar distintas funciones en base a T-normas y S-normas. Sin embargo, los más utilizados son los operadores de Mamdani y Larsen [Turksen (et. al., 1993) citado en Barragán Piña (2009)].

3.6.4.2 Fuzzificación o Borrosificador (*fuzzifier*)

El proceso de *fuzzificación*, consiste en convertir una variable real en el grado de pertenencia en cada uno de los conjuntos difusos definidos en el dominio de dicha variable. Por ejemplo, si una variable de entrada del Sistema es de tipo numérica, será necesario sustituir dicho valor por un conjunto difuso (*fuzzy set*).

3.6.4.3 Motor de Inferencia

El motor de inferencia difuso, es el mecanismo que permite obtener la salida de un sistema difuso en función de sus reglas y las entradas que le sean aplicadas. Esta salida, puede ser tanto un conjunto difuso como un valor numérico, según se requiera por la

aplicación. Asimismo, el motor de inferencia puede verse matemáticamente como un esquema de interpolación no lineal, ya que permite la fusión de la información de múltiples reglas en una conclusión única. Esta conclusión se obtiene a partir de la combinación de los consecuentes de las reglas del sistema, ponderadas según una función del grado de cumplimiento de cada una de ellas [Barragán Piña (2009)].

3.6.4.4 Desborrosificador (*defuzzifier*)

Cuando la salida del sistema de inferencia es un conjunto difuso pero la aplicación requiere que sea un valor numérico concreto, es necesario emplear un desborrosificador o *defuzzificador* (*defuzzifier*). Este elemento convierte dicho conjunto en un valor numérico del universo de discurso de salida, que será representativo de la conclusión obtenida. Si bien existe una variedad de métodos disponibles en la literatura, los más comunes son: el método del centroide, el de máxima pertenencia y el de centroide indexado [Barragán Piña (2009)]:

- *Método del Centroide*: La salida numérica de este *desborrosificador* se obtiene calculando el centro de gravedad del conjunto difuso de salida (y_0). Sea $D(y)$ el conjunto difuso de salida, y considerando que “ y ” es una variable continua, el centroide (y_0) se calcula mediante el cociente entre la integral definida (en el intervalo del conjunto difuso de salida) de la función de pertenencia de dicho conjunto ponderado por la variable “ y ”, y la misma integral definida para la función de pertenencia especificada (Ecuación XXVI):

$$y_0 = \frac{\int_a^b \mu D(y)y dy}{\int_a^b \mu D(y)dy} \quad \text{XXVI}$$

- *Método del centroide indexado*: Este método calcula el centro de gravedad de la parte correspondiente al conjunto difuso inferido, cuyo grado de pertenencia sea mayor de un determinado valor λ .
- *Método de la máxima pertenencia*: Siendo $D(y)$ el conjunto difuso de salida, el valor numérico ya obtenido mediante el *desborrosificador* del máximo, se corresponde con el valor de y para el que $\mu D(y)$ alcanza su máximo valor. Dado que pueden existir múltiples valores por los cuales la función $\mu D(y)$ alcance su máximo, resulta esencial seleccionar otro método adicional para especificar el valor de y . Por ejemplo, centroide de los máximos, mayor de los máximos, entre otros.

3.7 Consideraciones Finales del Capitulo

El presente Capitulo, ha tenido como objetivo básico describir los principales elementos teóricos y metodológicos que serán utilizados en el Modelo de Priorización de Requerimientos de Usuarios, a partir de la segmentación de las preferencias de los *stakeholders* (MPRUSPS). En efecto, se ha escogido el Método AHP, como técnica para el relevamiento de las preferencias de los *stakeholders* (3.2). Luego, los vectores de preferencia hallados por la aplicación de este último método, serán agrupados (segmentadas o agrupados) en función del grado de similitud obtenido entre ellos. Para ello, se utilizará las técnicas: Mapa Auto-Organizado (SOM) y el Índice de Davies-Bouldin como métrica indicativa de la calidad de los agrupamientos hallados por el método de Kohonen. (3.4). Por su parte, se justifica las eventuales configuraciones (grupo o segmentos hallados) en función de los aspectos cognitivos de los *stakeholders* (3.3).

Por último, y haciendo referencia a la necesidad de implementar un proceso de negociación de requerimientos, se ha establecido una fusión entre dos enfoques preexistentes: In (et. al., 2002) y Lu y Jing (2009). (3.5). Luego y a los efectos de modelar un escenario factible de negociación entre grupos, resultaba necesario incorporar alguna metodología que permita gestionar la vaguedad del lenguaje natural, por intermedio del cual, sería factible describir alguna hipótesis que permita mediante su verificación, iniciar el Proceso de Negociación. En consecuencia, se ha descrito en 3.6 los principales elementos de las Teoría de Conjuntos Difusos, la que se considera una herramienta adecuada para operar este tipo de hipótesis.

Finalmente, en el próximo Capitulo se desarrollará el “Modelo de Priorización de Requerimientos de Usuarios a partir de la Segmentación de las Preferencias de los Stakeholders (MPRUSPS)”. Para ello, se citarán todos los elementos descritos en la presente sección.

Capítulo 4

“Modelo de Priorización de Requerimientos de Usuarios a partir de la segmentación de las Preferencias de los Stakeholders (MPRUSPS)” Un enfoque integrado.

4.1 Introducción

El presente Capitulo, desarrollará el **Modelo de Priorización de Requerimientos de Usuarios a partir de la Segmentación de Preferencias de los Stakeholders (MPRUSPS)**. Para ello, se propone una integración de los elementos teóricos y metodológicos desarrollados en el Capitulo anterior. En consecuencia, se establecerán una serie de pasos secuenciales que el Ingeniero en Requerimientos debería ejecutar a los efectos de satisfacer el propósito general del mismo: **Lograr constituir un Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU) validado y consensuado por los stakeholders del Proyecto de Software.**

Para que el presente modelo pueda ejecutarse eficazmente, es necesario satisfacer **tres supuestos esenciales**. En primer lugar, se debe partir de una Especificación de Requerimientos validada por la totalidad de los *stakeholders* que participaron en su construcción (**Supuesto 1**). Por su parte, cada Requerimiento de Usuario que se desprenda de dicha Especificación, puede ser priorizado en forma autónoma. Es decir, su importancia radica en sí mismo, y se considera independiente de los restantes (**Supuesto 2**). Finalmente, se considera que cada *stakeholder* tiene el mismo peso relativo en la formulación del plan de priorización de requerimientos (**Supuesto 3**).

4.1.2 Arquitectura General del MPRUSPS

El Modelo de Priorización de Requerimientos de Usuarios a partir de la Segmentación de Preferencias de los *Stakeholders* (MPRUSPS), establece una serie de etapas, con el fin básico de formular y consensuar, el orden relativo de implementación de los Requerimientos de Usuarios disponibles en una determinada Especificación. Para ello, el modelo presenta una arquitectura, mediante la cual, será factible conocer como las diferentes etapas consignadas en él, son capaces de satisfacer este propósito básico.

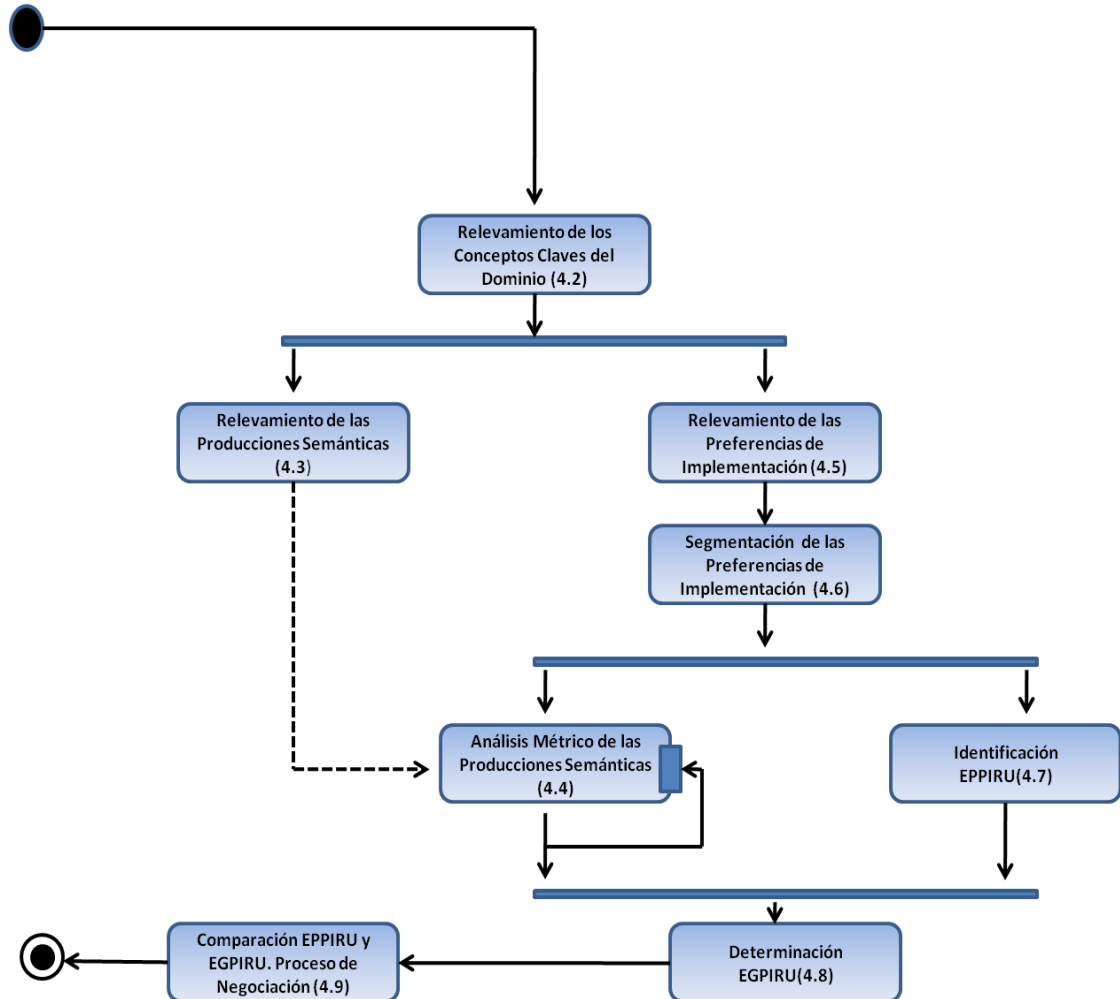


Figura 4.1: "Arquitectura del MPRUSPS". Fuente: Elaboración Propia.

En la Figura 4.1, se puede apreciar un Diagrama de Actividades que muestra de manera simplificada, como se relacionan y se ejecutan las diferentes etapas del Modelo. Habiendo satisfecho los supuestos del mismo, será factible el desarrollo de cada una de las actividades consignadas en él.

La primera actividad del modelo consiste en relevar aquel conjunto de conceptos claves, que según la óptica de los *stakeholders*, servirían para comprender las implicancias y usos de la Herramienta de Software en el dominio de aplicación (4.2). Luego del cumplimiento de dicha etapa, será factible el desarrollo concurrente de las actividades (4.3) y (4.5). El relevamiento de las producciones semánticas (4.3), implica un tratamiento socio-cognitivo de cada uno de los conceptos claves del dominio (4.2). Por su parte, el relevamiento de las preferencias de implementación (4.5) implica elicitar el orden de preferencia que cada uno de los *stakeholders*, ha decidido asignar a cada uno de los requerimientos de usuarios disponibles. A partir de esta última actividad, será factible segmentar o agrupar las preferencias de los *stakeholders*, según el patrón de comportamiento que ellas exhiban (4.6). El Análisis Métrico de las Producciones Semánticas (4.4) implica un estudio socio-cognitivo en cada uno de los grupos configurados a partir de (4.6). A la vez, se procura determinar en cada uno de ellos, el Esquema Particular de Prioridad Implementable de Requerimientos de Usuarios (EPPIRU) (4.7).

Luego de ejecutadas estas dos últimas actividades (4.4 y 4.7), se procederá a calcular el Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU en 4.8). A partir de ésta última actividad, se evaluará el grado de distancia que exista entre cada EPPIRU y el EGPIRU, y eventualmente, se aplicará un Proceso de Negociación en aquellos grupos consignados como “distantes” (4.9). A continuación se describirán cada una de estas Actividades.

4.2 Relevamiento de los conceptos claves del dominio.

En esta fase, se pretende relevar el conjunto de conceptos, que según la percepción de los *stakeholders*, resultarían representativos a los efectos de comprender los usos de la herramienta de software y sus implicaciones. Para ello, el Ingeniero en Requerimientos deberá entrevistarse con los *stakeholders* considerando los siguientes pasos:

1. Consultar a cada uno de *los stakeholders* del Proyecto, respecto de aquellos conceptos, que estima, resultarían representativo a los efectos de comprender los usos e implicancias de la futura herramienta software. (Metodología del relevamiento –Entrevista directa).
2. A partir de 1, elaborará una Lista Tentativa de los *Conceptos claves del dominio*. (LTCCD).
3. Distribuirá a cada uno de los *stakeholders*, una copia de la LTCCD (Punto 2). Luego, cada *stakeholders* valorará a cada uno de los conceptos indicando “1” (*voto favorable*) si lo considera esencial y “0” si estima lo contrario (*voto no favorable*).
4. Tomando en consideración los votos favorables de cada concepto de la LTCCD (Frecuencia Absoluta), el Ingeniero en Requerimientos los ordenara de mayor a menor.
5. Se consideraran como Conceptos Claves del Dominio (CCD) aquel subconjunto de conceptos ordenados, cuya Frecuencia Relativa Acumulada alcance al menos el 51% del total de votos favorables (mayoría absoluta).

4.3 Relevamiento de las Producciones Semánticas de los conceptos claves del Dominio.

Considerando los conceptos claves del dominio (CCD) especificados en 4.2, se procederá a elicitación la producción semántica de cada uno de ellos. Para ello, resulta esencial que cada *stakeholders* del universo responda la siguiente pregunta para cada uno de los conceptos en cuestión (CCD) [Capítulo 3. Sección 3.3]

En relación a cada uno de los conceptos claves del Dominio. ¿Cuales serian las primeras 10 palabras (en orden de importancia) que UD cree servirían para identificar a cada uno de ellos?

De esta manera, cada *stakeholder* elaborará una Lista de Definidoras (10 palabras) por cada concepto (CCD). A partir de allí, se procesará el total de Listas utilizando la herramienta de software “*Definition Finder*” (Capítulo 3. Sección: 3.3.1). En consecuencia, se obtendrá la producción semántica de cada concepto (CCD) para el universo de *stakeholders*. En términos matemáticos, la producción semántica de un concepto dado, hace referencia a una sucesión ordenada de pesos relativos (iguales o mayores a 0) inherentes a la sucesión de definidoras del concepto analizado.

Al respecto, cabe recordar que la producción semántica de un concepto dado depende fundamentalmente del conjunto de *stakeholders* que se considere. En efecto, puede ocurrir que la producción semántica de un concepto para un subconjunto de *stakeholders*, resulte diferente de aquella que surja a nivel global.

4.4 Análisis Métrico de las Producciones Semántica de los conceptos claves del Dominio

A partir de la etapa anterior (4.3), se obtendrá la sucesión de pesos relativos de cada uno de los conceptos (CCD). Luego, cada sucesión se utilizará como base de cálculo del *Método: “Valor Total de las Valorizaciones” (VTV)*, obteniendo así, el campo semántico de cada uno de los conceptos (Capítulo 3. Sección: 3.3.2). También, se calculará la *“Razón de Exclusión” (RE)* en cada uno de ellos (Capítulo 3 en Sección: 3.3.3).

De esta manera, el Ingeniero en Requerimientos podrá disponer de las métricas: “VTV”, “RE” de cada concepto clave del dominio (CCD). Luego, calculará los promedios de de VTV y RE para el total de conceptos analizados (n: cantidad de conceptos). En efecto aplicará:

$$\begin{aligned} \text{a. } \overline{VTV} &= \frac{\sum_{i=1}^{i=n} VTV_i}{n} \quad [\text{Ecuación (XXVI)}] \\ \text{b. } \overline{RE} &= \frac{\sum_{i=1}^{i=n} RE_i}{n} \quad [\text{Ecuación (XXVII)}] \end{aligned}$$

Finalmente, el Ingeniero en Requerimientos elaborara una Tabla denominada: *Síntesis de las Producciones Semánticas de los Conceptos Claves del Dominio (SPSCCD)*. La misma tendrá el siguiente formato (Tabla 4):

SPSCD		
Conceptos	VTV	RE
1	VTV_1	RE_1
2	VTV_2	RE_2
...	$VTV_{...}$	$RE_{...}$
N	VTV_n	RE_n
Promedios	\overline{VTV}	\overline{RE}

Tabla 4: “Ejemplo del cuadro de Síntesis de las Producciones Semánticas de los conceptos claves del dominio (SPSCCD)”.

Fuente Elaboración Propia.

4.5 Relevamiento de las preferencias de implementación de los Requerimientos de Usuarios.

En forma simultánea a la Etapa 4.3, el Ingeniero en Requerimientos deberá relevar para cada *stakeholders*, la función de preferencia de implementación de los requerimientos de usuarios. Para ello, utilizará el Método AHP (Capítulo 3. Sección: 3.2.1), considerando los siguientes pasos:

1. El Ingeniero en Requerimientos le solicita al *stakeholder* que valore de pares, los requerimientos de usuarios teniendo en consideración la escala provista por Karlsson (1996). (Tabla 2. Capítulo 3. Sección: 3.2.1).
2. En base a lo anterior, el Ingeniero en Requerimientos confecciona la matriz de comparaciones [Ecuación (I)] (Capítulo 3. Sección: 3.2.1).
3. Luego, estima el vector de preferencia [Ecuación (II)] del *stakeholder*. (Capítulo 3. Sección 3.2.1).
4. El Ingeniero en Requerimiento analiza el Nivel de Inconsistencia (CR) obtenido por el *stakeholder* [Ecuación (VIII)] (Capítulo 3. Sección 3.2.1)

Sí $CR > 0.1$ entonces

- i. Analizar esquema de preferencia del *stakeholder* (paso 1)
- ii. Evaluar errores en Transitividad de Preferencia (paso 2)
- iii. Informar al *stakeholder* respecto de los errores incurridos (Evaluar la posibilidad de recapacitar en el uso del método AHP al corriente *stakeholder*).
- iv. Rehacer paso 1 hasta lograr $CR \leq 0.1$ en la preferencia relevada por el *stakeholder*.

SI- NO

- i. El Ingeniero en Requerimiento, considera como función de preferencia del *stakeholder* aquella obtenida en el paso 3.

Finalizado con este bloque de instrucciones, el Ingeniero de Requerimiento confecciona la Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios (LFDPIRU), para el total de *stakeholders*:

LFDPIRU						
Stakeholders	RU ₁	RU ₂	RU ₃	RU ₄	RU ₅	RU _n
S ₁	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}
S ₂	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}
S ₃	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}
S ₄	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}
S _n	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}

Tabla 5: "Ejemplo de la Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios (LFDPIRU). Fuente Elaboración Propia.

Como se aprecia en la Tabla 5, a cada *stakeholder* le corresponde un vector de preferencia de implementación de requerimientos de usuarios (hallado en el paso 3). Luego, cada P_{RU_i} indica la frecuencia relativa o proporción de preferencia que un determinado *stakeholder*, le asigna a un Requerimiento de Usuario de orden i . En consecuencia:

$$S_i = [P_{RU_1} \dots P_{RU_n}]$$

En donde:

S_i : se refiere al *stakeholder* de orden i -ésimo.

$[P_{RU_1} \dots P_{RU_n}]$ = vector de preferencia de implementación de requerimientos de usuarios.

De manera tal que:

- $0 \leq P_{RU_i} \leq 1$
- $\sum_{i=1}^n P_{RU_i} = 1$

4.6 Segmentación (agrupamiento) de las Preferencias de los stakeholders.

Una vez obtenida la Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios (LFDPIRU en 4.5), el objetivo siguiente consistirá en segmentar (agrupar) a los *stakeholders*, de acuerdo al grado de similitud existente entre sus respectivos vectores de preferencias de implementación de requerimientos de usuarios. A tal efecto, se utilizará el método: Mapas Auto-organizados de Kohonen (Capítulo 3. Sección: 3.4.2), considerando los siguientes aspectos:

1. Cada *stakeholder* dispone de un vector de preferencia de implementación de requerimientos de usuarios. $S_i = [P_{RU_1} \dots P_{RU_n}]$. Con lo cual existirán “n” P_{RU} que deberán ser consideradas como variables de entrada del modelo.
2. Existe “p” cantidad de *stakeholders*.
3. La matriz de datos (LFDPIRU en 4.6) tendrá una dimensión inicial de “p” (cantidad de *stakeholders*) x “n” (cantidad de requerimientos de usuarios) y será utilizada durante el proceso de entrenamiento del Modelo.
4. El Modelo utilizará como criterio de similitud la distancia euclídea: (Capítulo 3. Ecuación XIV. Sección: 3.4.2). Asimismo, como salida de la Red se determinará un mapa bidimensional cuya dimensiones se ajustaran luego de conocer empíricamente la matriz de datos.
5. Tomando en consideración la regla de actualización de Kohonen (Capítulo 3. Ecuación XV. Sección: 3.4.2), se procederá a entrenar la red con al menos 100 interacciones, en donde:
 - α_t : la tasa de aprendizaje comenzará teniendo un valor máximo igual 1 e irá decreciendo en forma monótona a lo largo de las interacciones, hasta alcanzar un nivel mínimo y constante de 0,05.
 - $h_{r,t}$: el radio de vecindad comenzará teniendo un valor máximo igual al diámetro del mapa (especificado luego de conocer la cantidad de requerimientos a evaluar) e irá decreciendo en forma monótona a lo largo de las interacciones hasta alcanzar un nivel mínimo y constante de 1.
6. Una vez finalizado el proceso de entrenamiento del modelo, se evaluará la Matriz Unificada resultante. Es decir, se inspeccionará visualmente la cantidad de grupos existentes bajo el criterio de similitud del patrón de preferencia de implementación de requerimientos de usuarios. Acto seguido, se procederá a

calcular el Índice de Davies-Bouldin (Capítulo 3. Sección: 3.4.3) a los efectos de evaluar la bondad de las clasificaciones efectuadas por la red de Kohonen.

En función de los resultados obtenidos en el proceso anterior, el Ingeniero en Requerimiento deberá elaborar la Tabla: Grupos y Composición (GyC), utilizando los resultados del agrupamiento de los mapas auto-organizados. Cabe destacar que si luego de aplicar los pasos 1-6 de la presente Etapa, no resulta factible discernir una serie finita de subgrupos de *stakeholders*, el Ingeniero en Requerimientos considerará que el universo de *stakeholders* queda configurado en un sólo Grupo. Sin perjuicio de ello, y considerando la totalidad de casos que se presenten, el formato ejemplo utilizado para tabular los datos será el siguiente (Tabla 6):

Grupo I						
Stakeholders	RU ₁	RU ₂	RU ₃	RU ₄	RU ₅	RU _n
S ₁	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}
S ₂	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}
S ₃	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}
Grupo II						
Stakeholders	RU ₁	RU ₂	RU ₃	RU ₄	RU ₅	RU _n
S ₄	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}
S ₅	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}
S ₆	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}
S ₇	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}
Grupo III						
Stakeholders	RU ₁	RU ₂	RU ₃	RU ₄	RU ₅	RU _n
S ₈	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}
S ₉	P _{RU1}	P _{RU2}	P _{RU3}	P _{RU4}	P _{RU5}	P _{RU_n}

Tabla 6: "Ejemplo de la Tabla "Grupo y Composición". Fuente Elaboración Propia.

4.7 Identificación de Esquemas Particulares de Prioridad Implementable de Requerimientos de Usuarios (EPPIRU)

Luego de haber efectuado la segmentación de preferencias de los *stakeholders* (4.6), se establecerá el valor medio de preferencia de cada uno de los Requerimientos Usuarios valorados en cada uno de los Grupos identificados. Para ello, el Ingeniero en Requerimientos deberá aplicar la siguiente función en cada uno de los Grupos hallados en el proceso anterior y para Requerimiento de Usuario en particular [Ecuación (XXVIII)]

$$VMP_{RU(n)p} = \frac{\sum_{s=1}^{s=m_p} [P_{RU(n)(s)}]}{m_p} \text{XXVIII}$$

En donde:

$VMP_{RU(n)p}$ = representa el valor medio de preferencia de implementación del Requerimiento de Usuario de orden “n” del Grupo p.

m_p = representa la cantidad de *stakeholders* clasificados en el Grupo “p”.

$PRU_{(n)(s)}$: Preferencia de implementación del Requerimiento de Usuario de orden “n”, que el *stakeholder* de orden “s” le ha asignado.

Habiendo realizado el cálculo precedente, el Ingeniero en Requerimientos elaborará la Lista de Grupos con sus correspondientes Valores Medio de Preferencia de Implementación de Requerimientos de Usuarios ($LGVP_{RU}$). A continuación, se ofrece un formato ejemplo para tabular dichos datos (Tabla 7):

LGVM _{RU}						
Grupo	RU ₁	RU ₂	RU ₃	RU ₄	RU ₅	RU _n
1	VMP _{RU1}	VMP _{RU2}	VMP _{RU3}	VMP _{RU4}	VMP _{RU5}	VMP _{RU_n}
2	VMP _{RU1}	VMP _{RU2}	VMP _{RU3}	VMP _{RU4}	VMP _{RU5}	VMP _{RU_n}
...	VMP _{RU1}	VMP _{RU2}	VMP _{RU3}	VMP _{RU4}	VMP _{RU5}	VMP _{RU_n}
N	VMP _{RU1}	VMP _{RU2}	VMP _{RU3}	VMP _{RU4}	VMP _{RU5}	VMP _{RU_n}

Tabla 7: "Ejemplo de Lista de Grupos con sus correspondientes Valores Medio de Preferencia de Implementación de Requerimientos de Usuarios (LGVM_{RU})". Fuente: Elaboración Propia

En base a la LGVM_{RU}, el Ingeniero en Requerimientos determinará el Esquema Particular de Prioridad Implementable de Requerimientos de Usuarios (EPPIRU) en cada uno de los Grupos. Para ello:

1. Ordena los elementos VMP_{RU_n} de mayor a menor.
2. Asigna orden de prioridad implementable a los Requerimientos de Usuario. Correspondiéndole al elemento de mayor valor medio, el orden 1º y al menor el orden enésimo.
3. El EPPIRU del Grupo queda determinado por un vector que ordena a los Requerimientos de Usuario de acuerdo a su prioridad de implementación (1^{ero}, 2^{do}, ...enésimo).
4. Finalmente, el Ingeniero en Requerimientos confecciona la Lista de Grupos con sus correspondientes Esquema Particular de Prioridad Implementable de Requerimientos de Usuario (LEPPIRU):

$$LEPPIRU = \{ [EPPIRU_p] \forall p = 1 \dots n \}$$

En donde:

$$EPPIRU_p = \{ [(o^o)VMP_{Ru(t)}] \forall o^o = 1 \dots n / o^o \rightarrow VMP_{Ru(t)} \forall t = 1 \dots RU_n \}$$

4.8 Determinación del Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU)

En forma análoga a lo establecido 4.7, se procede a calcular el Valor Medio de Prioridad Implementable de los Requerimientos de Usuarios para el total de Grupos. Para ello, se utilizará la Lista de Grupos con sus respectivos Valores Medio de Preferencia de Implementación de Requerimientos de Usuarios ($LGVMP_{RU}$ en Tabla 7). Luego, el Ingeniero en Requerimientos deberá seguir los siguientes pasos:

1. Vincular las Listas de Definidoras correspondientes a un mismo Grupo de *stakeholders* y obtener la producción semántica de cada uno de los conceptos claves del dominio. Para ello, se debe seguir los pasos consignados en 4.4 considerando los grupos hallados en 4.6.
2. A partir de lo anterior, se calculan VTV y RE (Capítulo 3. Secciones: 3.3.2 y 3.3.3 respectivamente) para cada uno de los conceptos claves del dominio. Luego, se calculan sus respectivos promedios.
3. Finalizado lo anterior, el Ingeniero en Requerimientos elaborará una Tabla denominada: *Síntesis de las Producciones Semánticas de los Conceptos Claves del Dominio por Grupo (SPSCCDG)* (Tabla 8):

SPSCCDG		
<i>Grupo</i>	\overline{VTV}	\overline{RE}
1	\overline{VTV}_1	\overline{RE}_1
2	\overline{VTV}_2	\overline{RE}_2
....	$\overline{VTV}...$	$\overline{RE}...$
<i>N</i>	\overline{VTV}_n	\overline{RE}_n

Tabla 8: "Ejemplo del cuadro de Síntesis de las Producciones Semánticas de los conceptos claves del dominio por Grupo (SPSCCD)". Fuente Elaboración Propia.

4. En esta instancia, el Ingeniero en Requerimientos procede a calcular los Valores Medios de Prioridad Implementable de cada uno de los Requerimientos de Usuarios. Para ello, debe aplicar: [Ecuación (IXXX)]

$$VMP_{RU(i)total} = \sum_{p=1}^{p=k} \left(\frac{\overline{VTV}_p}{\overline{VTV}_{Acumulada}} \right) \cdot VMP_{RU(i)p}$$

En donde:

$VMP_{RU(i)total}$ = representa el valor medio de preferencia total del Requerimiento de Usuario de orden i.

K = cantidad de grupos de *stakeholders*.

$VMP_{RU(i)p}$ = representa el valor medio de preferencia del Requerimiento de Usuario de orden i asignado por el grupo de orden p.

\overline{VTV}_p = representa la VTV promedio de los conceptos claves del dominio establecido para el grupo de orden p.

$\overline{VTV}_{Acumulada}$ = representa la sumatoria de las \overline{VTV} registradas en cada grupo.

5. En base a los resultados de la ecuación anterior, el Ingeniero en Requerimientos obtiene el vector de Valores Medio de Prioridad de los Requerimientos de Usuario:

$$VMP_{RU} = \{ [VMP_{Ru(n)Total}] \forall n = 1 \dots RU_n \}$$

6. Se determina el Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU):
 - a. En base a 5, se reordena los elementos $VMP_{Ru(n)Total}$ de mayor a menor.
 - b. Luego, se asigna orden de prioridad implementable a los Requerimientos de Usuario. Correspondiéndole al elemento de mayor valor medio, el orden 1º y al menor el orden enésimo.
 - c. El EGPIRU queda determinado por un vector que ordena a los Requerimientos de Usuario de acuerdo a su prioridad de implementación (1^{ero}, 2^{do}, ...enésimo).

4.9 Comparación entre el Esquema General (EGPIRU) y cada Esquema Particular (EPPIRU). Análisis de la Implementación de un Proceso de Negociación Grupal.

En esta instancia, resulta necesario disponer de alguna técnica que permita especificar el grado de distancia que pueda existir entre el centroide de las preferencias globales (EGPIRU en 4.8) y los respectivos centros de las preferencias grupales (EPPIRU en 4.7). Para ello, se propone la especificación de un **Sistema de Memoria Asociativa Difusa (FAM)**. Para ello, se estima conveniente precisar algunas variables previas, que deberían constituirse, antes de formalizar dicho sistema.

En primer lugar, será necesario determinar la cantidad de requerimientos de usuarios que totalizarían al menos el 50 % superior de las preferencias agregadas. Para ello, y contando con el EGPIRU, se asignará a cada requerimiento ordenado su correspondiente nivel de preferencia. Luego, se procederá a sumar cada uno de las preferencias consignadas hasta alcanzar el nivel previamente especificado. Por ejemplo:

Orden	EGPIRU	Preferencia Agregada (%)	Preferencia Acumulada (%)
1º	5	25	25
2º	4	20	45
3º	3	15	60
4º	2	14	74
5º	1	12	86
6º	6	8	94
7º	7	6	100

Tabla 9 "Ejemplo para la determinación de las preferencias acumuladas en el EGPIRU". Fuente Elaboración Propia

Como se puede apreciar en la Tabla 9, el EGPIRU establece un orden hipotético de 7 Requerimientos. Considerando la preferencia de cada uno de ellos, se determinará aquel subconjunto de requerimientos que al menos supere el 50% de las preferencias acumuladas. Dado que los 2 primeros requerimientos (5 y 4) no satisfacen dicha condición (acumulan el 45 % de las preferencias), se agrega un tercer requerimiento (3). *En tal sentido, se establece que los 3 primeros requerimientos (5, 4, 3) totalizan al menos el 50 % de las preferencias agregadas.* A este punto, se lo denomina como: "Orden de Corte" (OC).

Considerando esta vez un determinado EPPIRU, se especificará el peso relativo que adquiere las preferencias acumuladas hasta el **Orden de Corte**. Por ejemplo:

Orden	EPPIRU(G1)	Preferencia Agregada (%)	Preferencia Acumulada (%)
1º	5	35	35
2º	4	25	60
3º	2	20	80
4º	3	8	88
5º	1	5	93
6º	7	4	97
7º	6	3	100

Tabla 10 "Ejemplo: cálculo del peso relativo de los primeros requerimientos de un EPPIRU". Fuente Elaboración Propia

Como se puede apreciar en la Tabla 10, el nivel de preferencia acumulado por el grupo hasta el Orden de Corte asciende al 80 %. No obstante a ello, se considera importante establecer el peso relativo que adquiere estos 3 primeros requerimientos (5, 4, 2) en el EGPIRU. En efecto, dichos requerimientos alcanzan el 59 % de preferencia en este último esquema (RU5: 25%, RU4: 20%, RU2: 14% en Tabla 9). Luego, el **Nivel de Preferencia del Grupo (NPG)** quedará determinado por el cociente entre ésta última cifra (59%), y la correspondiente tras acumular las preferencias en el EGPIRU hasta el Orden de Corte (60%). En este caso, $NPG = 0.98$

Por su parte, resulta importante conocer el grado de coincidencia existente entre el orden de requerimientos propuesto en un EGPIRU y aquel que acontezca en un determinado EPPIRU. Como base de comparación, se considerarán los requerimientos contenidos en el intervalo: [1...**Orden de Corte**].

Orden	EGPIRU	EPPIRU(G1)
1º	5	5
2º	4	3
3º	3	2
4º	2	3
5º	1	1
6º	6	7
7º	7	6

Tabla 11 "Ejemplo del Grado de Coincidencia entre el EGPIRU y un EPPIRU". Fuente Elaboración Propia

En base al ejemplo desarrollado hasta al momento, puede apreciarse en la Tabla 11 los tres niveles de coincidencia que podrán establecerse entre el EGIPIRU y un determinado EPPIRU: *Coincidencia Exacta, Relativamente Exacta o No-Coincidencia*. Se establece un nivel de **Coincidencia Exacta** cuando el Orden del Requerimientos n-ésimo en el EGIPIRU coincide con el Orden del Requerimientos n-ésimo en el EPPIRU. Por ejemplo, el Requerimiento quinto se propone como 1º en ambos esquemas (ver en Tabla 11). Por su parte, un nivel de **Coincidencia Relativamente Exacto**, es aquel que surge cuando un Requerimiento de orden n-ésimo, se encuentra ubicado en ambos esquemas (EGIPRU, EPPIRU) dentro del intervalo comprendido entre el primer y el n-ésimo **Orden de Corte**, pero no comparten el mismo orden de posición. Por ejemplo, el EPPIRU establece en segundo lugar el Requerimiento tercero, mientras que, dicho requerimiento, ocupa el tercer lugar en el EGPIRU. En este caso, se denomina que la coincidencia es “Relativamente Exacta” puesto que en ambos Esquemas (EGPIRU y EPPIRU) el tercer requerimiento se encuentra contenido dentro del **Orden de Corte**. Luego, se encuentra el caso de **No Coincidencia**. Es decir, el Requerimiento de Orden n-ésimo propuesto por el EPPIRU, no encuentra correspondencia en el EGPIRU dentro del rango establecido por el **Orden de Corte**. Por ejemplo, el Requerimiento segundo ocupa el tercer lugar en el EPPIRU, sin embargo, dicho Requerimiento se ubica en el cuarto lugar en el EGPIRU (orden mayor al **Orden de Corte**).

Respecto a la ponderación asignada a cada nivel de coincidencia, se establece: 1 para Coincidencia Exacta, 0.5 para Relativamente Exacta y 0 para No-Coincidencia. Finalmente, el **Grado de Coincidencia del Grupo (GCG)** se define como el cociente entre la sumatoria de los niveles de coincidencia y el **Orden de Corte**. En este caso, $GCG = 0.5$.

Aclarado estos aspectos, a continuación se procederá a definir cada una de las variables que componen el **Sistema de la Memoria Asociativa Difusa (FAM)**

a). Variable de Entradas del Sistema

1. $NPG : \{ NPG; T(NPG); U; G; M \}$

En Donde:

- *Nivel de Preferencia del Grupo (NPG)*: Como se ha mencionado, se refiere a una variable de tipo numérica (continua) que se obtiene a partir del cociente entre el peso relativo que adquiere la preferencia acumulada en los “n” Requerimientos del EPPIRU (1º hasta Orden de Corte) en el EGPIRU, y el correspondiente tras acumular la preferencia (1º hasta Orden de Corta) en el EGPIRU.

- $T(NPG)$: Representa el conjunto de valores lingüísticos:
 $\{\text{"Sub-Valorado"}; \text{"Igualmente Valorado"}; \text{"Sobre-Valorado"}\}$
- U : Se refiere al Universo del Discurso:
 $[0..2]$
- G : Especifica la forma que se genera los valores en $T(NPG)$:
 $\{\text{gaussiana}; \text{gaussiana}; \text{gaussiana}\}$
- M : Representa la forma de cada uno de los conjuntos difusos asociados a los valores lingüísticos (parámetros de la función de pertenencia):
 $\{\text{"Sub-Valorado"}: [0.442; -0,04231]; \text{"Igualmente Valorado"}: [0.4349; 0.988];$
 $\text{"Sobre-Valorado"}: [0.432; 2]\}$

2. $GCG : \{ GCG; T(GCG); U; G; M \}$

En Donde:

- *Grado de Coincidencia del Grupo (GCG)*: Como se ha mencionado, se refiere a una variable de tipo numérica (continua) que se obtiene a partir del cociente entre la sumatoria numérica que se le otorga a cada Nivel de Coincidencia y el Orden de Corte. Para una Coincidencia Exacta, se debe sumar una Unidad, para una Coincidencia Relativamente Exacta, se pondera media Unidad (0,5), y finalmente, para una No-Coincidencia, se debe adicionar 0 Unidad.
- $T(GCG)$: Representa el conjunto de valores lingüísticos:
 $\{\text{"Bajo"}; \text{"Moderado"}; \text{"Alto"}\}$
- U : Se refiere al Universo del Discurso:
 $[0..1]$
- G : Especifica la forma que se genera los valores en $T(GCG)$:
 $\{\text{gaussiana}; \text{gaussiana}; \text{gaussiana}\}$
- M : Representa la forma de cada uno de los conjuntos difusos asociados a los valores lingüísticos (parámetros de la función de pertenencia):
 $\{\text{"Bajo"}: [0.264; 0.106]; \text{"Moderado"}: [0.242; 0.51]; \text{"Alto"}: [0.199; 1]\}$

b) Variable de Salida del Sistema

1. GAG : { GAG; T(GAG);U; G; M}

En Donde:

- *Grado de Alineamiento del Grupo (GAG)*: Se refiere a una variable que mide el grado de alineamiento existente entre el EPPIRU propuesto por un determinado Grupo y el EGPIRU.
- *T(GAG)*: Representa el conjunto de valores lingüísticos:
{“Bajo”; “Moderado”; “Aceptable”}
- *U* : Se refiere al Universo del Discurso:
[0..1]
- *G*: Especifica la forma que se genera los valores en *T(GAG)*:
{*gaussiana*; *gaussiana*; *gaussiana*}
- *M*: Representa la forma de cada uno de los conjuntos difusos asociados a los valores lingüísticos (parámetros de la función de pertenencia):

{“Bajo”: [0.1853; 0.0159]; “Moderado”: [0.221; 0.4868]; “Aceptable”: [0.161; 1]}

c) Base de Reglas

El sistema de la memoria asociativa difusa (FAM), es una técnica que permite construir las reglas de decisión, a partir de la interacción que surge entre los términos lingüísticos de las variables de entrada y el resultado lingüístico previsto para la variable de salida del sistema. Luego, cada regla de decisión se asemeja a la opinión de los expertos en relación al resultado de dicha interacción [Kosco (1991)].

Para este caso en particular, la FAM queda configurada a partir de una serie de reglas lingüísticas expresadas como sentencias condicionales del tipo: SI-ENTONCES (IF-THEN). Ya que, el sistema se compone de dos variables lingüísticas con tres etiquetas cada una, se definirán 9 reglas (3^2) en función de las siguientes consideraciones:

	Grado de Coincidencia del Grupo (GCG)		
Nivel de Preferencia del Grupo (NPG)	Bajo	Moderado	Alto
Sub-Valorado	Bajo	Bajo	Moderado
Igualmente Valorado	Bajo	Moderado	Aceptable
Sobre-Valorado	Moderado	Aceptable	Aceptable

Tabla 12: "Grado de Alineamiento del Grupo (GAG) en función del comportamiento de las variables de entradas".

Fuente: Elaboración Propia.

Como puede apreciarse en la Tabla 12, se podrá constatar el Grado de Alineamiento del Grupo (GAG) en función del comportamiento que adquiera su Nivel de Preferencia (NPG) y su Grado de Coincidencia (GCG) durante el proceso de comparación de su respectivo EPPIRU con el EGPIRU.

d) Motor de Inferencia

Como se ha mencionado en el Capítulo 3 (Sección 3.6.4.3), el motor de inferencia difuso es el mecanismo que permite obtener la salida de un sistema difuso en función de sus reglas y las entradas que le sean aplicadas. Para este contexto en particular, se considerará lo siguiente:

- Se calculará el grado de cumplimiento de cada antecedente en función de la entrada del sistema. Para ello, se utiliza un operador de intersección (T-norma) sobre dichos conjuntos.
- Una vez calculado los antecedentes de cada regla, y considerando que las mismas se componen lingüísticamente por la condición "Y", se utilizará el operador de intersección T-norma, a los efectos de determinar el grado de activación de la misma.
- Se utiliza como operador de implicación (Entonces) el modelo de Mamdani (mínimo).
- Para agregar consecuentes y obtener una función de pertenencia en la variable salida, se utiliza una T-conorma.

- Puesto que la salida del motor de inferencia hace referencia a un conjunto difuso, resulta necesario convertirlo en un valor numérico concreto. A tal efecto, se aplicará el método del centroide a la variable de salida (Sección 3.6.4.4. Capítulo 3).

e) Análisis de los Resultados del Sistema

Dada la especificación previa, el sistema de la memoria asociativa difusa (FAM) queda determinado por 2 variables de entradas (NPG, GCG), una de salida (GAG defuzzifier), un total de 9 reglas lingüísticas y un Motor de Inferencia. A partir de la interacción de los componentes señalados, se obtiene la superficie decisión, tal cual puede apreciarse en la Figura 4.2.

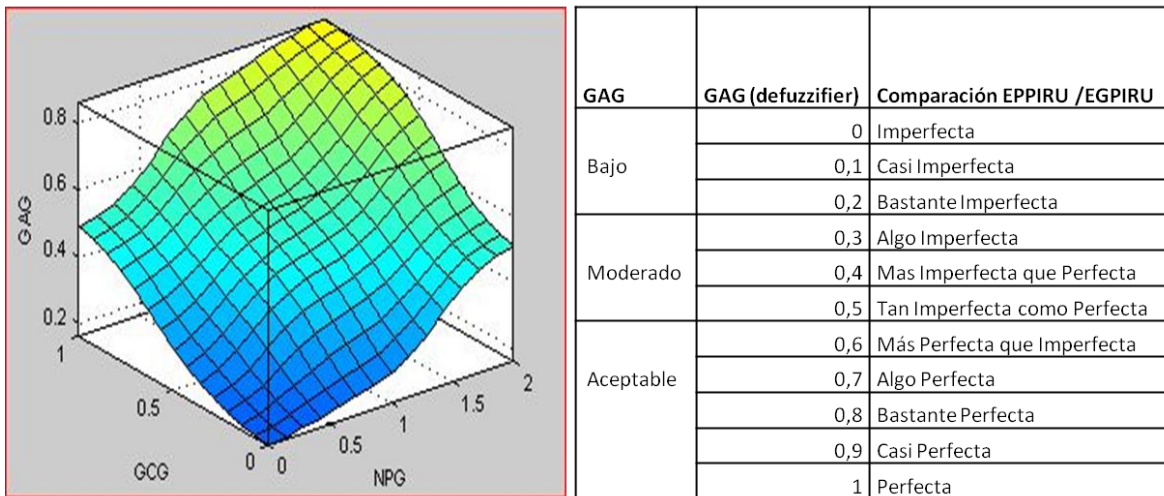


Figura 4.2 “Posibles grados de alineamiento del grupo tras la comparación entre su EPPIRU y el EGPIRU”. Fuente: Elaboración Propia

Asignando un determinado par de valores al conjunto de variables de entradas del Sistema (NPG y GCG), será factible conocer el Grado de Alineamiento del Grupo (GAG) en torno a la comparación entre su EPPIRU y el EGPIRU. A partir de dicho resultado, será factible determinar si las preferencias manifestadas por el grupo (EPPIRU) necesitan ser modificadas o no a través de un proceso de negociación focalizado (Sección 3.5). Al efecto, considérese:

“Aplicar un Proceso de Negociación a todo grupo de stakeholders cuyo grado de alineamiento de su EPPIRU sea Bajo ó Moderado en comparación con el EGPIRU”

En términos de la salida del sistema de la memoria asociativa difusa (FAM), debería ser aplicado un Proceso de Negociación, sí un determinado grupo de *stakeholders*, obtiene un valor de GAG (*defuzzifier*) menor o igual 0.5. Es decir, el resultado de la comparación entre su EPIRU y el EGPIRU debería ser menor o igual a “*Tan Imperfecto como Perfecto*” (ver Tabla en Figura 4.2). De esta manera, aquellos grupos que satisfagan la condición anterior, estarán sujetos al Proceso de Negociación consignado. Asimismo, su aplicación tendrá como objetivo básico lograr el mayor consenso posible entre aquellos *stakeholders*, cuyos niveles de preferencia, disten de forma significativa de los constituidos por el universo de ellos.

f) Tabulación de Resultados

En base a los datos obtenidos durante las diferentes etapas del sistema de la memoria asociativa difusa (FAM), el Ingeniero en Requerimientos confeccionará la Lista de Grupos Clasificados según Categoría de Alineamiento (LGCCA). A tal efecto, se recomienda seguir el siguiente formato (Tabla 13):

LGCCA						
Grupo	Orden de Corte	NPG	GCG	GAG(<i>defuzzifier</i>)	Categoría Alineamiento (GAG)	Resultado de la Comparación (RC)
1	OC ₁	NPG ₁	GCG ₁	GAG(d) ₁	GAG ₁	RC ₁
2	OC ₂	NPG ₂	GCG ₂	GAG(d) ₂	GAG ₂	RC ₂
...	OC...	NPG...	GCG...	GAG(d)...	GAG...	RC...
N	OC _N	NPG _N	GCG _N	GAG(d) _N	GAG _N	RC _N

Tabla 13: “Ejemplo de Lista de Grupos Clasificados según Categoría de Alineamiento (LGCCA)”. Fuente: Elaboración Propia

4.10 Consideraciones Finales del Capitulo

El presente Capitulo ha desarrollado cada una de las instrucciones del **Modelo de Priorización de Requerimientos de Usuarios a partir de la Segmentación de Preferencias de los Stakeholders (MPRUSPS)**. A partir del cumplimiento de cada uno de los supuestos consignados en la Introducción, será factible recorrer cada una de las instrucciones del modelo, esperando en consecuencia, establecer un Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU), consensuado y validado por el universo de *stakeholders* de un Proyecto de Software dado.

Capítulo 5

Experimentación: Ejemplo de aplicación del Modelo de Priorización de Requerimientos de Usuarios a partir de la Segmentación de Preferencias de los Stakeholders (MPRUSPS)

5.1 Introducción

El Modelo de Priorización de Requerimientos de Usuarios a partir de la Segmentación de las Preferencias de los *Stakeholders* (MPRUSPS), es un enfoque que permite integrar en un mismo proceso, los aspectos socio-cognitivos y de negociación que subyacen tras la asignación de un orden de prioridad entre los requerimientos disponibles. Dado que el presente implica un enfoque de tipo experimental, constatar su eficacia y demostrar su conveniencia, demandará sin lugar a dudas, contrastar la propuesta en diversos escenarios y dominios.

Si bien, la presente investigación prevé la aplicación del Modelo en un caso real, el presente Capítulo, propone una experimentación extra del modelo a través de un caso simulado. De esta manera se espera satisfacer, al menos en parte, lo requerido por un proceso de investigación, cuyo objeto básico, sea validar un modelo de carácter experimental.

5.2 Descripción general del contexto de aplicación del MPRUSPS

Dada su naturaleza, el Modelo de priorización propuesto debería ser aplicado en aquellos casos en donde los principales *stakeholders* del Proyecto de Software, puedan ser entrevistados personalmente a los efectos de elicitación de sus respectivas producciones semánticas en torno a los conceptos claves del dominio. De esta manera, y considerando adicionalmente el cumplimiento de los supuestos del Modelo, será factible aplicar el enfoque en aquellos escenarios en donde exista una multiplicidad de requerimientos de usuarios a evaluar, a través del juicio valorativo de un conjunto finito y heterogéneo de *stakeholders* (es decir, con diferentes intereses, experticia, conocimiento del dominio, entre otras cualidades).

Por tanto, se presenta el caso de un **Proyecto de Software** que dispone de unos **50 Requerimientos de Usuarios**. En primera instancia, el problema consistirá en ordenar los requerimientos disponibles, de acuerdo a las preferencias manifestadas por un conjunto de **10 stakeholders**. Adicionalmente, se espera que dicho esquema de ordenamiento, sea **representativo** del juicio valorativo de los involucrados. Eventualmente, sería deseable constatar la existencia de aquellos grupos de *stakeholders*, cuyo nivel de preferencia, diste de manera significativa de aquel especificado a nivel global (total de *stakeholders*). En caso de cumplirse esta última contingencia, sería recomendable la implementación de un proceso de negociación focal a los efectos de elicitación, fundamentar y consensuar las diferencias obtenidas. Finalmente, se supone que:

1. Los *10 stakeholders* residen en el mismo espacio geográfico en donde el Proyecto tendrá lugar.
2. Por aplicación de la Etapa 4.2 del Modelo (Sección 4.2. Capítulo 4), los *stakeholders establecen la existencia de k conceptos*, que según su percepción, resultarían representativos a los efectos de comprender los usos e implicaciones de la futura herramienta de software. Luego, la producción semántica obtenida por el colectivo social en referencia a dichos conceptos, se establece de la siguiente manera:

$$VTV = VTV_{i,k}$$

$$RE = RE_{j,k}$$

3. Los *stakeholders* constatan diferentes niveles de conocimiento en torno al dominio de la futura aplicación. Cada nivel de conocimiento, se fundamenta principalmente por el rol que desempeña cada uno de los *stakeholders* en relación al Proyecto, sus intereses, experticia, entre otros aspectos. Asimismo, dicha circunstancia

promueve juicios de preferencia diferenciales en torno a los requerimientos a evaluar. Luego:

- Por aplicación del supuesto establecido en la Sección 3.3 (Capítulo 3) cada juicio de preferencia será funcional a la configuración que adquiera la red semántica del sujeto en cuestión, en torno al conjunto de conceptos que se consideran claves a los efectos de comprender los usos e implicancias de la futura herramienta de software (paso 2).
4. Cada requerimiento de usuario ha sido correctamente especificado, es independiente del resto, y además, puede valorarse en forma autónoma (Supuesto 1 y 2 del Modelo).
 5. Cada juicio de preferencia, tiene la misma ponderación en el esquema de agregación de las preferencias (Supuesto 3 del Modelo).

5.3 Relevamiento de las preferencias de implementación de los Requerimientos de Usuarios.

A los efectos de satisfacer lo requerido por la Sección 4.5 del Capítulo 4, se ha procedido a simular la función de distribución de preferencia de cada uno de los *stakeholders* del Proyecto. Dicha simulación, ha sido efectuada utilizando el método de Montecarlo.

Por su parte, las funciones de distribución de preferencias provienen de la aplicación del Método AHP (Sección 3.2.1. Capítulo 3). Adicionalmente, se considera que cada uno de los *stakeholders* ha logrado un nivel de inconsistencia (CR) menor al 10%. Con lo cual, cada vector de distribución de preferencia se considera válido al efecto. Luego de lo consignado, se muestra en la Tabla 14, la Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios (LFDPIRU) para este Proyecto en particular:

Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios (LFDPIRU)

Requerimientos/Stakeholders	1	2	3	4	5	6	7	8	9	10
RU1	0,00234	0,00234	0,01231	0,05905	0,00473	0,00436	0,01984	0,01138	0,03231	0,01231
RU2	0,00023	0,00023	0,02342	0,02001	0,01485	0,00357	0,02359	0,02864	0,02342	0,01342
RU3	0,00025	0,00025	0,01574	0,01255	0,02453	0,00312	0,03856	0,02969	0,01574	0,01574
RU4	0,00123	0,00123	0,01302	0,0573	0,03484	0,00288	0,04879	0,02897	0,01302	0,01302
RU5	0,00123	0,00123	0,00436	0,06789	0,04185	0,02789	0,05768	0,00735	0,00436	0,00357
RU6	0,00238	0,00238	0,00357	0,05895	0,05679	0,03895	0,02239	0,00712	0,00357	0,00987
RU7	0,00473	0,00473	0,00312	0,01235	0,05789	0,01235	0,03231	0,00677	0,00312	0,00876
RU8	0,01485	0,01485	0,00288	0,01115	0,05905	0,01115	0,02342	0,00567	0,00288	0,00858
RU9	0,02453	0,02453	0,00213	0,00986	0,02001	0,00986	0,01574	0,00436	0,00213	0,00257
RU10	0,03484	0,03484	0,03235	0,01349	0,01255	0,01349	0,01302	0,00357	0,01235	0,03235
RU11	0,04185	0,00677	0,01359	0,01138	0,0573	0,01138	0,00436	0,00312	0,01346	0,02245
RU12	0,05679	0,00567	0,00567	0,07864	0,06789	0,01864	0,00357	0,00288	0,01674	0,01347
RU13	0,05789	0,00436	0,00436	0,08969	0,07895	0,01969	0,00312	0,00213	0,00198	0,01346
RU14	0,05905	0,00357	0,00357	0,05897	0,01235	0,00357	0,00288	0,00188	0,00198	0,00988
RU15	0,02001	0,00312	0,00312	0,00735	0,01115	0,00312	0,00213	0,00157	0,00136	0,00987
RU16	0,01255	0,00288	0,00288	0,00712	0,00986	0,00288	0,03235	0,00093	0,01984	0,00577
RU17	0,0573	0,00213	0,02342	0,00677	0,01349	0,00213	0,07245	0,01345	0,01359	0,00875
RU18	0,06789	0,00188	0,01574	0,00567	0,01138	0,00188	0,08674	0,01984	0,00186	0,00765
RU19	0,07895	0,00157	0,01302	0,00436	0,00986	0,00157	0,07976	0,01359	0,01279	0,00768
RU20	0,01235	0,00093	0,02477	0,00357	0,00897	0,00093	0,01984	0,01856	0,00259	0,00099
RU21	0,01115	0,01345	0,029	0,00312	0,0079	0,01345	0,01359	0,03879	0,02135	0,01345
RU22	0,00986	0,01984	0,01457	0,00288	0,00735	0,01984	0,01984	0,04357	0,01879	0,01984
RU23	0,01349	0,02359	0,03123	0,00213	0,01359	0,02359	0,01359	0,03567	0,03568	0,02359
RU24	0,01138	0,03856	0,0457	0,00188	0,01856	0,03856	0,01856	0,02123	0,02574	0,03856
RU25	0,00986	0,04879	0,03235	0,00157	0,01279	0,04879	0,01279	0,02347	0,02302	0,03879
RU26	0,00897	0,05768	0,05245	0,00093	0,00259	0,05768	0,00259	0,01236	0,02477	0,04768
RU27	0,0079	0,04239	0,06674	0,01345	0,00213	0,02239	0,00213	0,03652	0,039	0,03239
RU28	0,00735	0,03231	0,00677	0,01984	0,00188	0,03231	0,00188	0,04565	0,03457	0,03231
RU29	0,00712	0,02342	0,00567	0,02359	0,01568	0,02342	0,01568	0,02342	0,02123	0,02342
RU30	0,00677	0,01574	0,00436	0,03856	0,00234	0,02574	0,00234	0,02375	0,0457	0,01574
RU31	0,00567	0,01302	0,03567	0,03879	0,00023	0,02346	0,00023	0,00473	0,03235	0,03453
RU32	0,00436	0,02477	0,02123	0,00436	0,00025	0,04357	0,00025	0,02485	0,05245	0,03436
RU33	0,00357	0,029	0,03876	0,00357	0,00123	0,03867	0,00123	0,03453	0,04674	0,03876
RU34	0,00312	0,03457	0,04135	0,00312	0,00123	0,03223	0,00123	0,03484	0,03976	0,03112
RU35	0,00288	0,04123	0,03879	0,00288	0,00238	0,00288	0,00238	0,04185	0,03666	0,03673
RU36	0,00213	0,0457	0,02568	0,00213	0,00473	0,02235	0,00473	0,05679	0,03567	0,03891
RU37	0,00188	0,06235	0,02943	0,00188	0,01485	0,03435	0,01485	0,05789	0,05635	0,05235
RU38	0,00157	0,07245	0,03345	0,00157	0,02453	0,07895	0,02453	0,05905	0,02876	0,05682
RU39	0,00093	0,08674	0,07234	0,00093	0,03484	0,06874	0,03484	0,02001	0,02935	0,02239
RU40	0,01345	0,07976	0,05359	0,01345	0,00677	0,04296	0,00677	0,01355	0,0321	0,03976
RU41	0,01984	0,02666	0,01856	0,01984	0,00567	0,01484	0,00567	0,0273	0,03346	0,01984
RU42	0,02359	0,01335	0,01239	0,01359	0,03856	0,01359	0,03856	0,02346	0,02674	0,01359
RU43	0,03856	0,01121	0,01868	0,01856	0,02879	0,01984	0,02879	0,02357	0,02399	0,01984
RU44	0,04879	0,00783	0,02139	0,03879	0,03768	0,01359	0,03768	0,01235	0,01984	0,01359
RU45	0,05768	0,00497	0,0014	0,02768	0,03239	0,01856	0,03239	0,03567	0,00487	0,01556
RU46	0,04239	0,00346	0,00246	0,02239	0,02011	0,01279	0,02211	0,00312	0,00366	0,01799
RU47	0,03231	0,00259	0,00137	0,03231	0,01649	0,00259	0,00168	0,00288	0,00259	0,00299
RU48	0,02342	0,00281	0,00152	0,02242	0,01338	0,00213	0,01368	0,00213	0,00223	0,00293
RU49	0,01574	0,00168	0,00226	0,01554	0,01174	0,00188	0,01174	0,00198	0,00178	0,00179
RU50	0,01302	0,001	0,01792	0,01212	0,01109	0,01568	0,01109	0,00386	0,00112	0,00115
Total	1	1	1	1	1	1	1	1	1	1

Tabla 14: “Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios (LFDPIRU). Fuente Elaboración Propia.

5.4 Segmentación (agrupamiento) de las Preferencias de los stakeholders.

Una vez obtenida la Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios (LFDPIRU etapa anterior), el objetivo siguiente consistirá en segmentar (agrupar) a los *stakeholders*, de acuerdo al grado de similitud existente entre sus respectivos vectores de preferencias de implementación de requerimientos de usuarios. A tal efecto, se utiliza el método: Mapas Auto-Organizados de Kohonen (Capítulo 3. Sección: 3.4.2), según lo consignado por la Sección 4.6. Al efecto, se aplicará la versión 7.8.0 (R2009a) del software **Matlab** junto a la librería “*SOM Toolbox*” desarrollada por Alhoiemi (et. al., 2005) bajo licencia GNU (*General Public License*).

De esta manera, se obtiene:

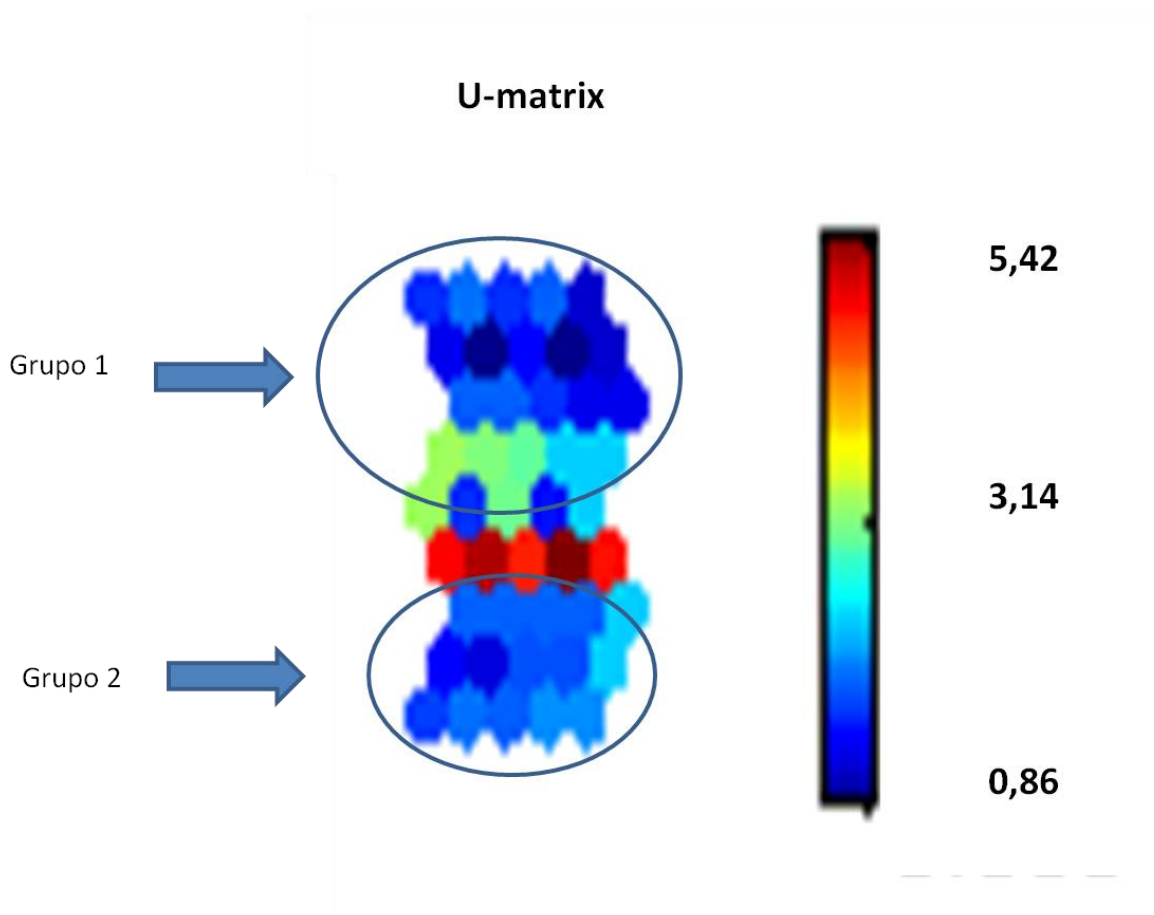


Figura 5.1 “Matriz unificada de distancia para el Caso propuesto en 5.3”. Fuente: Elaboración Propia

Partiendo de la disponibilidad de 10 vectores de preferencia (10 *stakeholders*) en torno al orden de implementación de los 50 Requerimientos de Usuarios del Proyecto, se procede a iniciar un proceso de agrupamiento de las preferencias manifestadas, en función del grado de similitud experimentado entre ellas. Como se puede apreciar en la Figura 5.1, la matriz de distancia unificada, advierte en primera instancia, la configuración de 2 grupos de *stakeholders* bien diferenciados. El primero de ellos (Grupo 1) se ubicaría en la sección superior de la grilla, mientras que, el segundo, se localizaría en la sección inferior de la misma (Grupo 2).

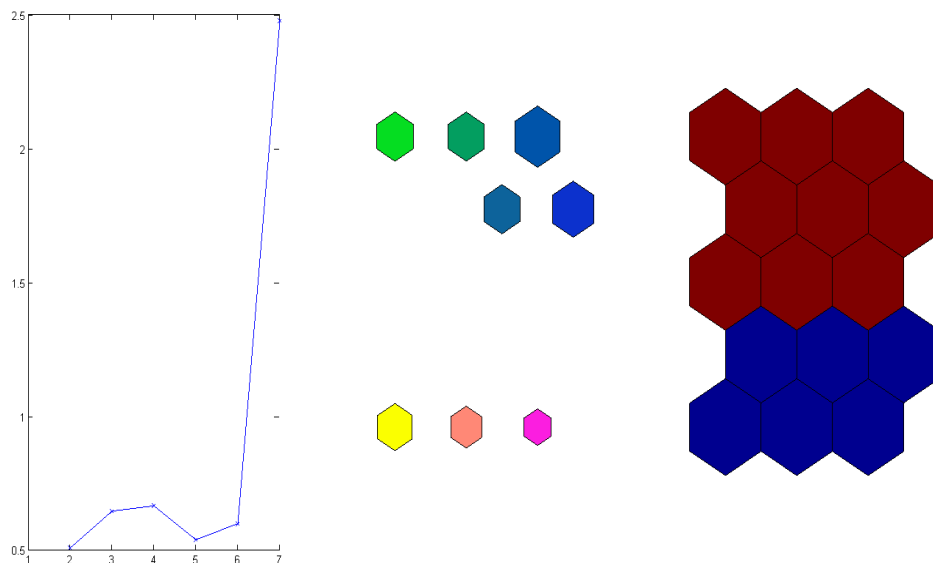


Figura 5.2 "Índice de Davies-Bouldin para el Caso propuesto en 5.3". Fuente: Elaboración Propia

En la Figura 5.2, claramente puede confirmarse la existencia de 2 grupos. En el gráfico de la izquierda de la Figura, puede determinarse que el valor que minimiza el índice de Davies-Bouldin asciende a 0,5. Luego, dicha cifra resulta adecuada. En los gráficos del centro y la derecha, se propone una configuración de grupos tal que la grilla queda particionada en los sectores: **Superior e Inferior**. En efecto, dicha configuración también ha sido divisada en la Matriz de distancia Unificada de la Figura 5.1.

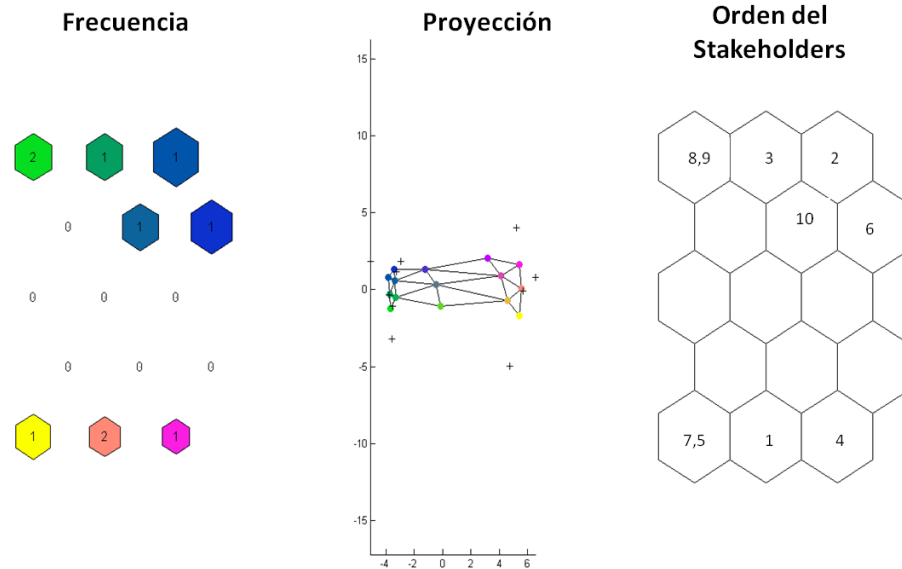


Figura 5.3 "Composición de los Grupos". Fuente: Elaboración Propia , en base a resultados Matlab.

En la Figura 5.3, se pueden apreciar los gráficos "Frecuencia", "Proyección" y "Orden de Stakeholder". En el gráfico "Frecuencia" se está indicando la cantidad de *stakeholders* que se clasifican en cada una de las celdas, en función de su patrón de preferencia de prioridades. De esta manera, se puede constatar que el Grupo 1 se compone de 6 *stakeholders*, mientras que, en el Grupo 2, residen 4 de ellos. Luego, el gráfico "Orden del Stakeholder", establece como se distribuyen los *stakeholders* entre los grupos hallados. Así, los *stakeholders*: 2, 3, 6, 8, 9 y 10 se localizarían en el Grupo 1, mientras que, los de orden: 1, 4, 5 y 7 se hallarían en el Grupo 2 (ver correspondencia de orden en Tabla 14) Por último, el gráfico "Proyección" confirma nuevamente la existencia de ambos grupos (el Grupo 1 con una proyección sesgada a la izquierda en el plano bidimensional, y el grupo 2 con un sesgo hacia la derecha).

Grupo	Preferencia Acumulada en Tramos de Requerimientos de Usuarios (%)					
	Tramos →	1 a 10	11 a 20	21 a 30	31 a 40	41 a 50
1		11,5	7,8	29,8	39,8	11,1
2		25,8	33,2	9,7	7,3	24

Tabla 15: "Preferencia Acumulada en tramos de Requerimientos". Fuente elaboración Propia.

En base a los análisis anteriores, se ha procedido a evaluar el comportamiento que ha adquirido la preferencia acumulada por tramos de requerimientos, en cada uno de los grupos consignados. Como se puede apreciar en la Tabla 15, el Grupo 1 ha valorado con mayor intensidad promedio, los Requerimientos de Usuarios comprendidos en el tramo: 31-40, mientras que, el Grupo 2 lo ha establecido en el tramo 11-20. Por su parte, el Grupo 1 ha valorado con menor intensidad promedio el tramo de requerimientos comprendidos en el tramo 11-20, mientras que, el Grupo 2 lo ha determinado para el tramo 31-40.

5.5 Identificación de Esquemas Particulares de Prioridad Implementable de Requerimientos de Usuarios (EPPIRU)

Una vez efectuado el proceso de segmentación de preferencias de los *stakeholders* (etapa anterior), se establecerá el valor medio de preferencia de cada uno de los Requerimientos Usuarios valorados en cada uno de los Grupos identificados. Para ello, el Ingeniero en Requerimientos deberá seguir las recomendaciones establecidas en la Sección 4.7. Acto seguido y de acuerdo a lo consignado en la misma etapa, se procederá a ordenar dichos valores de mayor a menor, al efecto de hallar el EPPIRU de cada uno de los Grupos.

Orden	EPPIRU(G1)	Fr.Acu. (%)	EPPIRU(G2)	Fr.Acu. (%)
1	38	5,49	13	5,74
2	39	10,48	12	10,91
3	37	15,36	19	15,24
4	40	19,72	18	19,53
5	26	23,94	5	23,75
6	27	27,93	44	27,82
7	33	31,70	45	31,57
8	36	35,45	17	35,32
9	25	39,04	4	38,88
10	34	42,60	6	42,39
11	24	46,08	14	45,72
12	32	49,43	11	48,59
13	35	52,73	43	51,46
...
40	13	95,22	25	95,18
41	9	95,98	21	96,08
42	46	96,71	37	96,91
43	50	97,39	28	97,69
44	7	98,03	27	98,33
45	16	98,62	26	98,71
46	14	99,03	36	99,05
47	15	99,40	35	99,31
48	47	99,65	33	99,55
49	48	99,88	32	99,78
50	49	100	34	100

Tabla 16 "EPPIRU asignado por Grupo". Fuente: Elaboración Propia.

En la Tabla 16, se muestra un fragmento de los respectivos esquemas de ordenación propuesto por cada uno de los grupos [EPPIRU (G1) y EPPIRU (G2)]. Como se puede apreciar, los 13 primeros requerimientos de usuarios capitalizan el 52,73 % de las preferencias asignadas por el Grupo 1, y el 51,46% en el Grupo 2. Luego, los últimos 10 Requerimientos de Usuarios (Tramo 40-50) adquieren una escasa significación para ambos grupos, puesto que, las preferencias acumuladas alcanzan el 4,78 % para el Grupo 1 (100 -95,22) y el 4,82 % para el Grupo 2 (100-95,18).

5.6 Determinación del Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU)

Con el objeto de calcular el Valor Medio de Prioridad Implementable de los Requerimientos de Usuarios para el total de Grupos, resulta esencial determinar el desempeño socio-cognitivo que ha tenido cada uno de los Grupos implicados. Para ello, se procede a considerar dos escenarios factibles de producciones semánticas en torno a los conceptos claves del dominio.

SPSCCDG				
Grupo	Escenario 1		Escenario 2	
	$\overline{VT\bar{V}}$	\overline{RE}	$\overline{VT\bar{V}}$	\overline{RE}
1	5	0,30	2	0,30
2	2	0,65	5	0,65

Tabla 17: " Síntesis de las Producciones Semánticas de los conceptos claves del dominio por Grupo (SPSCCD) y en diferentes escenarios". Fuente: Elaboración Propia.

Como se puede apreciar en la Tabla 17, se han formulado diferentes escenarios de desempeño socio-cognitivo en los Grupos identificados. En el primer Escenario, al Grupo 1 se le asigna un nivel promedio de $\overline{VT\bar{V}}$ (para los k conceptos señalados en 5.2.2) de 5 unidades. Es decir que el Grupo 1 ha logrado obtener un nivel promedio del campo semántico de los conceptos claves del dominio, que resulta 2,5 veces superior que el alcanzado por el Grupo 2 ($\overline{VT\bar{V}}= 2$). En este sentido, el Grupo 1 ha obtenido un mayor consenso en la identificación de las definidoras de los conceptos claves del dominio en términos relativos. Por su parte, y considerando la relación directa, aunque no lineal, entre el valor de $\overline{VT\bar{V}}$ y \overline{RE} , se podrá indicar que el grado de identificadoras excluidas en el Grupo 2 supera en proporción a las obtenidas por el Grupo 1. Luego, el Escenario 2 se propone como análogo al anterior, sin embargo, se ha invertido la asignación de valores en las variables mencionadas.

A continuación, y considerando los valores promedio de $\overline{VT\bar{V}}$ en cada uno de los Grupos, se procederá al cálculo de los Valores Medio de Prioridad Implementable de Requerimientos de Usuarios en los diferentes escenarios planteados. Acto seguido, y considerando el resto de las recomendaciones efectuadas en la Sección 4.8 del Capítulo 4, se procederá a determinar el EGPIRU también en cada escenario.

5.6.1 Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU) según el primer Escenario.

Considerando el caso en el que el Grupo 1 haya obtenido un desempeño socio-cognitivo significativamente mayor que el Grupo 2, el EGPIRU resultante, sería:

EGPIRU(E1)	Fr.Acu. (%)	EGPIRU(E1)	Fr.Acu. (%)	EPPIRU(G1)	Fr.Acu. (%)	EPPIRU(G1)	Fr.Acu. (%)	EPPIRU(G2)	Fr.Acu. (%)	EPPIRU(G2)	Fr.Acu. (%)
38	4,30	22	68,03	38	5,49	44	80,89	13	5,74	16	78,81
39	8,37	29	69,91	39	10,48	45	82,24	12	10,91	2	80,28
37	12,10	4	71,79	37	15,36	3	83,58	19	15,24	49	81,65
40	15,50	19	73,62	40	19,72	1	84,83	18	19,53	38	82,95
26	18,62	17	75,45	26	23,94	4	86,03	5	23,75	41	84,23
27	21,65	18	77,25	27	27,93	11	87,21	44	27,82	24	85,49
24	24,49	21	79,05	33	31,70	6	88,31	45	31,57	30	86,74
25	27,32	5	80,84	36	35,45	17	89,36	17	35,32	50	87,92
36	30,09	6	82,62	25	39,04	12	90,41	4	38,88	31	89,05
33	32,86	11	84,28	34	42,60	19	91,25	6	42,39	20	90,16
34	35,47	2	85,80	24	46,08	18	92,07	14	45,72	23	91,23
32	37,93	1	87,31	32	49,43	20	92,88	11	48,59	15	92,25
35	40,36	3	88,81	35	52,73	5	93,69	43	51,46	40	93,26
28	42,77	8	90,13	28	55,80	8	94,46	42	54,32	22	94,26
23	45,14	46	91,41	23	58,69	13	95,22	8	57,03	25	95,18
12	47,37	14	92,66	31	61,08	9	95,98	7	59,71	21	96,08
44	49,59	7	93,89	41	63,43	46	96,71	46	62,39	37	96,91
43	51,80	9	94,93	22	65,70	50	97,39	1	64,54	28	97,69
13	53,99	20	95,83	30	67,88	7	98,03	47	66,61	27	98,33
10	56,05	16	96,69	21	70,04	16	98,62	3	68,50	26	98,71
42	58,10	50	97,51	10	72,19	14	99,03	10	70,35	36	99,05
41	60,14	47	98,28	29	74,20	15	99,40	48	72,17	35	99,31
45	62,17	48	98,97	43	76,15	47	99,65	39	73,96	33	99,55
31	64,21	15	99,52	42	77,87	48	99,88	9	75,71	32	99,78
30	66,12	49	100	2	79,42	49	100	29	77,27	34	100

Tabla 18 “EGPIRU resultante en el Escenario 1”. Fuente: Elaboración Propia.

Como puede apreciarse en la Tabla 18, aparte del ordenamiento propuesto por el EGPIRU (Escenario 1), se muestran los relativos a los Grupos 1 y 2 [EPPIRU (G1) y EPPIRU (G2)]. Asimismo, puede constarse que los 18 primeros Requerimientos de Usuarios propuestos por el EGPIRU (desde el 38 al 43), alcanzan al menos el 50% superior de las preferencias agregadas (51,80%). Luego, el **Orden de Corte** queda establecido en este nivel (Sección 4.9 del Capítulo 4). Por su parte, el Grupo 1 asigna un 65,70 % a sus primeros 18 Requerimientos de Usuarios, mientras que, el Grupo 2, le concede un 64,54% de su preferencia.

Si se compara el grado de coincidencia obtenido entre el orden propuesto por el EGPIRU y los respectivos EPPIRU (G1 y G2), en estos primeros 18 Requerimientos de Usuario, se podrán apreciar algunas diferencias significativas. En primer lugar, y dado que el presente escenario estipula un desempeño socio-cognitivo mayor en el Grupo 1, tras

aplicar la Ecuación IXXX (Sección 4.8 del Capítulo 4), se podrá constatar que el EGIPIRU se encuentra más sesgado al orden propuesto por el Grupo 1 [EPPIRU (G1)], que por el Grupo 2 [EPPIRU(G2)]. En efecto, y según lo establecido por la Sección 4.9 (Capítulo 4), el Grupo 1 propone un orden exacto (Coincidencia Exacta) en 10 de los 18 Requerimientos de Usuarios propuesto por el EGIPIRU (38, 39, 37, 40, 26, 27, 32, 35, 28, 23), un orden relativamente exacto (Coincidencia Relativamente Exacta) en 5 de ellos (33, 36, 25, 34, 24), mientras que, tan sólo 3 Requerimientos (31, 41, 22) no tienen correlato en la ordenación superior propuesta por el EGIPIRU (No Coincidencia).

Por su parte, el Grupo 2 establece un orden de Requerimientos que resulta significativamente diferente al propuesto por el EGIPIRU. En efecto, tan sólo 3 de sus primeros 18 Requerimientos de Usuarios (12, 44, 43) obtienen un orden relativamente exacto, mientras que, los restantes 15 requerimientos, no adquieren correlato alguno en la ordenación superior propuesta por el EGIPIRU (No Coincidencia).

5.6.2 Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU) según el segundo Escenario.

En forma análoga al análisis propuesto en 5.6.1, se presenta el caso en el que el Grupo 2 obtiene un desempeño socio-cognitivo significativamente mayor que el Grupo 1 (Escenario 2). Luego, el EGIPIRU resultante sería:

EGPIRU(E2)	Fr.Acu. (%)	EGPIRU(E2)	Fr.Acu. (%)	EPPIRU(G1)	Fr.Acu. (%)	EPPIRU(G1)	Fr.Acu. (%)	EPPIRU(G2)	Fr.Acu. (%)	EPPIRU(G2)	Fr.Acu. (%)
13	4,32	25	67,97	38	5,49	44	80,89	13	5,74	16	78,81
12	8,31	29	69,65	39	10,48	45	82,24	12	10,91	2	80,28
44	11,65	27	71,25	37	15,36	3	83,58	19	15,24	49	81,65
19	14,97	23	72,84	40	19,72	1	84,83	18	19,53	38	82,95
18	18,27	41	74,42	26	23,94	4	86,03	5	23,75	41	84,23
5	21,52	47	75,97	27	27,93	11	87,21	44	27,82	24	85,49
45	24,58	30	77,49	33	31,70	6	88,31	45	31,57	30	86,74
17	27,56	2	78,98	36	35,45	17	89,36	17	35,32	50	87,92
4	30,45	31	80,46	25	39,04	12	90,41	4	38,88	31	89,05
6	33,27	26	81,93	34	42,60	19	91,25	6	42,39	20	90,16
39	35,97	9	83,40	24	46,08	18	92,07	14	45,72	23	91,23
43	38,58	28	84,83	32	49,43	20	92,88	11	48,59	15	92,25
42	41,11	48	86,20	35	52,73	5	93,69	43	51,46	40	93,26
38	43,61	22	87,56	28	55,80	8	94,46	42	54,32	22	94,26
14	46,11	36	88,88	23	58,69	13	95,22	8	57,03	25	95,18
11	48,49	16	90,15	31	61,08	9	95,98	7	59,71	21	96,08
8	50,65	21	91,41	41	63,43	46	96,71	46	62,39	37	96,91
46	52,77	33	92,66	22	65,70	50	97,39	1	64,54	28	97,69
7	54,87	34	93,83	30	67,88	7	98,03	47	66,61	27	98,33
37	56,86	35	94,96	21	70,04	16	98,62	3	68,50	26	98,71
40	58,83	32	96,09	10	72,19	14	99,03	10	70,35	36	99,05
10	60,76	50	97,12	29	74,20	15	99,40	48	72,17	35	99,31
1	62,65	49	98,16	43	76,15	47	99,65	39	73,96	33	99,55
24	64,55	20	99,19	42	77,87	48	99,88	9	75,71	32	99,78
3	66,28	15	100	2	79,42	49	100	29	77,27	34	100

Tabla 19 “EGPIRU resultante en el Escenario 2”. Fuente: Elaboración Propia.

Como puede apreciarse en la Tabla 19, el Escenario 2 promueve que el EGPIRU (E2) se encuentre más alineado al orden propuesto por el Grupo 2. A diferencia del escenario anterior (Sección 5.6.1), la cantidad de requerimientos de usuarios que totalizarían al menos el 50% superior de las preferencias agregadas (**Orden de Corte**), alcanzarían los 17 (50,65% en EGPIRU). Luego, los Grupos 1 y 2 acumularían un nivel de preferencia del orden del 63,43% y el 62,39% respectivamente, en torno a los primeros 17 Requerimientos de Usuarios propuestos en cada uno de ellos.

Respecto al grado de coincidencia (Sección 4.9 Capítulo 4) entre el orden propuesto por el EGPIRU y los respectivos EPPIRU (G1 y G2), en estos primeros 17 Requerimientos de Usuario, se podrán apreciar algunas diferencias significativas. En efecto, el Grupo 2 propone un orden exacto en 6 de los 17 Requerimientos de Usuarios propuesto por el EGPIRU (13, 12, 45, 17, 4, 6), un orden relativamente exacto en 9 de ellos (19, 18, 5, 44, 14, 11, 43, 42, 8), mientras que, tan sólo 2 Requerimientos (7, 46) no tienen correlato en la ordenación superior propuesta por el EGPIRU.

Por su parte, el Grupo 1 establece un orden de Requerimientos que resulta significativamente diferente al propuesto por el EGPIRU. En efecto, tan sólo 2 de sus primeros 17 Requerimientos de Usuarios (38, 39) obtienen un orden relativamente exacto, mientras que, los restantes 15 requerimientos, no adquieren correlato alguno en la ordenación superior propuesta por el EGIPIRU.

5.7 Comparación entre el Esquema General (EGPIRU) y cada Esquema Particular (EPPIRU). Análisis de la Implementación de un Proceso de Negociación Grupal.

En esta sección, se determinará el Grado de Alineamiento que ha adquirido cada uno de los EPPIRU (determinados en la Sección 5.5) en los diferentes escenarios planteados para el cálculo del EGPIRU (Sección 5.6). Para ello, se utilizará el sistema de la memoria asociativa difusa (FAM) especificado en la Sección 4.9 (Capítulo 4).

LGCCA						
Grupo	Orden de Corte	NPG	GCG	GAG(<i>defuzzifier</i>)	Categoría Alineamiento (GAG)	Resultado de la Comparación (RC)
1	18	1,22	0,694	0,6	Aceptable	Mas Perfecto que Imperfecto
2	18	0,45	0,083	0,2	Bajo	Bastante Imperfecto

Tabla 20: "Lista de Grupos Clasificados según Categoría de Alineamiento (Escenario 1)". Fuente: Elaboración Propia

En la Tabla 20, se establece la Lista de Grupos Clasificados según su Categoría de Alineamiento (LGCCA) para el Escenario 1. El Grupo 1 ha obtenido un Nivel de Preferencia "**Sobre-Valorado**" (NPG: 1,22) y un grado de coincidencia "**Moderado-Alto**" en comparación al EGPIRU (GCG: 0,694). En consecuencia, su Grado de Alineamiento es "**Aceptable**" ($GAG_{(d)} = 0,6$) y su esquema de prioridad (EPPIRU) resulta "**Mas Perfecto que Imperfecto**" comparado con el EGPIRU.

Por el contrario, el esquema particular de prioridad del Grupo 2 (EPPIRU), ha sido **“Bastante Imperfecto”** comparado con el EGPIRU lo que redundo en un grado de alineamiento de tipo **“Bajo”** ($GAG_{(d)} = 0,2$). Dicha circunstancia queda fundamentada por un nivel de preferencia **“Sub-Valorado”** (NPG : 0,45) y un grado de coincidencia **“Bajo”** en relación al EGPIRU.

De esta manera, y según lo dispuesto por la Sección 4.9, el Grupo 2 debería ser pasible del Proceso de Negociación especificado en la Sección 3.5 (Capítulo 3). A tal efecto, el Ingeniero de Requerimientos debería iniciar la Etapa de Pre-Negociación del Proceso, sintetizando y analizando los principales datos que se han obtenido para el Grupo en cuestión. **Luego, el Ingeniero en Requerimientos considera que el Grupo 2:**

- Se compone de 4 *stakeholders* (Identificados en el Orden: 1, 4, 5 y 7 en la Tabla 14).
- Obtiene un valor de VTV que resulta 2.5 veces inferior al registrado por el Grupo 1. Asimismo, propone una razón de exclusión (RE) que resulta aproximadamente del doble que la registrada por el Grupo 1 (ver en Tabla 17, Escenario 1). Luego, el Grupo 2 ha obtenido un desempeño socio-cognitivo significativamente menor que el Grupo 1, en términos de la identificación de aquellos conceptos considerados como claves para comprender los usos e implicancias de la futura herramienta de software.
- La preferencia que el Grupo 2 asigna a los primeros 18 requerimientos de usuarios del EGPIRU (Orden de Corte) asciende al 22, 8 % mientras que, el EGPIRU le asigna el 50,65 % (NPG: 045: etiqueta lingüística de la variable: Sub-Valorado).
- El Ingeniero de Requerimientos le comunica a los miembros del Grupo 2, el resultado provisto por el EGPIRU. Luego, fundamenta el porqué de la configuración del grupo (patrón de preferencia), e indica cuales son las principales diferencias que existen entre su esquema de prioridad (EPPIRU) y el acontecido a nivel general (EGPIRU). Finalmente, le solicita a los miembros del Grupo que se pronuncien a favor o en contra respecto de la implementación del EGPIRU. Si lo aceptan, se dará por finalizado el Proceso de Negociación y se aplicará el EGPIRU. Caso contrario se ejecutará la Etapa de Negociación del Proceso, según lo estipulado por la Sección 3.5.

Siguiendo con el mismo razonamiento anterior, pero considerando esta vez lo establecido por el Escenario 2, se obtiene la “Lista de Grupos Clasificados según Categoría de Alineamiento” (Tabla 21):

LGCCA						
Grupo	Orden de Corte	NPG	GCG	GAG(<i>defuzzifier</i>)	Categoría Alineamiento (GAG)	Resultado de la Comparación (RC)
1	17	0,53	0,058	0,2	Bajo	Bastante Imperfecto
2	17	1,19	0,62	0,6	Aceptable	Mas Perfecto que Imperfecto

Tabla 21: “Lista de Grupos Clasificados según Categoría de Alineamiento (Escenario 2)”. Fuente: Elaboración Propia

Como se aprecia en la Tabla 21, **el Grupo 1 será esta vez quien deba someterse al Proceso de Negociación**. Tal circunstancia, se fundamenta por el hecho de que su EPPIRU, a “**Sub-Valorado**” los primeros 17 Requerimientos propuestos por el EGPIRU (NPG: 0,53). También, ha registrado un “**Bajo**” Nivel de Coincidencia para este subgrupo de requerimientos (GCG: 5,8%). En consecuencia, su Grado de Alineamiento resulta “**Bajo**” ($GAG_d : 0,2$) con un esquema de prioridad “**Bastante Imperfecto**” comparado con el EGPIRU. Por el contrario, el Grupo 2 a “**Sobre-Valorado**” los primeros 17 requerimientos plantados por el EGPIRU (NPG: 1,19) y ha obtenido un grado de coincidencia “**Moderado-Alto**” (GCG: 0,6). De esta manera, su Grado de Alineamiento ha sido “**Aceptable**” ($GAG_{(d)} = 0.60$) con un esquema de prioridad “**Más Perfecto que Imperfecto**” comparado con el EGPIRU.

5.8 Análisis del Caso con un Método de Priorización alternativo: Karlsson (1996).

Con el objeto de validar los resultados obtenidos por la implementación del *MPRUSPS*, se procede a considerar la aplicación del enfoque de Karlsson (1996) en este mismo caso propuesto. En efecto, este último enfoque propone la aplicación del método AHP, como medio para determinar un vector que distribuya la preferencia de implementación de una serie de requerimientos disponibles. Luego, el mecanismo de agregación de los vectores se establece de manera lineal, ya que en Saaty (1980), se establece como una de las alternativas viables cuando se utiliza el AHP. En efecto, la determinación de cada EPPIRU en el *MPRUSPS* acontece de la misma manera.

No obstante a ello, una de las diferencias más notables entre el modelo de Karlsson (1996) y el *MPRUSPS*, se halla en el cálculo del orden global o general de los requerimientos. El primero de ellos, propone una agregación lineal entre todos los vectores de preferencia (*stakeholders*). Por su parte, el *MPRUSPS* propone una agregación ponderada (VTV) de cada una de las preferencias grupales (EPPIRU). A continuación, se muestra el resultado obtenido tras aplicar el modelo de Karlsson (1996) en el caso analizado.

KyS	Fr.Acu. (%)	KyS	Fr.Acu. (%)	EPPIRU(G1)	Fr.Acu. (%)	EPPIRU(G1)	Fr.Acu. (%)	EPPIRU(G2)	Fr.Acu. (%)	EPPIRU(G2)	Fr.Acu. (%)
38	3,82	35	65,39	38	5,49	44	80,89	13	5,74	16	78,81
39	7,53	6	67,45	39	10,48	45	82,24	12	10,91	2	80,28
37	10,79	10	69,48	37	15,36	3	83,58	19	15,24	49	81,65
40	13,81	41	71,39	40	19,72	1	84,83	18	19,53	38	82,95
13	16,57	31	73,28	26	23,94	4	86,03	5	23,75	41	84,23
12	19,27	11	75,14	27	27,93	11	87,21	44	27,82	24	85,49
26	21,94	29	76,96	33	31,70	6	88,31	45	31,57	30	86,74
27	24,59	30	78,77	36	35,45	17	89,36	17	35,32	50	87,92
24	27,18	22	80,54	25	39,04	12	90,41	4	38,88	31	89,05
25	29,70	21	82,19	34	42,60	19	91,25	6	42,39	20	90,16
44	32,22	1	83,80	24	46,08	18	92,07	14	45,72	23	91,23
36	34,61	14	85,38	32	49,43	20	92,88	11	48,59	15	92,25
33	36,97	3	86,94	35	52,73	5	93,69	43	51,46	40	93,26
43	39,29	8	88,48	28	55,80	8	94,46	42	54,32	22	94,26
45	41,60	2	90,00	23	58,69	13	95,22	8	57,03	25	95,18
19	43,83	46	91,50	31	61,08	9	95,98	7	59,71	21	96,08
34	46,06	7	92,96	41	63,43	46	96,71	46	62,39	37	96,91
18	48,26	9	94,12	22	65,70	50	97,39	1	64,54	28	97,69
42	50,43	47	95,10	30	67,88	7	98,03	47	66,61	27	98,33
5	52,61	16	96,07	21	70,04	16	98,62	3	68,50	26	98,71
23	54,77	20	97,00	10	72,19	14	99,03	10	70,35	36	99,05
28	56,92	50	97,88	29	74,20	15	99,40	48	72,17	35	99,31
4	59,06	48	98,75	43	76,15	47	99,65	39	73,96	33	99,55
17	61,20	49	99,41	42	77,87	48	99,88	9	75,71	32	99,78
32	63,30	15	100	2	79,42	49	100	29	77,27	34	100

Tabla 22 "Orden de Prioridad de los Requerimientos de Usuarios según el modelo de Karlsson (1998)". Fuente: Elaboración Propia.

Como se aprecia en la Tabla 22, se ha determinado por aplicación del modelo de Karlsson (1996), el orden de implementación de cada uno de los Requerimientos de Usuarios disponibles (Columna KyS). El Orden de Corte, se establece para los primeros 19 requerimientos de usuarios, los cuales totalizan el 50,43% de las preferencias globales. Comparativamente, el MPRUSPS ha determinado para el Escenario 1, un Orden de Corte de 18 requerimientos (ver en Tabla 18) totalizando el 51,8 % de las preferencias globales, mientras que, para el Escenario 2, se establece un Orden de Corte de 17 requerimientos concentrando el 50,65 % de las preferencias globales (ver Tabla 19). **Luego, no existirá para este concepto en particular (Orden de Corte) diferencias significativas en torno a la aplicación del MPRUSPS ó la propuesta de Karlsson (1996).** Por su parte, el modelo de Karlsson (1996), concluiría con el actual orden de implementación de los requerimientos de usuarios.

Sin perjuicio de ello, y aunque en Karlsson (1996) no se contemple, el patrón de preferencia de los *stakeholders*, fundamenta la existencia de dos grupos bien diferenciados, tal cual se ha determinado en la Sección 5.5 del presente. En consecuencia, la utilización de un esquema lineal para agregar preferencias que difieren

significativamente entre sí (al menos entre grupos) podría generar un orden de implementación que no sea representativo para ninguno de los grupos consignados. A fin de validar esta última hipótesis, se establece la Lista de Grupos Clasificados según Categoría de Alineamiento (Sección 4.9 Capítulo 4). Dicha lista puede ser aplicable, puesto que los EPIRU (en Tabla 22) se determinan de la misma manera que especifica Karlsson (1996). En consecuencia, el análisis devendría de la comparación entre el orden general propuesto por Karlsson (1996) -EPIRU en semántica del MPRUSPS-, y los EPIRU propuestos en cada uno de los grupos.

LGCCA						
Grupo	Orden de Corte	NPG	GCG	GAG(<i>defuzzifier</i>)	Categoría Alineamiento (GAG)	Resultado de la Comparación (RC)
1	19	1,11	0,39	0,5	Moderado	Tan Imperfecto como Perfecto
2	19	0,83	0,18	0,3	Moderado	Algo Imperfecto

Tabla 23: “Lista de Grupos Clasificados según Categoría de Alineamiento (Método de Karlsson)”. Fuente: Elaboración Propia

Como se puede apreciar en la Tabla 23, **se puede confirmar que el orden de implementación general propuesto por Karlsson (1996) pierde representatividad entre los grupos consignados.** Si bien el Grupo 1 ha obtenido un Nivel de Preferencia “**Sobre-Valorado**” (NPG: 1,1), su grado de coincidencia resulta “**Bajo**” en comparación con el esquema general ofrecido por Karlsson (GCG: 0,39). Entonces, su Grado de Alineamiento es “**Moderado**” ($GAG_{(d)} = 0,5$) ya que su esquema de prioridad (EPIRU) resulta “**Tan Imperfecto como Perfecto**” comparado con el orden general. Luego, se aprecia un comportamiento similar para el Grupo 2.

Dichas circunstancias, según el MPRUSPS, motivaría la aplicación de un proceso de negociación en ambos grupos. **Con lo cual, el orden de implementación consignado por Karlsson (1996) no sería representativo bajo el supuesto de que existan preferencias que difieran significativamente entre sí entre los stakeholders.**

5.9 Consideraciones finales del Capitulo

El *MPRUSPS*, se postula como un enfoque experimental para el tratamiento integral del problema que supone priorizar y negociar los requerimientos de un determinado Proyecto de Software. Como consecuencia de ello, el presente Capitulo propuso la experimentación del modelo, a través de la presentación de un típico Proyecto de Software, compuesto por 50 requerimientos a evaluar por parte de un conjunto representativo de *stakeholders* (10).

Por su parte, se ha planteado la aplicación del enfoque de Karlsson (1996), a los efectos de evaluar el orden de prioridad de los requerimientos del Proyecto presentado. Para ello, se ha establecido un mecanismo de agregación de tipo lineal entre las preferencias de los *stakeholders*. Luego, se ha comparado los resultados obtenidos por dicho enfoque y el *MPRUSPS*.

A través de este caso simulado, se ha podido constatar que la existencia de grupos de *stakeholders* (aglomerados por preferencias similares) genera una pérdida de representatividad del ordenamiento propuesto por el enfoque de Karlsson (1996). Esta circunstancia, acontece como consecuencia del esquema de agregación lineal de las preferencias. *Es decir, cuando las preferencias de los stakeholders se tornan muy diferentes entre sí, su agregación lineal, se verá influida por los valores extremos de las preferencias, perdiendo en consecuencia, la representatividad del centroide propuesto (EGPIRU).* Por su parte, si se hubiese planteado otro tipo de agregación, por ejemplo geométrico, se podría mitigar el efecto de los valores extremos, sin embargo, la cuestión de la representatividad del centroide, aun se vería cuestionada.

Por tal motivo, el *MPRUSPS* aplica una metodología de agregación un tanto diferente. En primera instancia, constata la existencia de subgrupos de *stakeholders* dentro del universo, bajo el criterio de preferencias similares (operado por la aplicación del algoritmo de Kohonen). De hallar dichos grupos, el *EPPIRU* propone un esquema de agregación lineal, tal cual propone Saaty (1980). Sin embargo, tal agregación acontece en un mismo grupo, el cual se supone, aloja *stakeholders* con preferencias similares. Luego, la agregación lineal propuesta en el *EPPIRU*, debería redundar en un esquema representativo del grupo. Finalmente, el *EGPIRU* se determinaría mediante la agregación de las preferencias inter-grupales (*EPPIRU*) ponderadas por el desempeño socio-cognitivo de cada uno de los grupos. En consecuencia, el orden general de implementación (*EGPIRU*) se encontraría mas sesgado hacia aquellos *EPPIRU* que se formulan a partir de grupos cuyo desempeño socio-cognitivo, sea relativamente más elevado que el resto.

Por su parte, en aquellos grupos en donde se constate un *Grado de Alineamiento* (GAG) “*Bajo*” ó “*Moderado*” en comparación con el EGPIRU, se le aplicará un proceso de negociación focalizado. En conclusión, el MPRUSPS no sólo ofrece un esquema general de prioridad (EGPIRU) más representativo que el provisto por el Karlsson (1996), sino que además, identifica aquellos grupos que distan de dicho orden. Luego, y en base a las particularidades de cada grupo (cantidad de *stakeholders*, VTV, RE, entre otras), implementa un proceso de negociación focalizado, con la finalidad de resolver los conflictos que se tengan a lugar, ante preferencias disímiles.

Capítulo 6

“Caso Experimental: Subsistema de Flujo de Trabajo del CIMEPB”

6.1 Introducción

El Centro de Investigación en Procesos Básicos, Metodología y Educación (CIMEPB), cuyo director es el Dr. Jorge Vivas, es un Órgano dependiente de la Facultad de Psicología de la Universidad Nacional de Mar del Plata. Asimismo, dicho Centro, se especializa en investigaciones inherentes a la Psicología Cognitiva, con una fuerte inclinación metodológica hacia las áreas de Sistemas, Cálculo e Inteligencia Artificial.

Por tal motivo, en el CIMEPB colaboran otros Grupos de Investigación, como el de Bioingeniería (GB) y el de Inteligencia Artificial aplicada a la Ingeniería (GIA), ambos pertenecientes a la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata. De esta manera, se logran producciones académicas, que promueven la colaboración de diferentes Investigadores en un contexto plenamente interdisciplinario.

En función de las circunstancias descritas, el CIMEPB se ha presentado recientemente a las convocatorias del Fondo para la Investigación Científica y Tecnológica (FONCYT), mediante la presentación de un Proyecto de Investigación, caratulado “Normas de Producción de atributos semánticos en castellano rioplatense para un conjunto de objetos vivos y no vivos”. Entre otras cuestiones, el Proyecto prevé el desarrollo de una Aplicación Web (CIMEPB- GIA), denominada “Sistema de Investigación Aplicada-CIMEPB-WEB” que servirá a los propósitos de difundir las investigaciones corrientes del Centro de Investigación, y también, la de promover un marco de trabajo colaborativo entre investigadores regionales en temáticas inherentes a la Psicología Cognitiva.

Dicha herramienta, entre otras funciones, deberá soportar la funcionalidad básica de un *“workflow”* o Flujo de Trabajo, a los efectos de automatizar el proceso de elaboración de los Documentos Electrónicos Académicos a través de las acciones coordinadas y colaborativas de los Participantes, de acuerdo a un conjunto de procedimientos especificados en el dominio de la aplicación. Por tal motivo, y a los efectos de la presente Tesis, se ha procedido a especificar los Requerimientos de Software que la mencionada Herramienta, debería satisfacer.

6.2 Alcance del Subsistema de Flujo de Trabajo (*Workflow*)

El *workflow*, se constituye como un subsistema del Sistema de Investigación Aplicada del CIMEPB-Web. El mismo, se denomina “Subsistema de Flujo de Trabajo” y proveerá la funcionalidad básica que permitirá la interacción de los Usuarios Registrados del Sistema, a los efectos de promover la elaboración colaborativa de los Documentos Académicos Electrónicos del CIMEPB.

6.2.1 Objetivo General del *Workflow*

- Facilitar la participación activa de los Usuarios Registrados, mediante la interacción coordinada de sus acciones académicas.

6.2.2 Objetivos Particulares del *Workflow*

- Promover la elaboración de documentos electrónicos académicos de acuerdo a los procedimientos de trabajo especificados.
- Gestionar la distribución de ítems de trabajo entre los colaboradores de un determinado Flujo de Trabajo.

6.2.3 Definiciones y Siglas

- *SIA-CIMEPB- Web*: Sistema de Investigación Aplicada del Centro de Investigación en Procesos Básicos, Metodología y Educación. Dicho Sistema, se refiere a una Aplicación Web.
- *SFT*: Subsistema de Flujo de Trabajo, de la aplicación *SIA-CIMEPB- WEB*.
- *DEA*: Documento Electrónico Académico, registro de información cuyo contenido puede albergar: Texto, Tabla, o Imagen.
- *Título del DEA*: Título del Documento Electrónico Académico.
- *Contenido del DEA*: Se refiere al contenido de información registrada en el Documento Electrónico Académico en un determinado momento.
- *DEA Finalizado*: Se refiere al momento por el cual, el Contenido del DEA se encuentra ya finalizado.

- *Listado de DEA Finalizados: Se refiere a una Tabla de 5 Columnas y “n” Filas (dada la cantidad disponible de DEA finalizado en un determinado momento), en el cual la Columna 1 titula: Colaborador Responsable, Columna 2: Fecha de Iniciación, Columna 3: Fecha de Finalización, Columna 4: Cantidad de Autores, Columna 5: Título del DEA.*
- *Colaborador Responsable: Se refiere al Colaborador (ver Perfil de Usuario) que crea un Flujo de Trabajo (FT) y se responsabiliza por su gestión.*
- *FT: Flujo de Trabajo: Se refiere a la colección de Ítems especificados por un Colaborador Responsable, a los efectos de constituir un DEA y los Colaboradores (ver perfil de usuario) afectados al mismo. Componentes del FT:*
 - *Colección de Ítems /Estados de cada uno de ellos.*
 - *Colección de Colaboradores.*
 - *Título del DEA.*
 - *Contenido del DEA.*
 - *Fecha de Creación.*
 - *Fecha de Finalización.*
- *FT finalizado: Circunstancias que acontece cuando todos los Ítem contenidos en el FT se encuentra en estado “finalizado”.*
- *Ítem: Se refiere a la actividad que debe desarrollar un cierto Colaborador sobre el DEA especificado. En cuanto a los atributos del Ítem, se considera:*
 - *Responsable Asignado: Se refiere al Colaborador asignado para ejecutar la acción prevista del Ítem.*
 - *Título del DEA: Se refiere al título del DEA al cual pertenece el ítem corriente.*
 - *Actividad: Acción concreta que el Responsable Asignado deberá ejecutar.*
 - *Contenido del DEA: Contenido de información registrada en el DEA, al momento de asignarse el ítem al Responsable Asignado.*

- *Fecha de Constitución: Día, hora y minutos, en que un determinado Ítem fue catalogado como: “Por asignar” por parte del Colaborador Responsable.*
- *Fecha de Asignación: Día, hora y minutos, en que un determinado Ítem, ha sido catalogado como “Asignado” por parte del Colaborador Responsable.*
- *Fecha de Iniciación: Día, hora y minutos, en que un determinado Ítem, ha sido catalogado como “En Proceso” por parte del Responsable Asignado.*
- *Fecha de Interrupción: Día, hora y minutos, en que un determinado Ítem fue catalogado “En Espera” por parte de un Responsable Asignado.*
- *Fecha de Finalización: Día, hora y minutos, en que un determinado Ítem, fue catalogado como “Finalizado” por parte del Responsable Asignado.*
- *Estados del Ítem: Se refiere a las secuencias por las que un determinado ítem transita a lo largo de su ciclo de vida. Éstos pueden ser:*
 - *Por Asignar: Estado inicial del Ítem. Es decir cuando el Colaborador Responsable lo crea y aún no lo ha asignado a un Responsable Asignado.*
 - *Asignado: Estado por el cual el ítem se asigna a un determinado Responsable Asignado, a los efectos que éste proceda con el cumplimiento de las acciones encomendadas.*
 - *En Proceso: Estado por el cual un Ítem comienza a ser ejecutado por un determinado Responsable Asignado.*
 - *En Espera: Estado por el cual un ítem sufre una espera intencionada por parte del Responsable Asignado.*
 - *Finalizado: Estado por el cual un ítem se da por cumplimentado por parte del Responsable Asignado.*

- *Reporte actualizado del FT: Se refiere a una versión particular del FT generada automáticamente por el SFT, cada vez que se procede a un cambio en los componentes de un FT:*
 - *Colección de Ítems /Estado de cada uno de ellos*
 - *Colección de Colaboradores.*
 - *Titulo del DEA.*
 - *Contenido del DEA*

6.2.4 Referencia

El “*MoReq*” (Model Requirements for Management of Electronic Records), es un Modelo de Requerimientos para la Gestión de Documentos Electrónicos, que surgió gracias a la iniciativa de la Unión Europea a través del DLM Forum (Document Lifecycle Management Forum) [Casellas i Serra (2009)]. Desde su primera publicación en 2001, el Modelo de Requerimientos *MoReq*, ha sido ampliamente utilizado en Europa por usuarios, expertos en gestión de documento electrónicos y por desarrolladores, con el fin de propiciar las bases necesarias para la constitución de Sistemas de Gestión de Documentos Electrónicos [DLM Forum(2008)].

Por ejemplo, la Especificación *MoReq2*, está escrita para describir parcialmente los aspectos funcionales y no funcionales de una Aplicación de Software destinada a gestionar documentos electrónicos. Asimismo, y de acuerdo al dominio de aplicación, será factible determinar cuáles requerimientos aplicar o no, de acuerdo a las necesidades locales [DLM Forum(2008)].

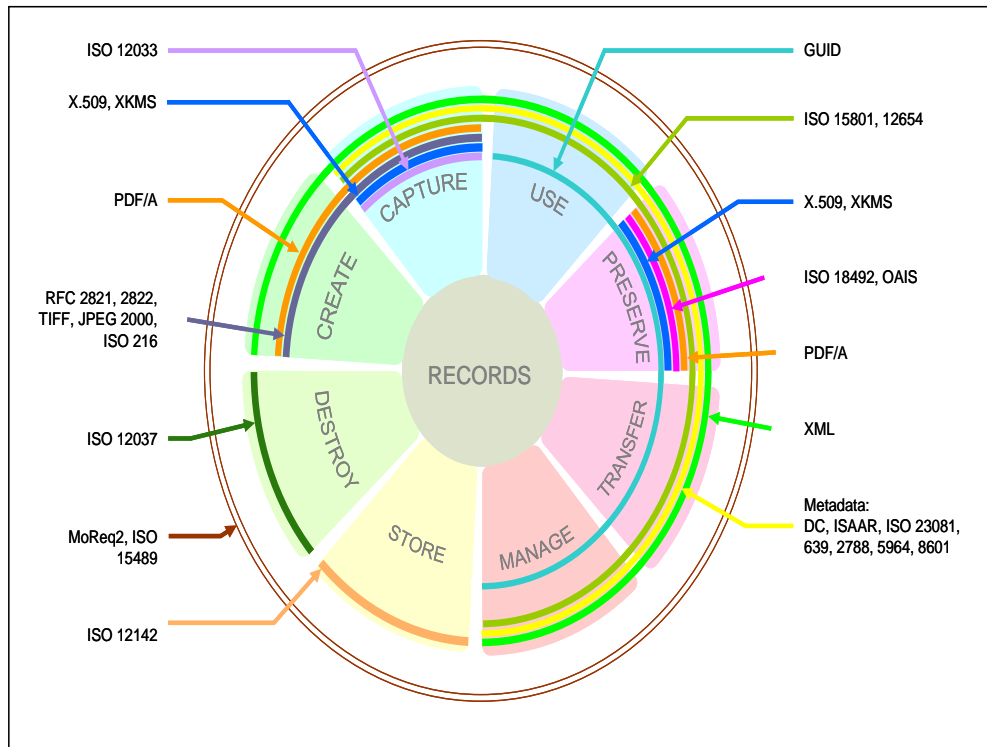


Figura 6.1: “Estructura del MoReq2: Procesos y Estándares”. Fuente: [DLM Forum(2008)].

Como se aprecia en la Figura 6.1, la Estructura del *MoReq2* se basa en procesos archivísticos de gestión documental, que incluyen las recomendaciones efectuadas por las Normas ISO y otros estándares. En consecuencia, se ha constituido como una Especificación de Requerimientos básica, que debería observarse en todo Proyecto de Software que tenga como propósito la Gestión de Documentos Electrónicos.

Por su parte, la especificación describe funcionalmente una serie de módulos opcionales. Uno de ellos, el *Workflow*, está descrito para ser utilizado como un subsistema para la gestión de documentos electrónicos. Asimismo, su Especificación Funcional, se basa en las recomendaciones provista por la *WfMC*.

La Asociación Internacional “*Workflow Management Coalition (WfMC)*”, es una Institución que promueve sus esfuerzos para el desarrollo de normas de flujo de trabajo. En efecto, define al “*workflow*” como la automatización de un proceso de negocio, en su totalidad o en parte, durante el cual, los Documentos, Tareas o Informaciones, son transferidas de un Participante a otro, a los efectos de propiciar una acción determinada de acuerdo a un conjunto de procedimientos especificados en el dominio de aplicación. Luego un “Participante” puede ser un Usuario, un Equipo de Trabajo o una Aplicación de Software.

En el contexto de especificación de un Sistema de Gestión de Documentos Electrónicos, el *Workflow* se concibe como una herramienta para administrar documentos entre usuarios autorizados. Generalmente, y siguiendo las disposiciones del DLM Forum(2008) el *Workflow* se utiliza entre otras funciones, para:

- Gestionar documentos de acuerdo a los procedimientos de trabajo especificados.
- Distribuir documentos entre miembros participantes a los efectos de propiciar su constitución/inscripción.
- Gestionar procesos críticos inherentes a la comunicación entre los participantes.
- Especificar acciones concretas basadas en procedimientos de trabajos, con el objeto de propiciar la constitución de Documentos erigidos a través de un contexto colaborativo.

6.3 Descripción General.

6.3.1 Perspectiva y funciones del SFT

El SIA-CIMEPB- WEB representa un sistema que se encuentra diseñado para trabajar en entorno WEB, lo que permitirá su utilización de manera descentralizada. Como se ha mencionado, uno de los objetivos del sistema, consiste en difundir las investigaciones corrientes del CIMEPB. En consecuencia, el sistema deberá implementar aquella funcionalidad que le permita gestionar documentos electrónicos académicos.

Por su parte, el *SFT* se refiere a un Subsistema de Flujo de Trabajo, que permitirá soportar las acciones colaborativas de los Participantes, en pos de la elaboración de Documentos Electrónicos Académicos (DEA). En consecuencia, las funciones básicas de la herramienta, constarán de:

- Soportar la generación de Flujo de Trabajo.
- Asignar Ítems a los Usuarios Registrados para que contribuyan desde su perspectiva a la elaboración de los DEA.
- Comunicar a los Usuarios Registrados, respecto de las versiones que irán adquiriendo los Flujos de Trabajo que inicien o en los que participen.

6.3.2 Interfaces del SFT

Dado que el SFT debe interactuar con otros componentes y subsistemas a especificar de la aplicación general SIA-CIMEPB-WEB, deberá establecer una *Interface de Software* a los efectos de transaccionar datos con dichos elementos. En ese sentido, utilizará la Base de Datos aplicable al SIA-CIMEPB y su Servidor de Internet.

Respecto a la *Interfaz de Usuario*, una vez peticionada la opción de Iniciar el Subsistema, las pantallas desplegadas, se navegarán al igual que con el resto de los elementos de la aplicación general. Es decir, mediante el uso de un *Browser*, como por ejemplo MS Explorer, Mozilla, entre otros. Asimismo, el *SFT* deberá incluir interfaces gráficas para que los Usuarios Registrados puedan definir, mantener y editar *FT*.

6.3.3 Perfil de Usuario

El SIA-CIMEPB- WEB, (SIA- CIMEPB- WEB) deberá interactuar con diferentes tipos de Usuarios. En consecuencia, el *SFT* operará en base a las tipologías básicas previstas para el Sistema general. Los tipos de usuarios contemplados son:

- *Visitante*: Persona catalogada como “Usuario no Registrado” que visita la Página Web del SIA- CIMEPB- WEB, para consultar la información allí registrada. Al efecto, no requiere credencial alguna para la acción descripta. Tendrá la posibilidad de contactarse (Vía Pagina WEB) con el Administrador del Sistema (Rol a especificarse en el SIA- CIMEPB- WEB) , a los efectos de poder establecerse en alguna categoría de “Usuario Registrado”
- *Colaborador*: Persona catalogada como “Usuario Registrado” que contribuye, desde diferentes perspectivas, con la producción académica del CIMEPB.
- *Investigador en Psicología*: Persona que se desempeña profesionalmente como Investigador o Becario en Psicología dentro del ámbito del CIMEPB, ó en su defecto, en otro organismo. Asimismo, se categoriza como una clase de *Colaborador*.
- *Asesor en Modelado de Datos*: Persona que se desempeña profesionalmente como Investigador o Becario en áreas de Ingeniería, Matemática, Sistemas, u otra disciplina, que se desempeña en el CIMEPB, ó en su defecto, en otro organismo, y que contribuye con el modelado de los datos relevados por el

Centro de Investigación. Asimismo, se categoriza como una clase de *Colaborador*.

- *Recopilador de Datos*: Persona que se desempeña como entrevistador o encuestador a los fines investigativos del CIMEPB. Asimismo, se categoriza como una clase de *Colaborador*.

6.3.3.1 Jerarquía de Usuarios

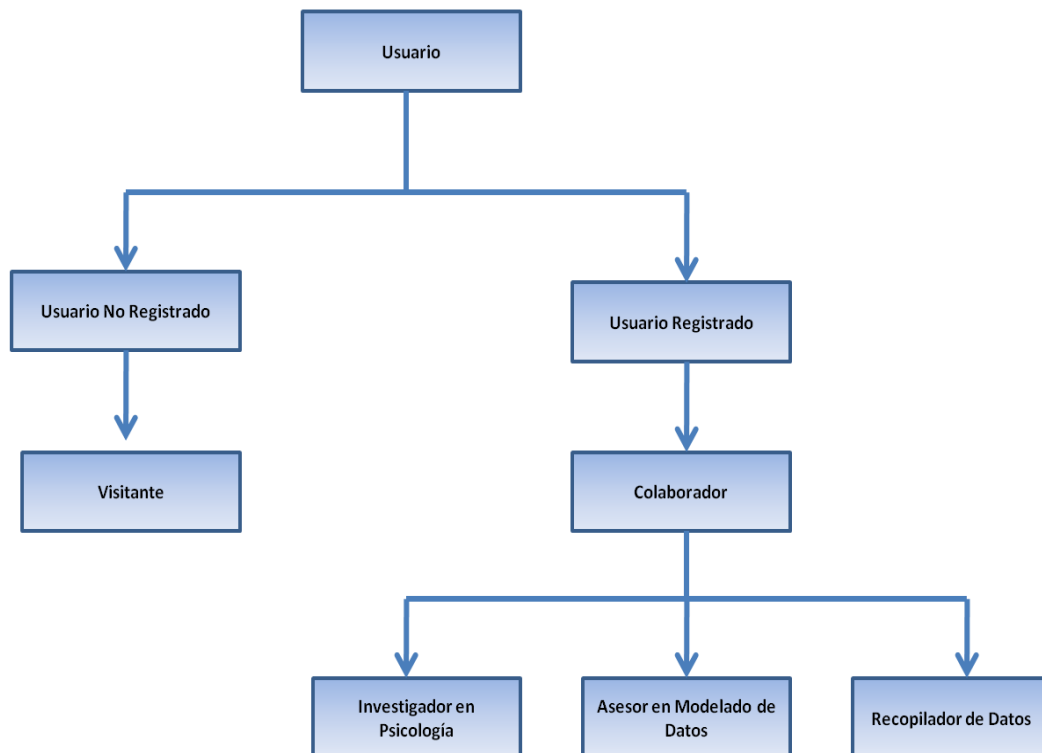


Figura 6.2 “Jerarquía de Usuarios del SIA- CIMEPB- WEB”. Fuente: Elaboración Propia.

Como se aprecia en la Figura 6.2, se establece la Jerarquía de Usuarios prevista para el SIA-CIMEPB-WEB. Luego, el Subsistema de Flujo de Trabajo, deberá interactuar con la tipología en cuestión. La categoría de Usuario, es clasificable en dos tipos: No Registrado y Registrado. Como se ha mencionado, el Usuario No Registrado, hace referencia a una persona que visita la Página Web del SIA- CIMEPB- WEB, para consultar la información allí registrada. Al efecto, no requiere credencial alguna para la acción descrita (Visitante).

Por su parte, el Usuario Registrado, hace referencia a una persona que contribuye con la producción académica del CIMEPB (Colaborador). Asimismo, un Colaborador es una persona que ejerce un rol específico dentro del proceso de investigación del CIMEPB. En efecto, los roles previstos son: Investigador en Psicología, Asesor en Modelado de Datos y Recopilador de Datos. Finalmente, para que un Usuario tenga condición de Registrado, deberá ser validado previamente por el Administrador del Sistema.

6.3.4 Casos de Uso relacionados al SFT

A continuación se ofrece el modelo del Subsistema de Flujo de Trabajo (SFT) desde el punto de vista de los usuarios. Como se puede apreciar en la Figura 5.3, se establece el Caso de Uso de cada uno de los participantes (Colaborador Responsable, Responsable Asignado, Colaborador) que interactúan con el mencionado subsistema.

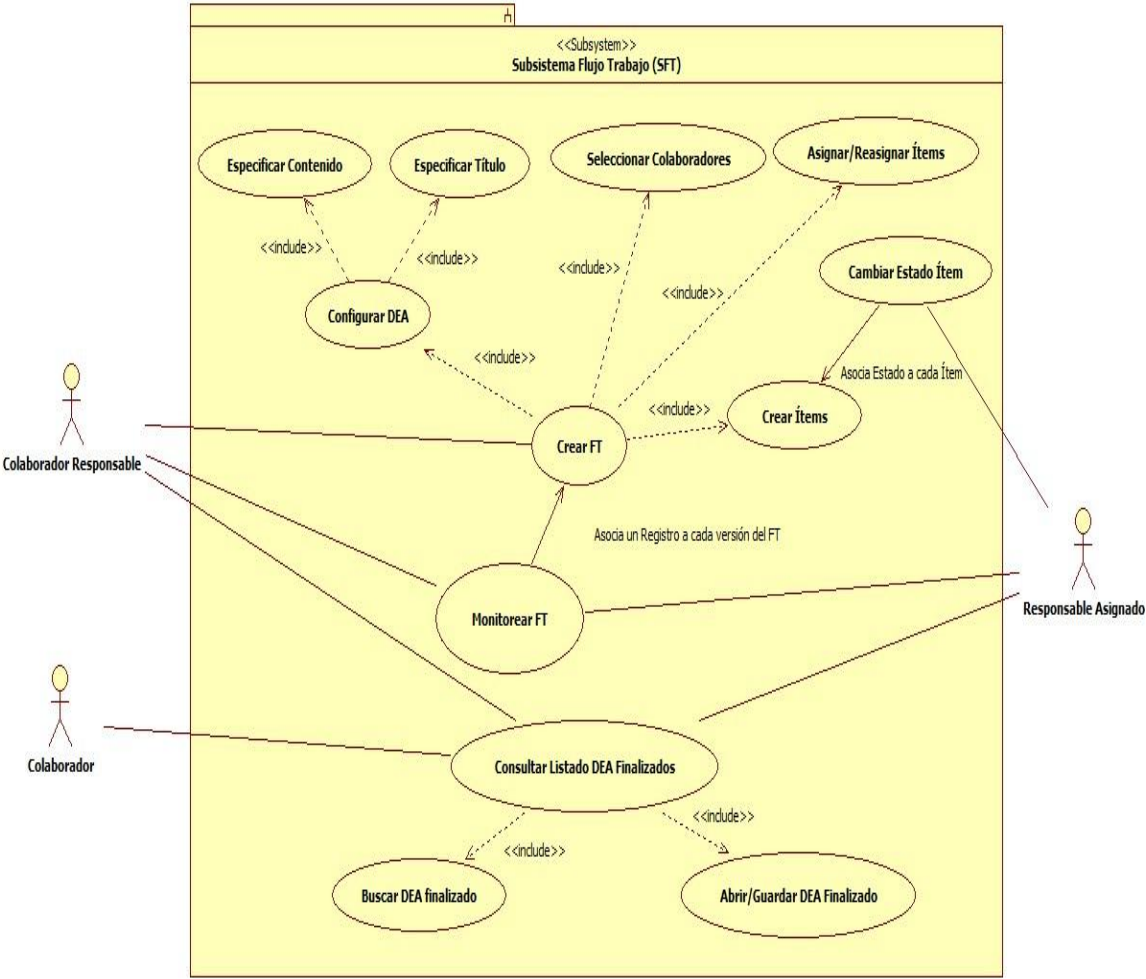


Figura 6.3: “Casos de Uso del SFT”. Elaboración Propia.

En la Figura 6.3 puede apreciarse los diferentes casos de uso del Subsistema de Flujo de Trabajo. A continuación, se establecerán 4 Requerimientos de Usuarios que deberán ser satisfechos por los diversos requerimientos del sistema:

RU: 1 *“Un Colaborador Responsable podrá ser capaz de crear Flujos de Trabajo y/o monitorear el progreso del Flujo de Trabajo que haya creado o en el que participe y/o consultar el listado de DEAs finalizados”*

RU: 2 *“Un Colaborador podrá ser capaz de consultar el listado DEAs finalizados”*

RU: 3 *“Un Colaborador Responsable podrá ser capaz de verificar si un flujo de trabajo se encuentra finalizado. Asimismo deberá ser capaz de constatar los metadatos resultantes del DEA”*

RU: 4 *“Un Responsable Asignado podrá ser capaz de verificar el Estado de un Ítem. Asimismo podrá ser capaz de cambiar su Estado. Por su parte, y dependiendo de las instrucciones del Ítem, podrá modificar el contenido del DEA”*

6.4 Especificación de los Requerimientos Funcionales del Subsistema de Flujo de Trabajo (SFT)

A continuación, se enuncian los Requerimientos Funcionales del SFT. Los mismos tendrán como objetivo básico, soportar la funcionalidad básica que demanda la satisfacción de los 4 Requerimientos de Usuarios señalados con anterioridad.

1. El SFT debe permitir al *Colaborador Responsable* iniciar un *Flujo de Trabajo (FT)*, especificando: el *Título del DEA*, los *Ítems* a desarrollar, los *Colaboradores* afectados al corriente *FT*, y el *Contenido del DEA*. En esta instancia, y por defecto, el SFT debe catalogar a los *Ítem* en *Estado “Por Asignar”*, con su correspondiente *Fecha de Constitución* y el *Contenido del DEA* en un Documento en Formato Texto en blanco con su correspondiente *Título del DEA*. El SFT establece la *Fecha de Iniciación del FT*.
2. El SFT debe notificar a un *Colaborador*, cada vez que se le asigne un *Ítem*. El *estado del ítem* pasará de ser *“Por asignar”* a *“Asignado”*. El SFT establece la *Fecha de Asignación del Ítem*.
3. El SFT debe permitir al *Responsable Asignado* a un determinado *ítem*, cambiar su *estado a*: *“En Proceso”*, *“En Espera”* o *“Finalizado”* según circunstancias. El SFT

establece al efecto, las *Fecha de Iniciación, Interrupción o Finalización* según corresponda para el corriente *ítem*.

4. El SFT debe permitir que el *Responsable Asignado* modifique el *Contenido del DEA* creando, borrando o cambiando Texto, Tabla o Imagen. (Actividad) Sin embargo, el SFT debe restringir la acción de modificación de aquel *Contenido del DEA* especificado por otro/s *Responsables Asignados* del corriente FT.
5. El SFT debe permitir almacenar un *Reporte actualizado* de cada FT en donde conste: la colección de *Ítems* especificados, el historial de *estados*, los *Responsables Asignados*, el *Título* y *Contenido del DEA* al momento de consulta del reporte.
6. El SFT debe permitir que los *Colaboradores* puedan monitorear (chequear) el progreso de los Flujos de Trabajo (*Reporte actualizado del FT*) que Inicien (*Colaborador Responsable*) y en los que participen (*Responsables Asignados*).
7. El SFT debe permitir al *Colaborador Responsable*, reasignar un *Ítem* a otro *Colaborador*, si el *Responsable Asignado* del ítem en cuestión, ha especificado un estado “En espera”. Luego, el *Ítem* pasa de estado “En espera” a “Por Asignar” hasta que el *Colaborador Responsable* procesa a su reasignación.
8. Cuando todos los *Ítem* de un FT en particular, se encuentren en estado “Finalizado”, el SFT deberá notificar al *Colaborador Responsable* de dicha circunstancia. El SFT establece la condición de FT *finalizado*, constituyendo la *Fecha de Finalizado* en el FT.
9. Cuando un FT se encuentra en estado finalizado, el SFT considera la condición de *DEA finalizado*. En esta instancia, el SFT debe agregar en el margen superior izquierdo del DEA, los siguientes metadatos:
 - Título del DEA.
 - Autores: Colección de Responsables Asignados.
 - Cantidad de Autores: Cantidad Responsables Asignados.
 - Fecha de Iniciación: Fecha de creación del FT.
 - Fecha de Finalización: Fecha de finalización del FT.

- Tiempo en Días Transcurridos: Fecha de finalización del FT - Fecha de creación del FT.

10. El SFT debe permitir a cualquier *Colaborador* consultar los DEA *Finalizados*, utilizando los criterios de búsqueda:

- *Por Autor*
- Por Fecha de Iniciación
- Por Fecha de Finalización
- O por defecto, consultar Todos los disponibles al momento.

Luego, los datos que se mostrarán de cada DEA, serán los siguientes: Colaborador Responsable, Responsables Asignados, Título, Fecha de Iniciación y Fecha de Finalización.

11. El SFT debe permitir a cualquier *Colaborador* abrir (sólo para lectura) y Guardar en su PC, los DEA *finalizados* disponibles en el *Listado de DEA Finalizados*.

6.5 Trazabilidad Enriquecida: Relación entre los Requerimientos de Usuarios y los Requerimientos del Sistema

Dado que el objetivo básico es modelar un esquema de priorización de requerimientos de usuarios, necesariamente se deberá describir alguna técnica que permita trazar los mismos, a partir de la disponibilidad de una serie de requerimientos del sistema. En este contexto, se define a la Trazabilidad de los Requerimientos como una Técnica que permite comprender como los requerimientos de alto nivel se transforman en requerimientos de bajo nivel. En efecto, y tal cual lo indica Motupally (2008), dicho procedimiento resulta clave en el objeto de enlazar requerimientos de usuarios (alto nivel) con requerimientos funcionales del sistemas (bajo nivel).

La Trazabilidad Enriquecida, es una técnica introducida por Dick (2000) que propone implementar una declaración textual entre los requerimientos de usuarios y una relación definida entre sus correspondientes requerimientos del sistema. La relación en cuestión, es de tipo lógica y podrá ser disyuntiva (or) o conjuntiva (&). Será disyuntiva cuando resulta necesario la contribución de al menos un Requerimiento del Sistema, para que el argumento de satisfacción contenido en el Requerimiento del Usuario, pueda

mantenerse. Por otro lado, una relación de conjunción demanda la contribución de todos los requerimientos del sistema para que el argumento de satisfacción contenido en el Requerimiento del Usuario, pueda mantenerse [Hull (et. al., 2005)]. A continuación, se aplicará ésta Técnica a los efectos de trazar los Requerimientos de Usuarios (6.3.4) con los respectivos Requerimientos Funcionales (6.4).

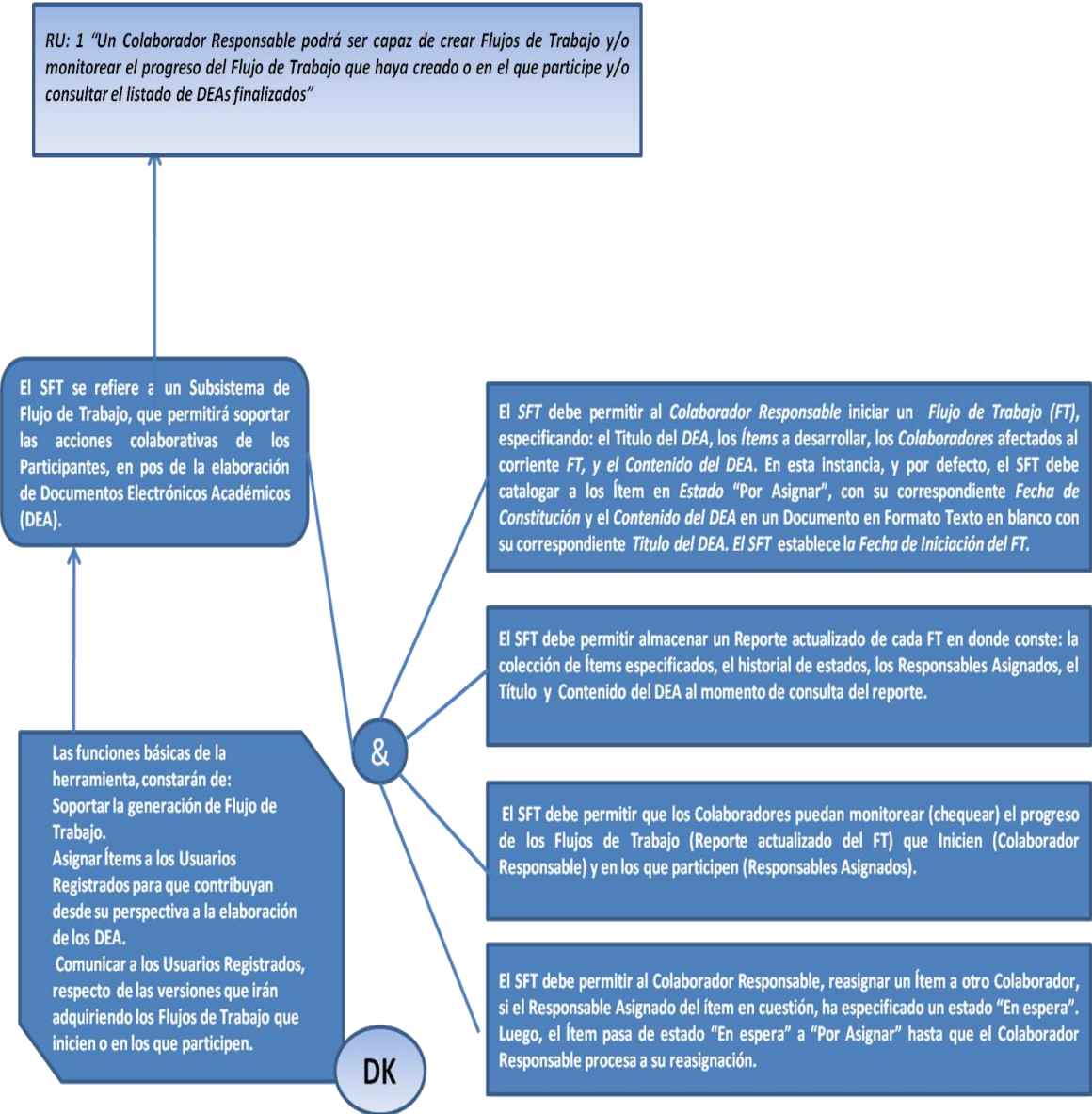


Figura 6.4 "Trazabilidad Enriquecida: Primer Requerimiento de Usuario". Fuente: Elaboración Propia.

En la Figura 6.4, se ha aplicado la Técnica de Trazabilidad Enriquecida al primer Requerimiento de Usuario especificado. Como se puede apreciar, el mismo puede satisfacerse, sólo y sólo sí, los requerimientos funcionales: 1, 5, 6 y 7 son implementados en forma conjunta [“&” Indica conjunción en sentido de Dick (2000)]. Asimismo, se ha insertado dos etiquetas textuales que definen al Flujo de Trabajo y sus funciones básicas en este dominio en particular.

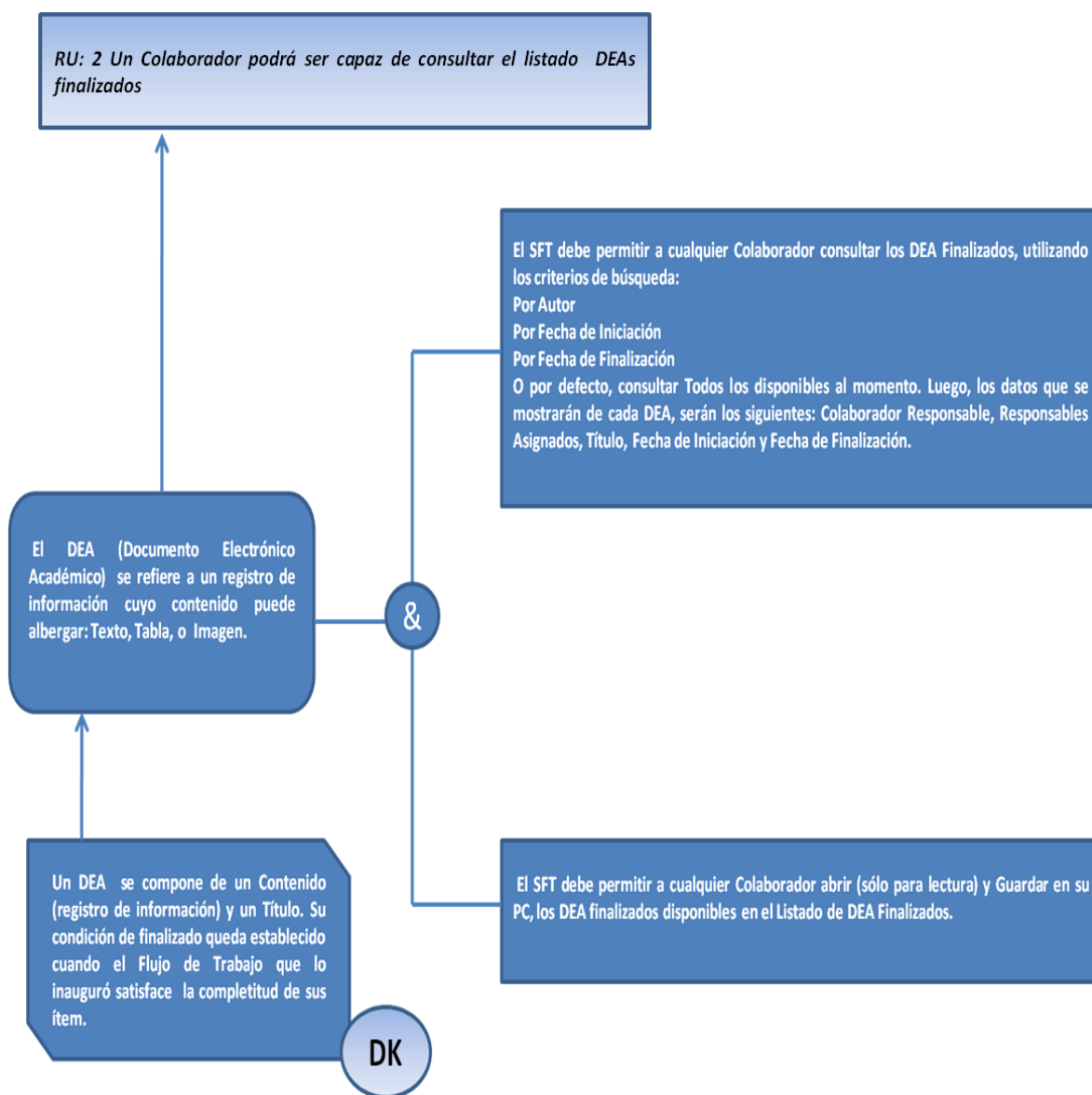


Figura 6.5 “Trazabilidad Enriquecida: Segundo Requerimiento de Usuario. Fuente: Elaboración Propia

Por su parte, la Figura 6.5 especifica el segundo Requerimiento de Usuario. Para satisfacerlo, se han de implementar en forma conjunta (&) los Requerimientos

Funcionales: 10 y 11. Para ello, las etiquetas textuales revelan el significado de Documento Electrónico Académico y sus componentes básicos en este contexto en particular.

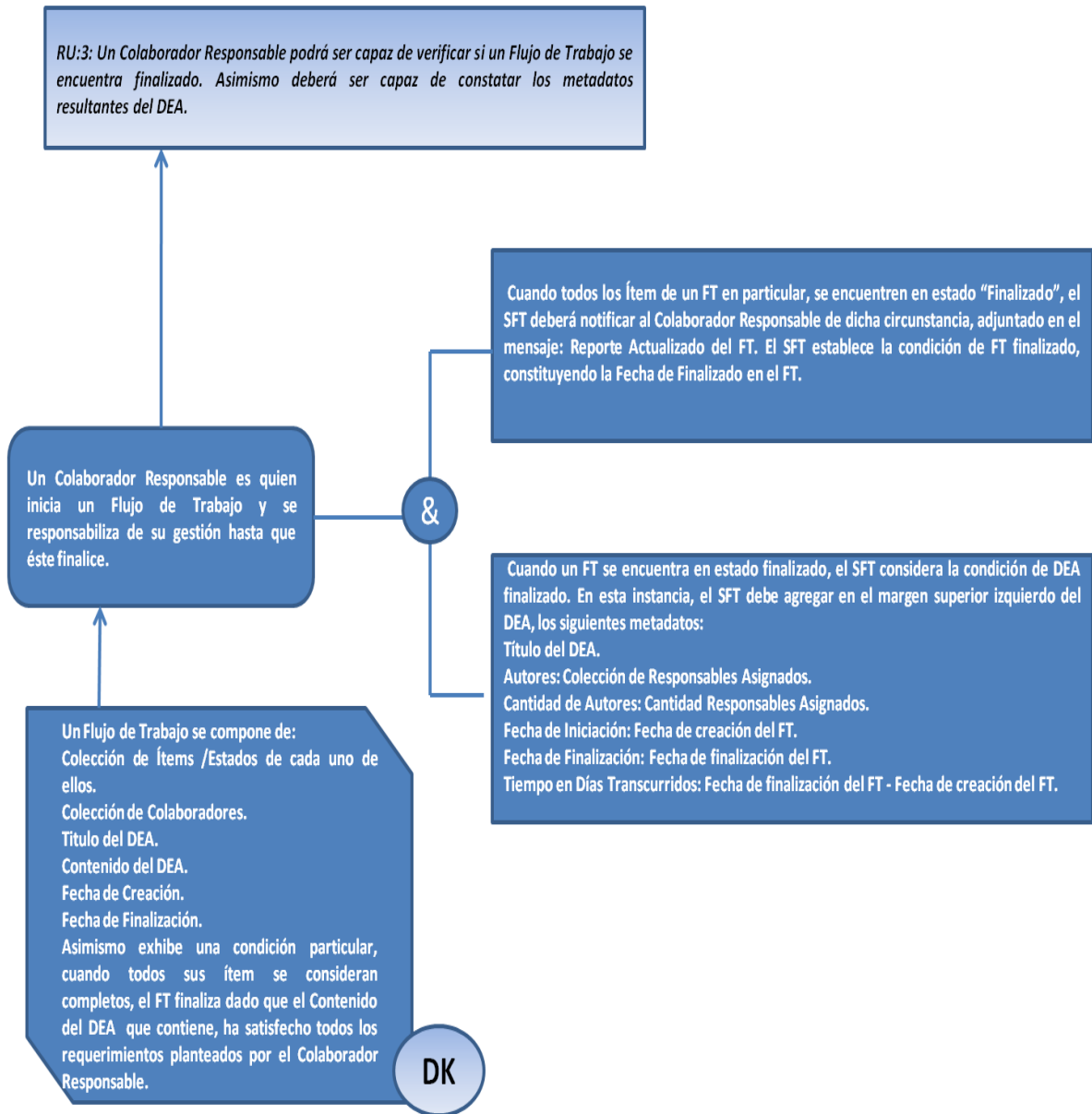


Figura 6.6 "Trazabilidad Enriquecida: Tercer Requerimiento de Usuario. Fuente: Elaboración Propia

La Figura 6.6 indica que los Requerimientos Funcionales 8 y 9 deben ser implementados en forma conjunta (&) a los efectos de satisfacer el tercer Requerimiento

de Usuario. Asimismo, se establece el significado de “Colaborador Responsable” y se describe cómo se compone un Flujo de Trabajo en este dominio.

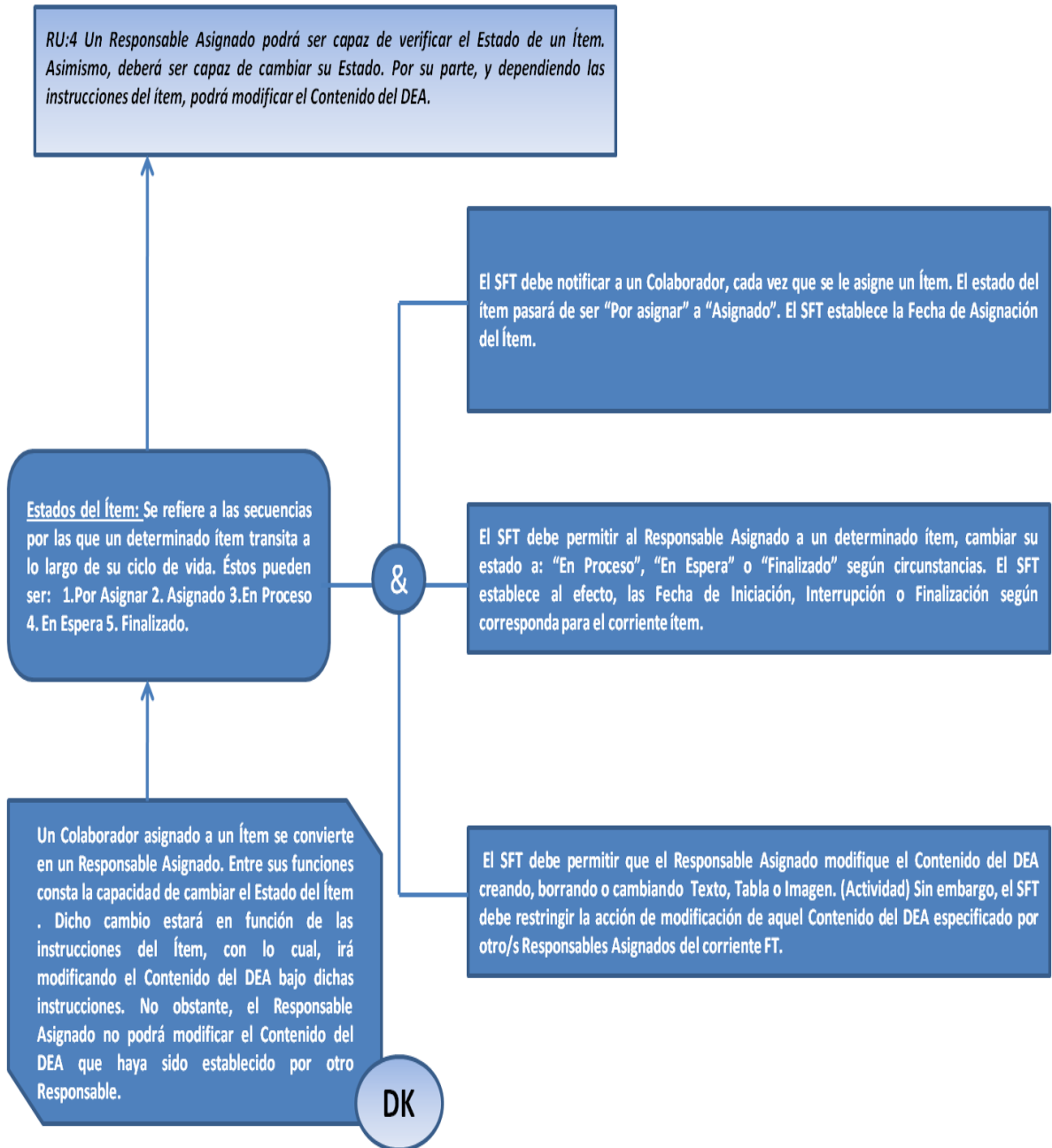


Figura 6.7 “Trazabilidad Enriquecida: Cuarto Requerimiento de Usuario. Fuente: Elaboración Propia

Por último, la Figura 6.7 indica que para satisfacer el cuarto Requerimiento de Usuario, es necesario implementar en forma conjunta (&) los Requerimientos Funcionales 2, 3 y 4. Por su parte, la información del dominio advierte respecto de la semántica que obtiene el “Estado de un Ítem” y la condición que exhibe un “Responsable Asignado”.

6.6 Particularidades del Proceso de Elicitación y Validación de los Requerimientos Funcionales del SFT.

A los efectos de establecer la presente especificación de requerimientos funcionales del Subsistema de Flujo de Trabajo (SFT), se han aplicado una serie de Técnicas de Elicitación, con el objetivo básico de comprender el dominio del problema y detectar las necesidades de los *stakeholders* del CIMEPB. Por su parte, es dable considerar que la totalidad de participantes del proyecto asciende a un total de **12 stakeholders**.

Respecto a la distribución de los *stakeholders*, se hallan 6 personas que se desempeñan en forma directa en el CIMEPB. Entre ellos, se encuentra el Director del Centro y Becarios de tipo Doctoral y Post-Doctoral en Psicología. Luego, las restantes 6 personas se desempeñan como investigadores en la Facultad de Ingeniería. Asimismo, 3 de estos últimos, serían los encargados en primera instancia de desarrollar el Producto.

6.6.1 Técnicas de Elicitación Utilizadas:

6.6.1.1 Análisis de Documentos: En base a la participación en el Proyecto de Investigación, caratulado “Normas de Producción de atributos semánticos en castellano rioplatense para un conjunto de objetos vivos y no vivos” (FONCyT-2010), se ha procedido a delinear los requerimientos de software iniciales del SIA-CIMEPB-Web. En función de dicha circunstancia, se detecta la necesidad básica de implementar un Subsistema Flujo de Trabajo que permita promover la elaboración de documentos electrónicos académicos, de acuerdo a ciertos procedimientos de trabajo especificados previamente por un determinado *stakeholder*.

6.6.1.2 Reúso de requerimientos: En base a la actividad anterior, y de acuerdo a las necesidades generales que debería satisfacer el SIA-CIMEPB-WEB, y en particular el SFT, se considera en la Especificación *MoReq2 (Model Requirements for Management of Electronic Records 2)*, la funcionalidad del módulo adicional de

Flujo de Trabajo (*Workflow*), y las recomendaciones generales efectuadas por la *Workflow Management Coalition (WfMC)*.

6.6.1.3 Entrevista de comienzo y final abierto: Se ha procedido a entrevistar tanto al Director del CIMEPB como a otros integrantes del Centro (Investigadores, Becarios, Alumnos), respecto de los objetivos generales que debería satisfacer el SIA-CIMEPB-WEB. Asimismo, y con especial atención, se ha tratado las características del Subsistema de Flujo de Trabajo, respecto de las funcionalidades básicas que éste debería ser capaz de proveer, en miras de constituir un marco de trabajo colaborativo. A partir de esta experiencia, se ha podido obtener los lineamientos generales del Subsistema en cuestión.

6.6.1.4 Análisis de Tareas: Habiendo cumplimentado las Actividades previamente señaladas, y en base a la Observación de Campo, se ha procedido a analizar las tareas de los diferentes *stakeholders* relativas a la elaboración de los Documentos Electrónicos Académicos (DEA). Se ha detectado que utilizan el correo electrónico como medio de comunicación. De esta manera, se intercambian datos, textos, diagramas, tablas entre los diferentes *stakeholders* a los fines de ir elaborando los DEA. Finalmente, se ha detectado que el medio de comunicación resulta inadecuado, dado que no permite un control de los flujos de trabajos, ni las versiones del mismo, y que dificulta en sí, las tareas de los investigadores. Se concluye que resulta necesario implementar una herramienta de software que soporte la funcionalidad demandada por los *stakeholders*.

6.6.2 Validación

El proceso de Validación, consta en chequear el conjunto de requerimientos que han sido definidos, a los efectos de descubrir eventuales omisiones, conflictos y ambigüedades. Con este proceso, se pretende asegurar que los requerimientos siguen un determinado estándar de calidad [Sommerville y Sawyer (1997)].

Si bien no es objeto de este trabajo detallar el proceso de validación utilizado, así como tampoco, enunciar y/o profundizar las técnicas utilizadas al efecto, se considera pertinente señalar que se ha presentado la lista de requerimientos funcionales a los *stakeholders*, con el fin básico, de recabar su opinión respecto *a la completitud, no ambigüedad, consistencia, minimalidad y la no redundancia de la misma*. **Entendiendo que dichas cualidades son las deseables en cualquier especificación de requerimientos, se ha obtenido en esta instancia, el beneplácito de los *stakeholders*.**

6.7 Consideraciones Finales del Capítulo

En el presente Capítulo, se ha especificado tanto los Requerimientos de Usuarios como los Requerimientos Funcionales del Subsistema de Flujo de Trabajo del CIMEPB. Asimismo, se ha implementado la Técnica de Trazabilidad Enriquecida a los efectos comunicar los requerimientos mencionados. De esta manera, queda presentado el Caso Experimental: *“Subsistema de Flujo de Trabajo del CIMEPB”*.

En el próximo Capítulo, se analizará la aplicación del **Modelo de Priorización de Requerimientos de Usuario a partir de la Segmentación de las Preferencias de los Stakeholders (MPRUSPS)**, en este caso en particular. El objetivo básico, consistirá en analizar cómo se comporta el MPRUSPS, ante un escenario en donde existan “pocos requerimientos de usuarios a evaluar”.

Capítulo 7

“Resultados de la aplicación del MPRUSPS en el Caso Experimental: Subsistema de Flujo de Trabajo del CIMEPB”

7.1 Introducción

El presente Capítulo, propone la aplicación del **Modelo de Priorización de Requerimientos de Usuario a partir de la Segmentación de Preferencias de los Stakeholders** (MPRUSPS: Capítulo 4), en el caso experimental: **“Subsistema de Flujo de Trabajo”** del Centro de Investigación en Procesos Básicos, Metodología y Educación (CIMEPB) de la Facultad de Psicología de la Universidad Nacional de Mar del Plata (Capítulo 6). En este caso, el problema consiste en determinar el orden de implementación de un conjunto reducido de requerimientos de usuarios (Sección 6.5, Capítulo 6). Asimismo, se considerará como **universo de stakeholders**, aquellos sujetos que han participado en el proceso de elicitación, especificación y validación de la herramienta en cuestión (Sección: 6.6, Capítulo 6).

Luego de la aplicación **MPRUSPS**, se espera especificar un **Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU)** que sea validado y consensado por el total de *stakeholders* consignados. De esta manera, se dispondrá de una guía básica que permitirá planificar el desarrollo del Producto de Software: **“Subsistema de Flujo de Trabajo del CIMEPB”**.

7.2 Etapas del MPRUSPS y Software a utilizar en el procesamiento de datos

A continuación, se procederá a especificar el orden de ejecución de las diferentes Etapas del MPRUSPS. Para ello se considera la Arquitectura del MPRUSPS (Sección 4.1.2, Capítulo 4). Luego, se establece:

1. El relevamiento de los conceptos claves del dominio (Sección: 4.2)
2. El relevamiento de las Producciones Semánticas de los conceptos claves del Dominio (Sección: 4.3)
3. El relevamiento de las preferencias de implementación de los Requerimientos de Usuarios (Sección 4.5).
4. Un análisis métrico de las Producciones Semántica de los conceptos claves del Dominio (Sección 4.4).
5. Una segmentación (agrupamiento) de las Preferencias de los *stakeholders* (Sección 4.6).
6. La Identificación de los diferentes Esquemas Particulares de Prioridad Implementable de Requerimientos de Usuarios (EPPIRU). (Sección 4.7).
7. La determinación del Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU). (Sección 4.8).
8. Una comparación entre el Esquema General (EGPIRU) y cada Esquema Particular (EPPIRU). Eventualmente, se analizará la Implementación de un Proceso de Negociación Grupal (Sección 4.9).

Con el objeto de relevar las diferentes producciones semánticas de los conceptos claves del dominio (Etapa 3), se utilizará la Versión 0.93 del Software **Definition Finder** (Capítulo 3. Sección: 3.3.1) desarrollado en forma conjunta por el GIA (Grupo Inteligencia Artificial, Facultad Ingeniería, Universidad Nacional de Mar del Plata) y el CIMEPB (Centro de Investigación en Procesos Básicos, Metodología y Educación, Universidad Nacional de Mar del Plata). Asimismo y a los efectos de segmentar (agrupar) las preferencias de los *stakeholders*, se aplicará la versión 7.8.0 (R2009a) del software **Matlab** junto a la librería “*SOM Toolbox*” desarrollada por Alhoiemi (et. al., 2005) bajo licencia GNU (*General Public License*).

7.3 Análisis de los Resultados.

En función de lo establecido en 7.2, se procederá a analizar los resultados obtenidos tras la aplicación de las diferentes etapas del MPRUSPS, en el caso experimental “Sistema de Flujo de Trabajo del CIMEPB” (Capítulo 6). En consecuencia, se considera:

7.3.1 El relevamiento de los conceptos claves del Dominio

En esta etapa, se pretende relevar el conjunto de conceptos, que según la percepción de los *stakeholders*, resultarían representativos a los efectos de comprender los usos e implicaciones de la futura herramienta de software en el dominio analizado. A tal efecto y considerando lo establecido en la Sección 4.2, se obtiene:

Conceptos Claves Candidatos
Software
Investigación
Trabajo Colaborativo
Interdisciplina
Herramienta de Colaboración
Documento Electrónico Académico
Flujo de Trabajo
Aplicación
Producto
Ítems
Interacción
Método

Tabla 24: “Lista de Lista Tentativa de Conceptos claves del dominio. (LTCCD) ”. Fuente: Elaboración Propia.

Como se aprecia en la Tabla 24, se dispone de la Lista Tentativa de Conceptos claves del dominio (LTCCD). Según el conjunto de *stakeholders*, dichos conceptos resultarían, en primera instancia, representativos a los efectos de comprender los usos e implicaciones de la futura herramienta de software. A los efectos de identificar los concepto claves del dominio (CCD), se procede a evaluar la distribución de los votos favorables de cada uno de los conceptos que integran LTCCD. Luego se obtiene:

Conceptos	Votos Nominales	Frecuencia Relativa (%)	Frecuencia Relativa Acumulada (%)
Software	7	18,92	18,92
Investigación	5	13,51	32,43
Trabajo Colaborativo	4	10,81	43,24
Interdisciplina	3	8,11	51,35
Herramienta de Colaboración	3	8,11	59,46
Documento Electrónico Académico	2	5,41	64,86
Flujo de Trabajo	3	8,11	72,97
Aplicación	2	5,41	78,38
Producto	2	5,41	83,78
Ítems	2	5,41	89,19
Interacción	2	5,41	94,59
Método	2	5,41	100,00
Totales	37	100,00	-

Tabla 25: "Distribución de los votos para los Conceptos Claves del Dominio". Fuente: Elaboración Propia.

En la Tabla 25, se especifica el diagrama de distribución de los votos favorables, que el conjunto de *stakeholders*, a resuelto establecer en referencia a la Lista Tentativa de Conceptos claves del dominio (LTCCD). Por su parte, las votaciones nominales (Frecuencias Absolutas) fueron ordenadas de mayor a menor y transformadas en frecuencias relativas. A partir de allí, se obtiene:

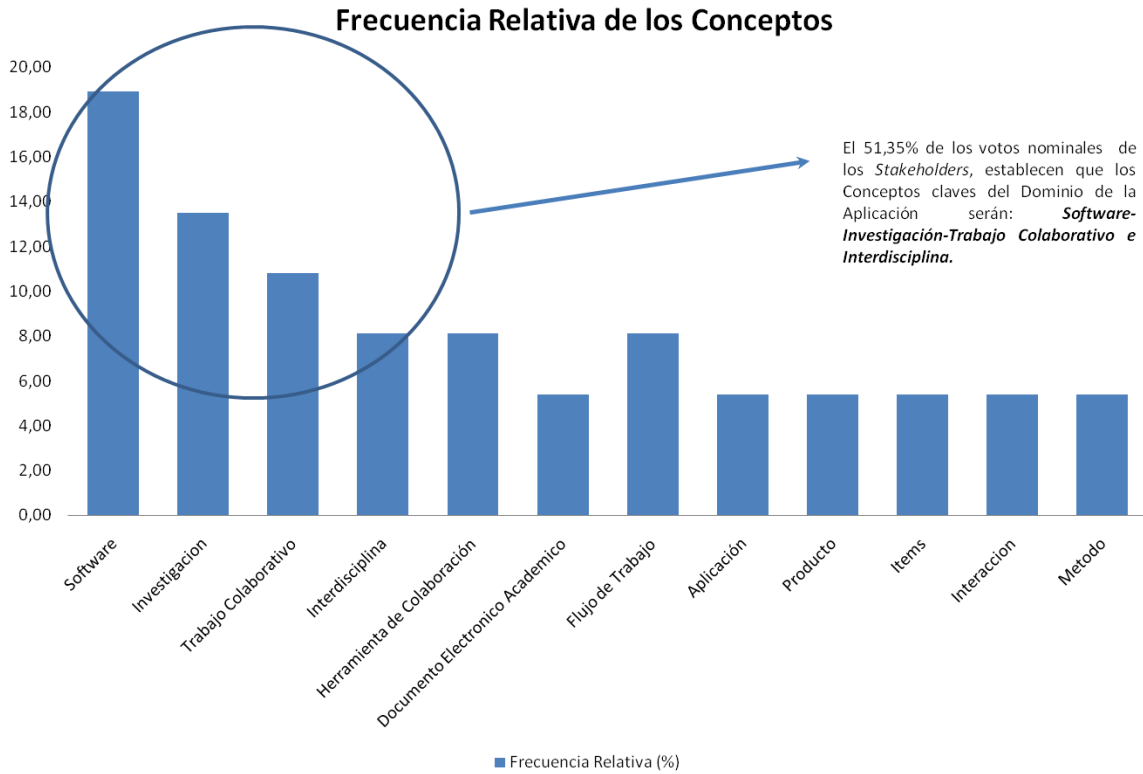


Figura 7.1: “Determinación de los Conceptos Claves del Dominio” .Fuente: Elaboración Propia.

En base a la tarea precedente, puede apreciarse en la Figura 7.1 el conjunto de Conceptos Claves del Dominio: **Software, Investigación, Trabajo Colaborativo e Interdisciplina**. Dichos conceptos, se consideran claves en este dominio, puesto que concentran la mayoría absoluta de los votos favorables totales (Sección 4.2.5, Capítulo 4).

7.3.2 El relevamiento de las Producciones Semánticas de los conceptos claves del Dominio

Considerando los conceptos claves del dominio (CCD) relevados en 7.3.1, se procederá a elicitar la producción semántica de cada uno de ellos. Para ello, se utilizará el *Definition Finder* según lo consignado en la Sección 4.3:

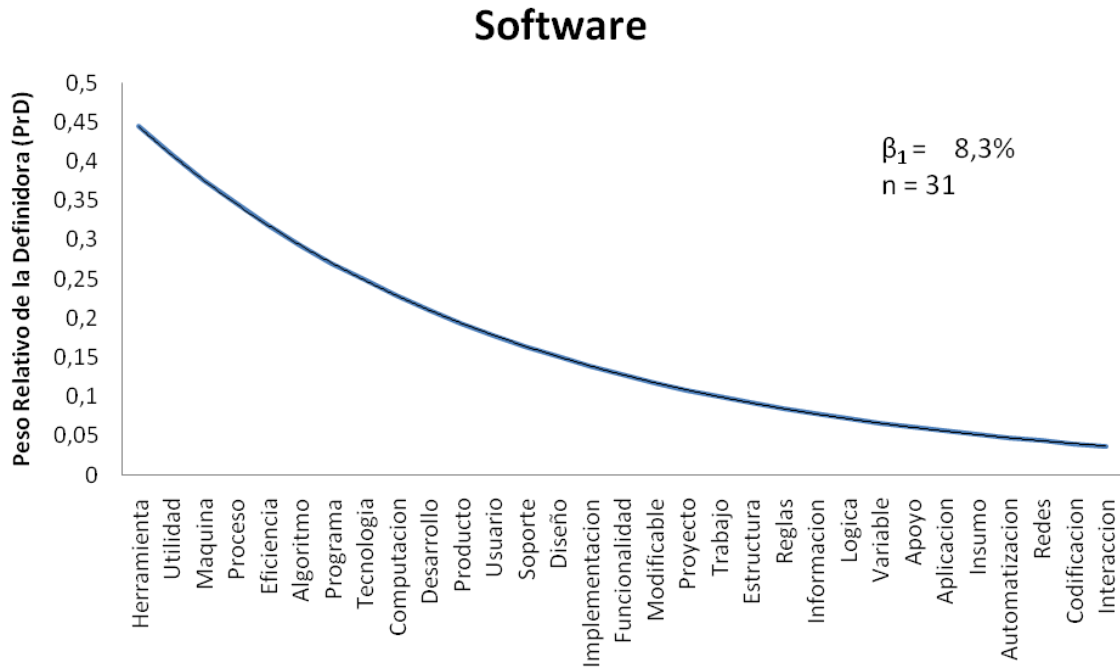


Figura 7.2: “Producción Semántica de Software” .Fuente: Elaboración Propia.

Como se aprecia en la Figura 7.2, se especifica la producción semántica social del concepto **Software**. Tal producción, ha sido relevada por el total del colectivo social de *stakeholders*. Analizando el parámetro β_1 , se podrá indicar que la tasa promedio a la cual se degrada el peso relativo de cada una de las definidoras del concepto Software, asciende al orden del 8,3 % (Efecto Distintivo. Sección 3.3.2). Por su parte, el total de definidoras socialmente consensuadas asciende a 31 (parámetro n). Dado el comportamiento que adquiere la sucesión de los pesos relativos (*Definition Finder*) de cada una de las definidoras que componen al concepto en cuestión, se podrán indicar algunos aspectos relevantes.

En efecto, se presume que para el universo de *stakeholders*, el Software hace referencia a un instrumento utilitario que beneficia una labor (**Herramienta**), y que se “identifica” porque corre o se ejecuta en una **Maquina** (supuestamente una Computadora). Sin embargo, se podrá apreciar que este colectivo social resulta ser muy heterogéneo. **Si bien existe una simplificación en la base semántica del concepto, (probablemente una visión global de tipo Usuario), también existen otros atributos inherente al concepto de Software como “eficiencia”, “modificable” o componentes que intervienen en su proceso de desarrollo y gestión como “variable”, “estructura”, “algoritmo”, “reglas” “proceso”, “proyecto” y “diseño”.**

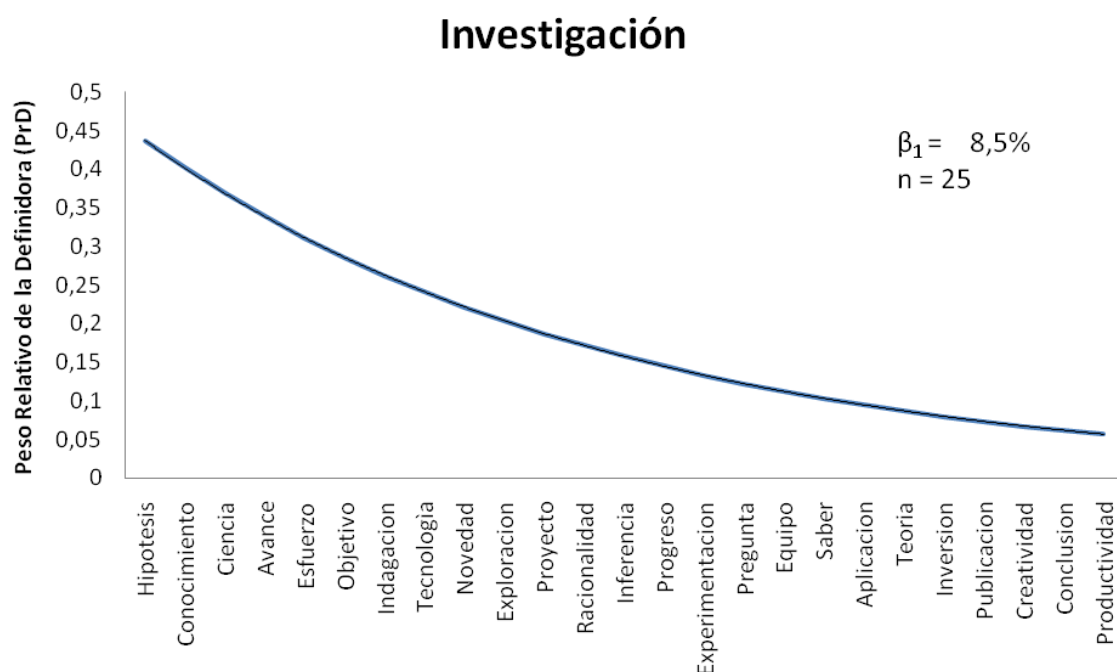


Figura 7.3: “Producción Semántica de Investigación” .Fuente: Elaboración Propia.

En la Figura 7.3, se halla la producción semántica del concepto **Investigación**. En forma similar al caso Software, el efecto distintivo en Investigación asciende al 8,5%. La cantidad de definidoras socialmente consensuadas asciende a 25. La base semántica queda establecida en los conceptos “Hipótesis” “Conocimiento” y “Ciencia” (se recuerda que la mayoría de los *stakeholders* se desempeñan como investigadores). **Sin embargo y una vez más, la producción semántica resulta muy variada en el grupo social. Por un lado, se hallan definidoras que son inherentes a la labor investigativa como “Hipótesis”,**

“Objetivos”, “Pregunta”, “Exploración” “Inferencia”, “Experimentación”, “Indagación”, “Conclusión”, “Teoría”, y otras que presumiblemente son consecuentes de dicha labor, como por ejemplo “Avance”, “Esfuerzo”, “Tecnología”, “Progreso”, “Saber”, “Inversión”, “Productividad”.

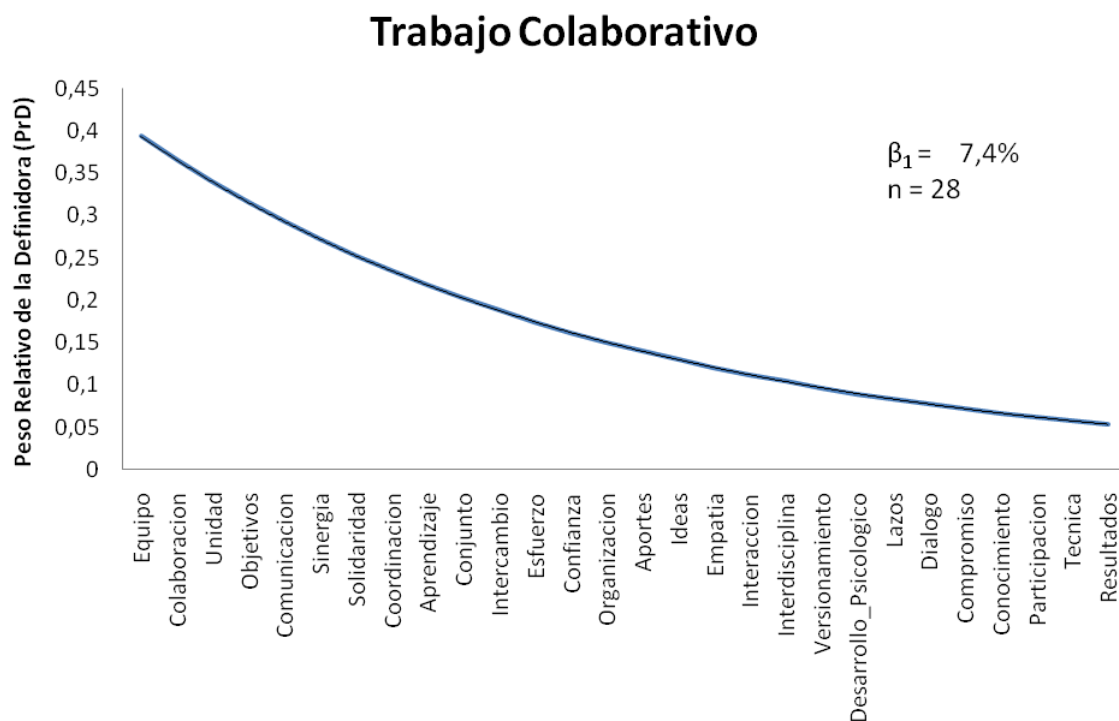


Figura 7.4: “Producción Semántica de Trabajo Colaborativo”. Fuente: Elaboración Propia.

En la Figura 7.4, se establece la producción semántica del concepto **Trabajo Colaborativo**. El efecto distintivo en Trabajo Colaborativo resulta algo menor que los casos anteriores, en efecto, asciende al 7,4%. La cantidad de definidoras socialmente consensuadas asciende a 28. Por su parte, la base semántica queda establecida en los conceptos “Equipo” “Colaboración” y “Unidad”. **Asimismo a lo largo de la producción semántica, aparecen definidoras que podrían facilitar la condición de trabajo colaborativo como ser: “Objetivos”, “Comunicación”, “Compromiso” “Coordinación”, “Compromiso”, “Confianza”, “Empatía”, “Solidaridad”, “Diálogo” “Organización” “Interacción” y definidoras que podrían ser consecuentes del Trabajo Colaborativo, por ejemplo: “Sinergia”, “Desarrollo Psicológico”, “Aprendizaje”, “Aporte”, “Ideas” , “Resultados”.**

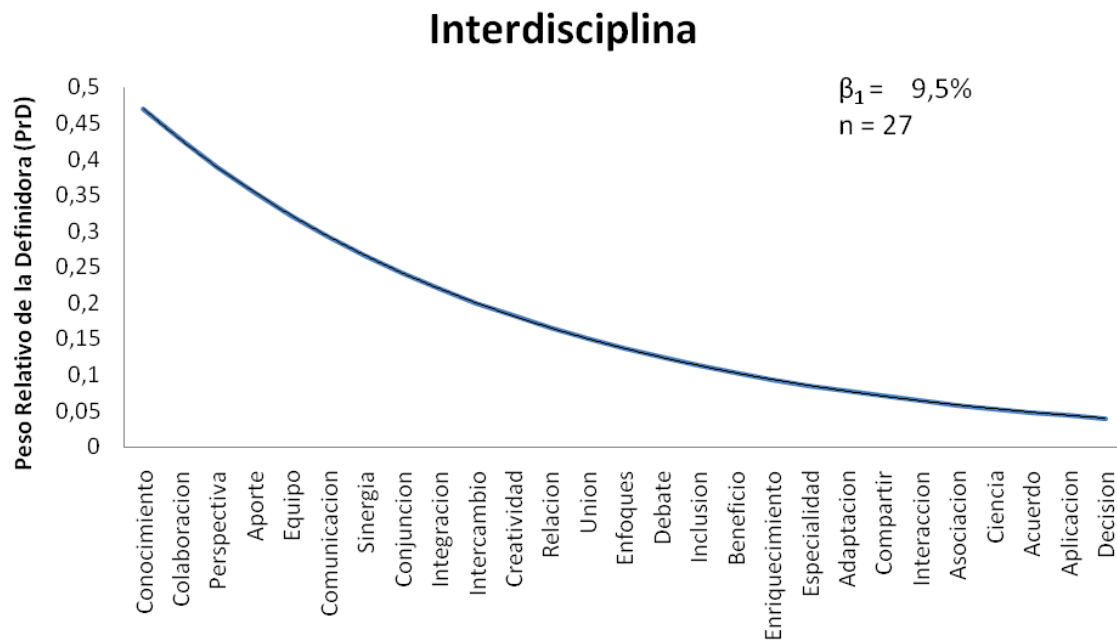


Figura 7.5: “Producción Semántica de Interdisciplina” .Fuente: Elaboración Propia.

Por último, en la Figura 7.5 se observa la producción semántica del concepto **Interdisciplina**. Lo más saliente del concepto, redundando en que su efecto distintivo es el mayor comparado con los conceptos previos, en efecto, éste asciende al 9,5%. Tal suceso acontece como consecuencia que la base semántica concentra el mayor peso relativo en la identificación del concepto. Con lo cual, la contribución del resto de las definidoras resulta relativamente menor que en los casos anteriores. Por otra parte, la cantidad de definidoras socialmente consensuadas asciende a 27. **La base semántica queda establecida en los conceptos “Conocimiento” “Colaboración” y “Perspectiva”**. Asimismo, se presume que el resto de la producción semántica se orienta a las características “dinámicas” del concepto, como ser: **“Aporte”, “Sinergia”, “Integración”, “Creatividad”, “Enriquecimiento”, “Debate”, entre otras.**

7.3.3 Análisis métrico de las Producciones Semántica de los conceptos claves del Dominio

En esta etapa, se utilizará cada sucesión de los pesos relativos establecidos para cada uno de los conceptos (obtenidos en la etapa anterior), a los efectos de propiciar el cálculo de las variables: “Valor Total de las Valorizaciones” (Capítulo 3. Sección 3.3.2) y “Razón de Exclusión” (Capítulo 3. Sección 3.3.3) según lo consignado por la Sección 4.5 del Modelo:

SPSCD		
Conceptos	VTV	RE
Software	4,91	0,09
Investigación	4,47	0,52
Trabajo Colaborativo	4,59	0,32
Interdisciplina	4,53	0,55
Promedios	4,62	0,37

Tabla 26: “Síntesis de la Producción Semántica de los Conceptos claves del Dominio según total de stakeholders. Fuente: Elaboración Propia.

En la Tabla 26, se puede apreciar que el conjunto de *stakeholders* del proyecto ha tenido una valoración similar para cada uno de los conceptos claves del dominio, ya que, cada uno de los valores de VTV oscila en forma cercana al promedio general (4.62). Sin embargo, las diferentes razones de exclusión (RE) adquieren un comportamiento diferente. En primer lugar, y en forma aproximada, cada tres identificadoras socialmente consensuadas en los conceptos del dominio, ha existido una de ellas que ha sido propuesta por un *stakeholder* en particular, sin obtener correlato alguno en el resto del grupo. ($RE_{\text{promedio}} = 0,37$). No obstante, el comportamiento es diferente tomando por separado los conceptos. Por ejemplo, cada 10 identificadoras socialmente consensuadas sólo ha existido una sin correlato para el concepto Software ($RE = 0,09$), mientras que en el caso de Interdisciplina, la proporción asciende a 2:1 aproximadamente ($RE = 0,55$).

7.3.4 Relevamiento de las preferencias de implementación de los Requerimientos de Usuarios

En esta instancia, el Ingeniero en Requerimientos deberá relevar para **cada stakeholders**, la función de preferencia de implementación de los requerimientos de usuarios. Para ello, utilizará el Método AHP (Capítulo 3. Sección: 3.2.1), considerando los pasos establecidos en la Sección 4.6:

LFDPIRU				
Stakeholders	RU ₁	RU ₂	RU ₃	RU ₄
S ₁	0,5015	0,1357	0,1457	0,2171
S ₂	0,149	0,2062	0,2265	0,4183
S ₃	0,39	0,09	0,15	0,37
S₄	0,462	0,061	0,123	0,3525
S ₅	0,4256	0,06255	0,177	0,3348
S ₆	0,56	0,03	0,13	0,28
S₇	0,57	0,047	0,166	0,217
S ₈	0,2	0,21	0,25	0,34
S ₉	0,18	0,215	0,25	0,355
S ₁₀	0,6125	0,043	0,088	0,2565
S₁₁	0,096	0,23	0,2855	0,3885
S ₁₂	0,595	0,0552	0,1186	0,2312

Tabla 27: "Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios". Fuente: Elaboración Propia.
En negrita: distribución final luego de 1 interacción por inconsistencia.

Como se puede apreciar en la Tabla 27, se dispone de la Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios (LFDPIRU). Dado un nivel de inconsistencia inicial significativamente elevado en *los stakeholders 4, 7 y 11* (CR = 1,12, CR = 2,11 y CR = 0,89 respectivamente), se ha procedido a reformular los juicios emitido de los *stakeholders* señalados, de manera tal, de lograr en cada uno de

ellos, un nivel de inconsistencia menor al 10 % en su juicio de valoración (según lo establecido en la Sección: 4.5.4).

7.3.5 Segmentación (agrupamiento) de las Preferencias de los stakeholders

Una vez obtenida la Lista de Funciones de Distribución de Preferencia de Implementación de Requerimientos de Usuarios (LFDPIRU etapa anterior), el objetivo siguiente consistirá en segmentar (agrupar) a los *stakeholders*, de acuerdo al grado de similitud existente entre sus respectivos vectores de preferencias de implementación de requerimientos de usuarios. A tal efecto, se utilizará el método: Mapas auto-organizados de Kohonen (Capítulo 3. Sección: 3.4.2), según lo consignado por la Sección 4.6. De esta manera, se obtiene:

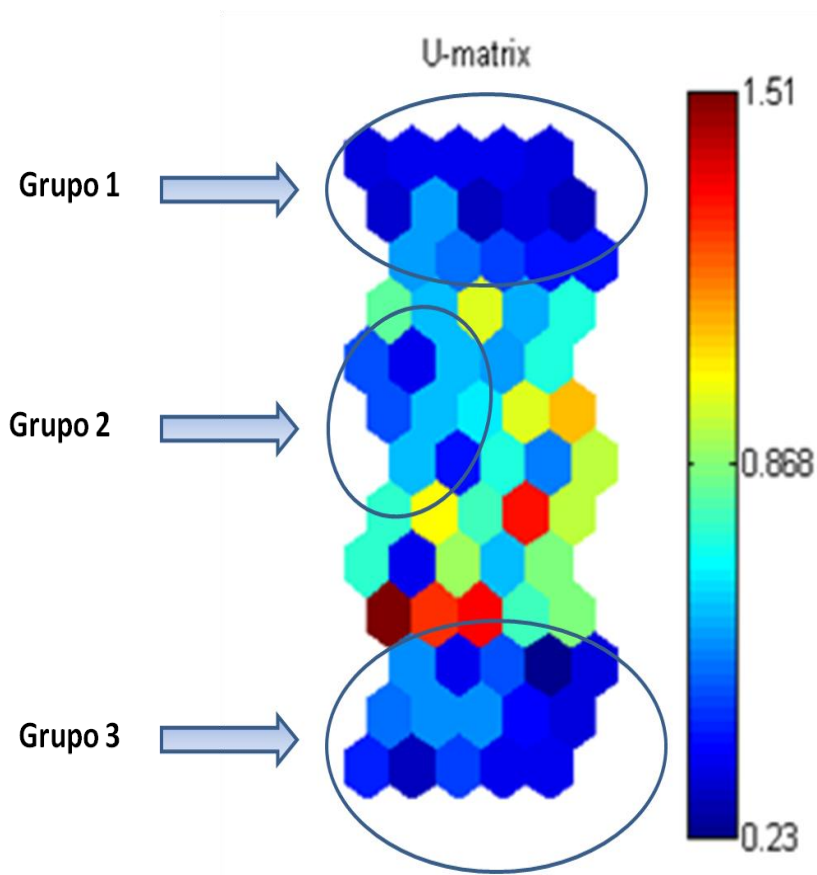


Figura 7.6 "Matriz de Distancia Unificada". Fuente: Elaboración Propia en base a resultados obtenidos en Matlab.

En la Figura 7.6, se especifica la Matriz de Distancia Unificada (*U-Matrix*). La misma, surge a partir del entrenamiento propuesto en la red de Kohonen (SOM). Asimismo, y desde la geometría subyacente de los datos, inicialmente se podría constatar la existencia

de 3 grupos de *stakeholders*. Cada uno de ellos, estaría formado por un subconjunto de *stakeholders* que exhiben un cierto patrón de preferencia en la implementación de los requerimientos de usuarios evaluados. La localización de cada uno de los grupos en la grilla, estaría delimitada por las regiones: “Superior” (Grupo1), “Central con sesgo Izquierdo” (Grupo 2), Inferior (Grupo 3). En la Figura 7.7, se identifican los rasgos característicos de cada uno de estos grupos.

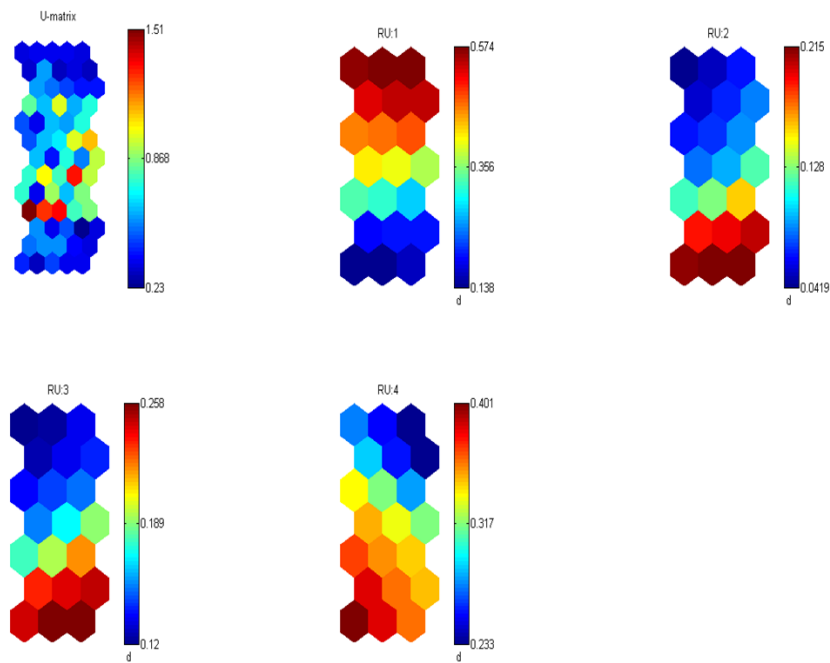


Figura 7.7 “Identificación de grupos y rasgos característicos”. Fuente: Elaboración Propia en base a resultados obtenidos en Matlab.

Como se puede apreciar en la Figura 7.7, se halla nuevamente la Matriz de Distancia Unificada y una serie de grillas que se corresponden a cada una de las variables de entrada del SOM (Requerimientos de Usuarios). A partir del análisis conjunto de estas grillas, se podrá evaluar la configuración que adquiere cada uno de los grupos hallados (Figura 7.6), en función de las preferencias que los *stakeholders* han relevados para cada uno de los Requerimientos de Usuarios evaluados. De esta manera, se obtiene:

Grupo 1: Localizado en la Sección Superior de la Matriz U. Caracterizado por una fuerte valorización del primer Requerimiento de Usuario (R.U : 1), el cual versa: “Un Colaborador Responsable podrá ser capaz de crear Flujos de Trabajo y/o monitorear el progreso del Flujo de Trabajo que haya creado o en el que participe y/o consultar el listado de DEAs

finalizados". Asimismo, el nivel de preferencia juzgado por los *stakeholders* clasificado en este clúster, advierten una puntuación de más del 55% para este requerimiento en particular. (Ver en grafico "RU:1" de la Figura 7.7, la fuerte tonalidad en Rojo en su Sección Superior). Siguiendo en el orden de preferencia, se halla el cuarto Requerimiento de Usuario (R.U: 4), el cual establece: *"Un Responsable Asignado podrá ser capaz de verificar el Estado de un Ítem. Asimismo podrá ser capaz de cambiar su Estado. Por su parte, y dependiendo de las instrucciones del Ítem, podrá modificar el contenido del DEA"*. Respecto al nivel de preferencia, el grupo de *stakeholders* ha ponderado a dicho Requerimiento con una importancia relativa superior al 20% (Ver en grafico "RU:4" de la Figura 7.7, matices del color "azul" en la región Superior). En tercer y cuarto lugar el grupo ha valorado en orden de importancia los requerimientos de usuario 3 y 2 respectivamente (RU: 3 y RU: 2). El nivel de preferencia del RU: 3 oscila en promedio entre el 12 y 16% (Ver en grafico "RU:3" de la Figura 7.7, la matices del color "azul" en la región Superior) y en RU:2 se localiza en torno al 4 % y 8 % (Ver en grafico "RU:2" de la Figura 7.7, matices del color "azul" en la región Superior).

Grupo 2: Localizado en la Sección Central con sesgo Izquierdo en la Matriz U. El grupo de *stakeholders* clasificado en este clúster, mantiene el mismo orden de preferencia de los Requerimientos de Usuario que en el caso anterior. Es decir, en primer lugar localiza al RU:1, luego RU:4, en tercer lugar RU3 y por último RU2. Sin embargo, la diferencia clave entre los Grupos 1 y 2, se halla porque este último grupo, ha bajado significativamente el orden de prioridad de RU1 y ha volcado dicha diferencia en el RU4, manteniendo en efecto, similar nivel de preferencia en los Requerimientos de Usuario 3 y 2. En efecto RU:1 adquiere un nivel de preferencia con un promedio aproximado en el orden del 42 % (frente al 55 % en el Grupo 1), RU:4 el 35 % (frente al 23 % en el Grupo 2), RU:3 el 15 % (similar al análogo anterior) y por último RU:2 con el 7 % (similar al análogo anterior).

Grupo 3: Localizado en la Sección Inferior con sesgo derecho en la Matriz U. El presente grupo de *stakeholders* cambia significativamente su patrón de preferencia entre los Requerimientos de Usuarios disponibles. En efecto, será el RU: 4 quien detente la mayor valorización relativa entre los requerimientos, alcanzando un promedio de preferencia cercano al 37 % (ver en "RU:4" en la Figura 7.7 la Sección Inferior con sesgo derecho, matices de color entre naranja y rojo, en un rango de variabilidad que se extiende entre el 31,7 y 40%). Siguiendo con el orden de preferencia, se halla el requerimiento de usuario número 3, el cual establece que *"Un Colaborador Responsable podrá ser capaz de verificar si un flujo de trabajo se encuentra finalizado. Asimismo deberá ser capaz de constatar los metadatos resultantes del DEA"* con un promedio aproximado del 25 % (ver en "RU:3" en la Figura 7.7, la Sección Inferior con sesgo derecho, matices de color entre rojo y marrón,

en un rango de variabilidad que se extiende entre el 18,9 y 25.8%). En tercer lugar se encuentra el requerimiento de usuario número 2, el cual indica que: *“Un Colaborador podrá ser capaz de consultar el listado DEAs finalizados”*, con un promedio aproximado del orden del 20 % de preferencia (ver en “RU:2” en la Figura 7.7, la Sección Inferior con sesgo derecho, matices de color fuertemente marrón, en un rango de variabilidad que se extiende entre el 12,8 y 21.5%). En cuarto y último lugar, se encuentra el Requerimiento de Usuario número 1, con un promedio aproximado de preferencia asignada del orden del 16 %. (ver en “RU:1” en la Figura 7.7, la Sección Inferior con sesgo derecho, matices de color fuertemente azul, en un rango de variabilidad que se extiende entre el 13,8 y 35.6%).

Por su parte, y con el objeto de validar lo que sugiere una simple inspección visual de la Matriz Distancia Unificada (Figura 7.6 y 7.7), se utilizará el índice de Davies-Bouldin (DB). Tal cual se indicó en la Sección 3.4.3 del Capítulo 3, dicho índice permite evaluar la bondad o calidad de los agrupamientos hallados. Luego, dicho índice busca minimizar por un lado, la distancia entre los elementos que componen un determinado grupo, y por el otro, promueve que las configuraciones de los grupos resultantes maximicen la distancia entre ellos.

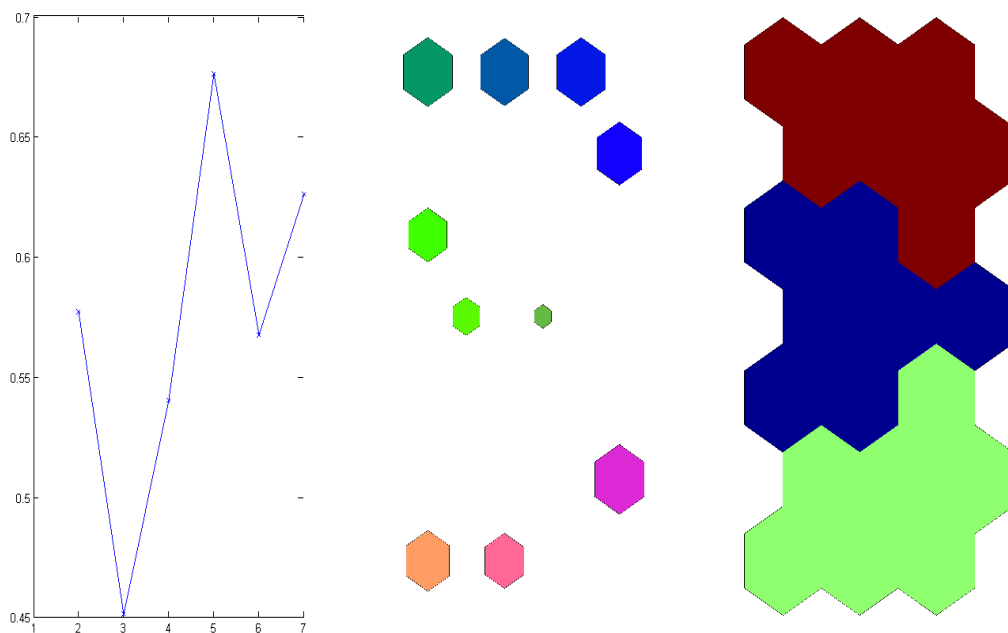


Figura 7.8 “Análisis de Grupos mediante el Índice Davies-Bouldin. Fuente: Elaboración Propia en base a resultados obtenidos en Matlab.

Como se aprecia en la Figura 7.8, claramente puede confirmarse la existencia de tres grupos. En el gráfico de la izquierda de la Figura, puede determinarse que el valor que minimiza el índice asciende a 0,45. Luego, dicha cifra resulta adecuada. En los gráficos del centro y la derecha, se propone una configuración de grupos tal que la grilla queda particionada en los sectores: **Superior, Centro con sesgo izquierdo, e Inferior**. En efecto, dicha configuración también ha sido divisada en la Matriz de distancia Unificada de la Figura 7.6.

Validada la existencia de 3 grupos de *stakeholders*, se procede a evaluar cómo se componen cada uno de ellos. Para ello se analizará la Figura 7.9:

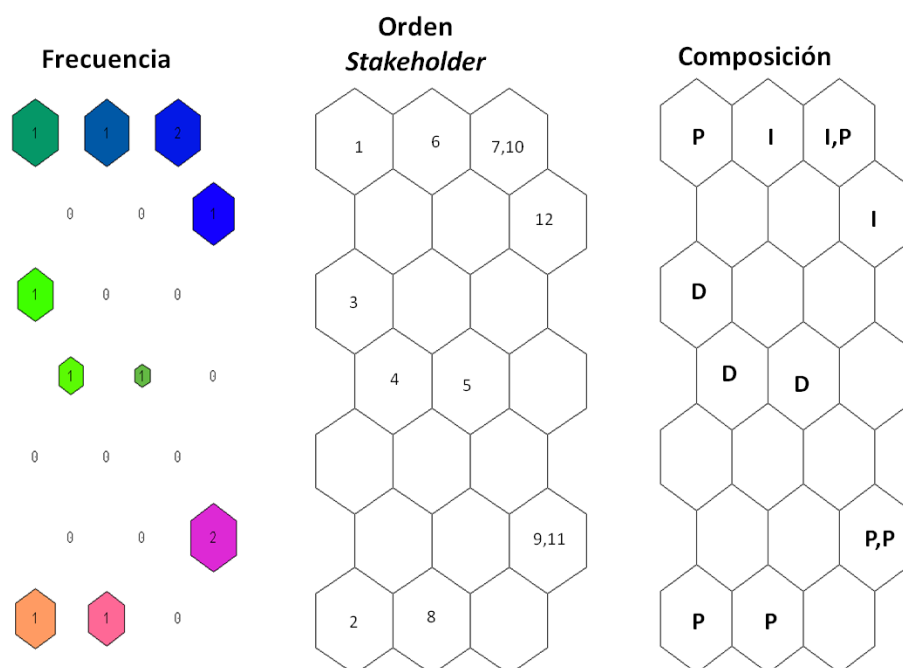


Figura 7.9 "Composición de los Grupos". Fuente: Elaboración Propia , en base a resultados Matlab.

En la presente Figura 7.9 se pueden apreciar los gráficos "Frecuencia", "Orden de Stakeholder" (que hace referencia al orden especificado en la Tabla 27) y "Composición". En el gráfico "Frecuencia" se establece la cantidad de *stakeholders* que se clasifican en cada una de las celdas en función de su patrón de preferencia de prioridades. De esta

manera, se podrá inferir que el Grupo 1 se compone de 5 *stakeholders* (ver en “Frecuencia” Sección: Superior y “Orden de *Stakeholder*”: 1, 6, 7, 10, 12) de los cuales 3 se desempeñan en la Facultad de Ingeniería y 2 en el CIMEPB (ver en “Composición” Sección: Superior: P-I-I-P-I). Luego, el Grupo 2 se compone de 3 *stakeholders* (ver en “Frecuencia” Sección: Central con sesgo Izquierdo y “Orden de *Stakeholder*”: 3, 4, 5) que se desempeñaran como los Desarrolladores de la Aplicación (ver en “Composición” Sección: Central con sesgo Izquierdo: D-D-D). Finalmente, el Grupo 3 se compone de 4 *stakeholders* (ver en “Frecuencia” Sección: Inferior con sesgo derecho y “Orden de *Stakeholder*”: 2, 8, 9, 11) quienes se desempeñan en su totalidad en el CIMEPB (ver en “Composición” Sección: Inferior con sesgo derecho: P-P-P-P).Luego, y en base a los resultados provistos en esta etapa, se tabularán los datos obtenidos en la Tabla: Grupos y Composición (Tabla 28):

Grupos y Composición (GyC)				
Grupo 1				
Stakeholders	RU₁	RU₂	RU₃	RU₄
S₁	0,5015	0,1357	0,1457	0,2171
S₆	0,56	0,03	0,13	0,28
S₇	0,57	0,047	0,166	0,217
S₁₀	0,6125	0,043	0,088	0,2565
S₁₂	0,595	0,0552	0,1186	0,2312
Grupo 2				
Stakeholders	RU₁	RU₂	RU₃	RU₄
S₃	0,39	0,09	0,15	0,37
S₄	0,462	0,061	0,123	0,3525
S₅	0,4256	0,06255	0,177	0,3348
Grupo 3				
Stakeholders	RU₁	RU₂	RU₃	RU₄
S₂	0,149	0,2062	0,2265	0,4183
S₈	0,2	0,21	0,25	0,34
S₉	0,18	0,215	0,25	0,355
S₁₁	0,096	0,23	0,2855	0,3885

Tabla 28: “Tabla de Grupos y Composición (GyC)”. Fuente: Elaboración propia.

7.3.6 Determinación de los diferentes Esquemas Particulares de Prioridad Implementable de Requerimientos de Usuarios (EPPIRU)

Una vez efectuado el proceso de segmentación de las preferencias de los *stakeholders* (7.3.5), se establecerá el valor medio de preferencia de cada uno de los Requerimientos Usuarios valorados en cada uno de los Grupos identificados. Para ello, el Ingeniero en Requerimientos deberá seguir las recomendaciones establecidas en la Sección 4.7. Luego, se obtiene:

LGVMP_{RU}				
Grupo	RU ₁	RU ₂	RU ₃	RU ₄
1	0,5678	0,06218	0,12966	0,24036
2	0,4258	0,07118	0,15	0,3524
3	0,15625	0,2153	0,253	0,37545

Tabla 29: "Lista de Grupos con sus correspondientes Valores Medio de Preferencia de Implementación de Requerimientos de Usuarios (LGVMP_{RU})". Fuente: Elaboración Propia

A partir de los resultados obtenidos en la Tabla 29, se establecerán los respectivos Esquemas Particulares de Prioridad Implementable de Requerimientos de Usuarios (EPPIRU) en cada uno de los Grupos. Luego y considerando el resto de las recomendaciones efectuadas en la Sección 4.7, se obtiene:

$$EPPIRU_{Grupo:1} = \{ [1^{\circ}(RU_1); 2^{\circ}(RU_4); 3^{\circ}(RU_3); 4^{\circ}(RU_2)] \}$$

$$EPPIRU_{Grupo:2} = \{ [1^{\circ}(RU_1); 2^{\circ}(RU_4); 3^{\circ}(RU_3); 4^{\circ}(RU_2)] \}$$

$$EPPIRU_{Grupo:3} = \{ [1^{\circ}(RU_4); 2^{\circ}(RU_3); 3^{\circ}(RU_2); 4^{\circ}(RU_1)] \}$$

7.3.7 Determinación del Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU)

A continuación, se procederá a calcular el Valor Medio de Prioridad Implementable de los Requerimientos de Usuarios para el total de Grupos. Para ello, el Ingeniero en Requerimientos deberá confeccionar, previamente, la *Síntesis de las Producciones Semánticas de los conceptos claves del dominio por Grupo (SPSCCD)*. Para tal fin, seguirá las recomendaciones efectuadas en la Sección 4.8. De esta manera, obtiene:

SPSCCDG		
<i>Grupo</i>	\overline{VTV}	\overline{RE}
1	4,69	0,19
2	3,61	0,75
3	2,66	2

Tabla 30: "Síntesis de las Producciones Semánticas de los conceptos claves del dominio por Grupo (SPSCCD)". Fuente Elaboración Propia.

Como se aprecia en la Tabla 30, se ha calculado los valores promedio de **VTV** (Valor Total de las Valorizaciones, Sección: 3.4.2) y **RE** (Razón de Exclusión, Sección 3.4.3) para la totalidad de los conceptos claves del dominio, según la percepción de cada uno de los grupos identificados. El Grupo 1, ha obtenido el mayor desempeño socio-cognitivo en relación a los conceptos claves del dominio, puesto que, ha obtenido un nivel promedio de VTV más alto en comparación al resto de los grupos (4.69), y en forma simultánea, la menor razón de exclusión de definidoras (0.19). Con lo cual, se podrá indicar que los *stakeholders* aglomerados en el Grupo 1 han obtenido una producción semántica de los conceptos claves del dominio, más articulada y consensuada sí se compara con lo registrado para los Grupos 2 y 3.

En segundo lugar, se encontraría el Grupo 2 con un valor promedio de VTV de 3,61 y una RE del orden de 0.75. Por último, el Grupo 3 obtiene el menor consenso en la producción semántica de los conceptos. Se deduce tal cuestión, producto del menor valor que adquiere la VTV promedio (2,66) y la mayor razón de exclusión promedio de las definidoras (2). Esto último merece un comentario especial. Resulta que por cada definidora consensuada socialmente, cada *stakeholder* aglomerado en el Grupo 3, ha

aportado 2 Definidoras que no han tenido correlato o representación semántica afín en el grupo en cuestión. Dicha proporción, resulta altísima comparado con lo registrado en los Grupos 2 y 3. En efecto, para la aparición de dos definidoras excluidas, se necesitaron aproximadamente 3 consensuadas socialmente en el Grupo 2, y 10 en el Grupo 1.

Finalmente, y considerando no sólo estos últimos resultados (Tabla 30), sino que también, la Lista de Grupos con sus correspondientes Valores Medio de Preferencia de Implementación de Requerimientos de Usuario (Tabla 29), se calculará el Valor Medio de Prioridad Implementable de Requerimientos de Usuarios para el total de grupos ($VMP_{RU_{Total}}$). En efecto, siguiendo las recomendaciones de la Sección 4.8 del Capítulo 4, se obtiene:

$$VMP_{RU_{Total}} = \{[0,421; 0,1023; 0,1663; 0,31]\}$$

De esta manera, el $VMP_{RU_{Total}}$ hace mención a un vector de distribución de preferencias de implementación de los Requerimientos de Usuarios considerando el total de los grupos. Luego, ordenando el vector según lo establecido por la misma sección (Sección 4.8), se obtiene el **Esquema General de Prioridad Implementable de Requerimientos de Usuario (EGPIRU)**:

$$EGPIRU = \{[1^{\circ}(RU_1); 2^{\circ}(RU_4); 3^{\circ}(RU_3); 4^{\circ}(RU_2)]\}$$

7.3.8 Comparación entre el Esquema General (EGPIRU) y cada Esquema Particular (EPPIRU). Análisis de la Implementación de un Proceso de Negociación Focal

En esta sección, se determinará el Grado de Alineamiento que ha adquirido cada uno de los EPPIRU (determinados en la Sección 7.3.5) en relación al EGPIRU (Sección 7.3.7). Para ello, se utilizará el sistema de la memoria asociativa difusa (FAM) especificado en la Sección 4.9 (Capítulo 4).

LGCCA						
Grupo	Orden de Corte	NPG	GCG	GAG(<i>defuzzifier</i>)	Categoría Alineamiento (GAG)	Resultado de la Comparación (RC)
1	2	1,1	1	0,8	Aceptable	Bastante Perfecto
2	2	1,06	1	0,8	Aceptable	Bastante Perfecto
3	2	0,73	0,25	0,4	Moderado	Más Imperfecto que Perfecto

Tabla 31: "Lista de Grupos Clasificados según Categoría de Alineamiento (Caso Real)". Fuente: Elaboración Propia

En la Tabla 31, se establece la Lista de Grupos Clasificados según su Categoría de Alineamiento (LGCCA) para este caso en particular. El Grupo 1 ha obtenido un Nivel de Preferencia "**Sobre-Valorado**" (NPG: 1,1) y un grado de coincidencia "**Alto**" en comparación al EGIPIRU (GCG: 1). En consecuencia, su Grado de Alineamiento es "**Aceptable**" ($GAG_{(d)} = 0,8$) y su esquema de prioridad (EPPIRU) resulta "**Bastante Perfecto**" comparado con el EGPIRU. Asimismo, el Grupo 2 adquiere igual comportamiento.

Por su parte el Grupo 3 ha obtenido un Nivel de Preferencia "**Sub-Valorado**" (NPG: 0,73) y un grado de coincidencia "**Bajo**" en comparación al EGIPIRU (GCG: 0,41). En consecuencia, su Grado de Alineamiento es "Moderado" ($GAG_{(d)} = 0,4$) y su esquema de prioridad (EPPIRU) resulta "**Más Imperfecto que Perfecto**" comparado con el EGPIRU.

Dado los resultados obtenidos por el Grupo 3, y considerando las recomendaciones establecidas en la Sección 4.9 del Capítulo 4, se deberá implementar para dicho grupo el

proceso de negociación especificado en la Sección 3.5 del Capítulo 3. En consecuencia y según lo establecido en la Etapa de Pre-Negociación del Proceso, se obtiene:

Síntesis del Grupo 3					
Stakeholders	RU ₁	RU ₂	RU ₃	RU ₄	Función
S ₂	0,149	0,2062	0,2265	0,4183	P
S ₈	0,2	0,21	0,25	0,34	P
S ₉	0,18	0,215	0,25	0,355	P
S ₁₁	0,096	0,23	0,2855	0,3885	P
$VMP_{cluster}^{RU_{III}} [0,1562; 0,2153; 0,253; 0,37545]$				$VMP_{RU_{Total}} = \{[0,421; 0,1023; 0,1663; 0,31]$	
$EPPIRU_{III} [1^{\circ}(RU_4); 2^{\circ}(RU_3); 3^{\circ}(RU_2); 4^{\circ}(RU_1)]$				$EGPIRU[1^{\circ}(RU_1); 2^{\circ}(RU_4); 3^{\circ}(RU_3); 4^{\circ}(RU_2)]$	
$\overline{VTV}_{III} = 2,66; \overline{RE}_{III} = 2;$				$\overline{VTV}_{General} = 4,63; \overline{RE}_{General} = 0,37;$	

Tabla 32: "Síntesis de datos del Grupo 3". Fuente: Elaboración Propia.

La Tabla 32, ofrece una síntesis de los datos que le corresponden al Grupo 3. El grupo en cuestión, se compone de 4 *stakeholders* que se desempeñan como investigadores en el CIMEPB de la Facultad de Psicología (P: Psicólogos). Respecto al desempeño obtenido en la producción semántica de los conceptos claves del dominio, se podrá indicar que el promedio VTV resulta el más bajo entre los grupos identificados (Tabla 30) y en consecuencia, más bajo que a Nivel General (Tabla 26). Por su parte, la razón de exclusión promedio excede significativamente los niveles determinados a nivel general (RE: 0,37). Comparando las funciones de distribución de los valores medio de preferencia de implementación de los requerimientos de usuarios, tanto del grupo como a

nivel general, se podrá apreciar que las mismas exhiben comportamientos un tanto similares. Al respecto se analizará la Figura 7.10:

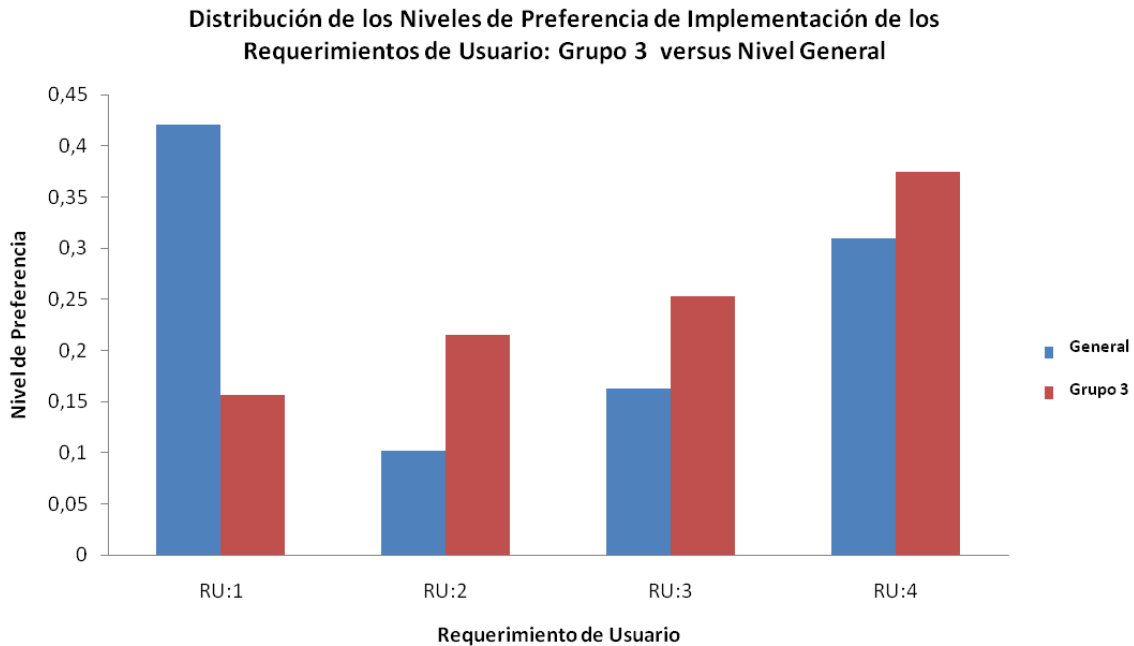


Figura 7.10: “Funciones de distribución de preferencia de implementación: Grupo 3 versus Nivel General”. Fuente: Elaboración Propia.

Como se observa en la Figura 7.10, el Grupo 3 establece una función de distribución de preferencias que resulta ascendente a partir del Requerimiento de Usuario: 1 (el cual se le asigna un mínimo de preferencia). Luego, asigna el valor máximo de preferencia al Requerimiento de Usuario: 4. Por el contrario, la función de distribución de preferencias a nivel general, asigna un máximo de preferencia al Requerimiento de Usuario 1, muestra un mínimo de valorización en el Requerimiento de Usuario 2, y asciende su nivel de preferencia a partir del Requerimiento de Usuario: 3. No obstante a ello, se puede apreciar que a partir del Requerimiento de Usuario 2 el Grupo 3 exhibe el mismo comportamiento que el acontecido a nivel general. Es decir, aplica una valorización creciente a partir del mencionado requerimiento.

Si bien el Grado de Alineamiento del grupo es “Moderado” en relación al EGPIRU, resultaría factible establecer un nivel “Aceptable” en dicha variable, sí los miembros del

Grupo en cuestión, comprenden que han *subvalorado* comparativamente el Requerimiento de Usuario 1.

En base al análisis precedente, se ha informado a los miembros del Grupo 3 que su esquema de priorización resultó bastante similar comparado con el Nivel General. Sin embargo, dada la subvaloración relativa del Requerimiento de Usuario 1, han constituido un esquema que no lo contempla como el primero implementable como sí es el caso a Nivel General. Por tal motivo, se les ha solicitado a los miembros del Grupo que se pronuncien a favor o en contra respecto de la posibilidad de implementar el EGPIRU, como guía base para desarrollar el plan de implementación del “Subsistema de Flujo de Trabajo”. Fundamentando al efecto, que el EGPIRU ha sido constituido por la elección de la mayoría de los *stakeholders* (8 sobre 12). **Finalmente, habiendo relevado las diferentes opiniones de los miembros del grupo, no se ha detectado objeción alguna respecto de la implementación del EGPIRU.**

Como consecuencia, el Equipo de Desarrollo deberá implementar los Requerimientos Funcionales contenidos en cada uno de los Requerimientos de Usuarios, considerando el siguiente orden:

1° *“Un Colaborador Responsable podrá ser capaz de crear Flujos de Trabajo y/o monitorear el progreso del Flujo de Trabajo que haya creado o en el que participe y/o consultar el listado de DEAs finalizados” (RU:1).*

2° *“Un Responsable Asignado podrá ser capaz de verificar el Estado de un Ítem. Asimismo podrá ser capaz de cambiar su Estado. Por su parte, y dependiendo de las instrucciones del Ítem, podrá modificar el contenido del DEA” (RU:4)*

3° *“Un Colaborador Responsable podrá ser capaz de verificar si un flujo de trabajo se encuentra finalizado. Asimismo deberá ser capaz de constatar los metadatos resultantes del DEA” (RU:3)*

4° *“Un Colaborador podrá ser capaz de consultar el listado DEAs finalizados” (RU:2)*

Finalmente, el presente EGPIRU se postula como una guía útil para la determinación del orden de implementación de los requerimientos de usuarios pertenecientes al proyecto: “Subsistema de Flujo de Trabajo del CIMEPB”. Luego, la característica más saliente del esquema, es que ha sido determinado por la totalidad de los stakeholders y de una manera explícitamente consensuada.

7.4 Consideraciones finales del Capitulo

El presente Capitulo, ha propuesto la aplicación del MPRUSPS en el caso de experimentación real: “Subsistema de Flujo de Trabajo del CIMEPB”. Considerando la necesidad de ordenar los Requerimientos de Usuarios de acuerdo a las preferencias de los *stakeholders* del Proyecto, ha sido factible la determinación de un Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU) explícitamente consensuado por el total de los participantes.

Asimismo, ha sido sumamente significativo segmentar las preferencias de los *stakeholders* (agruparlas), a los efectos de constatar el comportamiento que ha adquirido la preferencia relevada en cada grupo (EPPIRU). De esta manera, se ha podido identificar aquel segmento de *stakeholders* (EPPIRU del Grupo 3), cuyo patrón de preferencia, ha diferido de aquel constituido a nivel general. A partir de este último análisis, ha sido factible la aplicación de un proceso de negociación focal, que ha considerado los factores por los cuales dicho grupo de preferencia, se ha alejado del esquema general. Gracias a ello, se ha satisfecho el propósito de diseñar un EGPIRU consensuado por el colectivo de *stakeholders*, lo que redundará en menores chances de conflictos futuros, asociados al orden de implementación de los requerimientos de usuarios.

Capítulo 8

Discusión general respecto de la implementación del MPRUSPS

8.1 Introducción

De acuerdo a los resultados obtenidos en los casos experimentales propuestos (Capítulos: 5 y 6) se podría presumir que el Modelo de Priorización de Requerimientos de Usuario a partir de la Segmentación de las Preferencias de los Stakeholders (MPRUSPS), es capaz de proporcionar un EGPIRU representativo para el total de los *stakeholders*. No obstante a ello, resultaría conveniente debatir su faz metodológica, como así también, su conveniencia de adopción práctica.

Como consecuencia de lo descripto, el presente capítulo iniciará el debate de los aspectos metodológicos del MPRUSPS, para luego, considerar las restricciones prácticas que se deberán considerar, a la hora de adoptarlo como modelo para la priorización de requerimientos de usuarios. Finalmente, se concluye respecto de aquel escenario en donde la aplicabilidad del MPRUSPS resultaría más aconsejable.

8.2 Discusión sobre los aspectos metodológicos del MPRUSPS

El modelo en cuestión, formaliza el relevamiento de las preferencias de implementación de los requerimientos de usuario de los *stakeholders*, mediante la aplicación de la Técnica AHP. Ante la posibilidad de aplicar el modelo en aquellos casos en donde la cantidad de requerimientos a priorizar sea muy acotada (por ejemplo como el caso experimental “Subsistema de Flujo de Trabajo del CIMEPB”), se ha recopilado una serie de trabajos a los efectos de evaluar si la cantidad de requerimientos incide o no sobre la factibilidad técnica del uso del método AHP:

Autor	Año	Cantidad Requerimientos Priorizados
Jung	1999	14
Iqbal* (et. al.)	2010	4,9,10,11,8
Karlsson y Ryan	1997	14
Karlsson (et. al.)	1998	11
Motupally	2008	3,8
Perini (et. al.)	2009	20
Sadiq (et. al.)	2009	5
Sadiq (et. al.)	2010	8

Tabla 33: “Experimentos de Priorización de Requerimientos de Software basados en AHP”.

Fuente: Elaboración Propia.

*Utiliza Priorización con AHP por bloque de Requerimientos (bin).

Como se puede apreciar en la Tabla 33, se han citado algunos trabajos que han utilizado el método AHP, como fundamento básico, a la hora de implementar un determinado proceso de priorización de requerimientos de software. Más allá de las particularidades de cada uno de los enfoques propuestos, la cantidad de requerimientos a utilizar, parece no condicionar la aplicación del método en cuestión. **En consecuencia, se considera que una reducida cantidad de requerimientos de usuarios a evaluar en un Proyecto de Software, no condicionaría la aplicabilidad de la técnica de AHP a los efectos de relevar los vectores de preferencia de cada uno de los *stakeholders*.**

Otra cuestión a considerar, se halla en la practicidad de aplicar el método AHP, cuando la cantidad de requerimientos a evaluar sea considerable (como por ejemplo en el caso experimental simulado en el Capítulo 5). **Si bien la cantidad de requerimientos no impediría la aplicabilidad teórica de la técnica AHP, algunos autores consideran que la**

comparación por pares propuesta por el método AHP, se incrementaría notablemente ante la existencia de un conjunto extenso de requerimientos a valorar. Luego, el proceso se volvería muy complejo en términos prácticos para el *stakeholder* [Iqbal (et. al., 2010); Lethola (et. al., 2004); Motupally (2008), entre otros]. Al respecto, otros autores como por ejemplo Perini (et. al., 2009) indican que la aplicabilidad de la técnica AHP sería viable en términos prácticos, si la cantidad de requerimientos a evaluar no superan los 20. Otros en tanto, consideran que la existencia del indicador de inconsistencia de la técnica AHP (CR en Ecuación VIII), promueve un mecanismo de control sobre el juicio provisto por el *stakeholder*, independientemente de la cantidad de criterios que haya tenido que valorar. En síntesis, existen opiniones diversas respecto de la implementación práctica del AHP, en aquellos escenarios en donde la cantidad de requerimientos a evaluar sea muy extensa.

No obstante a ello, en Iqbal (et. al., 2010) por ejemplo, han propuesto una solución viable a este problema. Los autores, indican que la misma consistiría en clasificar a los requerimientos disponibles en bloques predeterminados, para luego, iniciar un proceso de priorización a nivel de bloque y luego entre bloques. De esta manera, la cantidad de comparaciones entre pares se reduciría significativamente, lo que permitiría aplicar eficazmente la técnica AHP en aquellos contextos en donde la cantidad de requerimientos a evaluar, sea muy extensa. **Luego, la cuestión de la practicidad de la técnica AHP no se vería afectada en estos contextos.**

Por su parte, y a diferencia de los enfoques citados en la Tabla 33, y de otros tantos como por ejemplo los desarrollados en la Sección 2.5 (Priorización de Requerimientos: “Enfoque Métodos”), el MPRUSPS no sintetiza las preferencias de los *stakeholders* en un sólo vector de distribución. Por el contrario, intenta agruparlas de acuerdo al patrón de preferencia exhibido por los *stakeholders*. Luego, pretende relevar un esquema particular de prioridad (EPPIRU) en cada grupo, y en función del desempeño socio-cognitivo obtenido a partir de la identificación de aquellos conceptos considerados como claves en el dominio (VTV), se pretende agregar cada esquema particular (EPPIRU) en uno global, denominado en esta tesis, como Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU). Este aporte en particular, trae aparejado una serie de beneficios.

En primera instancia, se podrá constatar rápidamente la existencia de aquellos grupos de *stakeholders* cuyos niveles de preferencia disten de forma significativa de aquel constituido a nivel global. De esta manera, será factible aplicar un proceso de negociación focalizado. Con lo cual, se evitará la necesidad de implementar un proceso de interacción *ad-hoc* a los efectos de encontrar grupos conflictivos, como sí resulta en los casos de:

Boehm y Ross (1989); Boehm (et. al., 1995); Boehm y In (1996); In (et. al. 2001); Fricker y Grünbacher (2008); Lehtola y Kauppinen (2004); Lu y Jing (2009).

Asimismo, se ha fundamentado que la configuración de los grupos responde a los aspectos socio-cognitivos de los *stakeholders* (Sección 3.3, Capítulo 3). A tal efecto, se ha incorporado e integrado una serie de métodos cognitivos con la finalidad básica de ponderar la preferencia de los *stakeholders* de acuerdo al grado de conocimiento que ellos detentan, respecto del uso e implicaciones de la futura herramienta de software (operado por los conceptos claves del dominio). Con lo cual, el MPRUSPS ponderará con más fuerza aquellas preferencias que provengan de grupos (de *stakeholders*) cuyo grado de conocimiento de los conceptos claves del dominio, sea relativamente más alto comparados con otros. De esta manera, se pretende satisfacer las inquietudes teóricas planteadas por autores como Martín (et. al, 2003); Martínez Carod y Cebrich (2005 y 2007), respecto de la necesidad de profundizar aquellas investigaciones que pretendan integrar los aspectos cognitivos de los *stakeholders*, con enfoques de priorización de requerimientos de software (Sección 2.3 del Capítulo 2). **Sin embargo, el MPRUSPS no sólo incorpora los aspectos cognitivos de los *stakeholders* como fundamentación teórica de las diferencias halladas en torno las preferencias relevadas, sino que, además, pretende contar con una serie de métricas socio-cognitivas que servirán como insumo básico, ante un eventual proceso de negociación focalizado.**

En este contexto, el Ingeniero en Requerimientos recopilará la información básica del grupo no alineado (Etapa Pre-Negociación), para luego conducir en caso de no llegar a un acuerdo, una Etapa de Negociación según lo recomendado por la Sección 3.5.2 (Capítulo 3). En esta instancia, el Ingeniero en Requerimientos actuará como Calificador de los argumentos generados por el Grupo No Alineado. **Esta estrategia, queda respaldada por las conclusiones establecidas en Damian (et. al. 2008), puesto que los autores consideran que un proceso de negociación de requerimientos, será más efectivo a medida que los *stakeholders* compartan un entendimiento común del dominio, y además, promuevan una comunicación cuyo contenido sea semánticamente válido y aceptado por los diferentes involucrados.** Luego, se prevé una instancia de aprendizaje tanto del Ingeniero en Requerimientos como del Grupo, respecto de las implicancias de proponer al EGPIRU como el plan de implementación de los requerimientos de usuarios en un determinado proyecto.

Por otro lado, vale la pena mencionar las implicancias de un caso en particular. Como se ha mencionado, el MPRUSPS propone segmentar (agrupar) las preferencias de los *stakeholders*, de acuerdo al grado de similitud exhibido entre ellas. Para ello, implementa la red neuronal de Kohonen y el Índice de Davies-Bouldin. Pero, *¿Qué*

sucedería en el caso en que tanto la red neuronal como el Índice Davies-Bouldin propongan la existencia de un sólo Grupo entre la totalidad de los stakeholders del proyecto?. **Pues en este caso, y de acuerdo a los resultados provistos en las Ecuaciones: XXVIII y IXXX (Secciones 4.7 y 4.8), el EPPIRU será igual EGPIRU. Luego, el EGPIRU será igual al propuesto por el enfoque de Karlsson (1996).** En este escenario en particular, se deberá constatar si los *stakeholders* aceptan o no el EGPIRU, y llegado el caso, implementar un proceso de negociación (Sección 3.5) entre quienes estén a favor y en contra de tal implementación.

En relación a esto último, se fundamenta el porqué no se ha determinado en el Capítulo 5, la aplicación de un escenario en donde los Grupo 1 y 2 obtengan el mismo nivel de VTV (ó aproximados). Dado que, la existencia de grupos bien diferenciados en torno al patrón de preferencia, hace suponer de la existencia de perfiles socio- cognitivo también diferentes (Supuesto establecido en la Sección 3.3). Luego, de existir más de un grupo, no es aplicable un mismo nivel de VTV en cada uno de ellos. ***En consecuencia, los EGPIRU propuestos en MPRUSPS y en Karlsson (1996) difieren cuando existen más de un Grupo, tal cual se ha demostrado en el Capítulo 5.***

8.3 Restricciones a considerar a la hora de aplicar el MPRUSPS

Para poder aplicar el Modelo de Priorización de Requerimientos de Usuarios a partir de la segmentación de las preferencias de los *stakeholders* (MPRUSPS), resulta esencial satisfacer los supuestos en que éste se basa. Por un lado, se debe partir de una Especificación de Requerimientos validada por aquellos *stakeholders* que han intervenido mayoritariamente en su formulación (Supuesto 1). Asimismo, el Ingeniero en Requerimientos deberá ser capaz de formular una serie de Requerimientos de Usuarios que puedan ser valorados en forma autónoma por parte del *stakeholder* (Supuesto 2). Esto último significa que el *stakeholder* puede, en base a su percepción, discernir la importancia relativa que adquiere cada uno de los Requerimientos de Usuarios. Por último, se considera que cada *stakeholder* tiene el mismo peso relativo en la formulación del plan de priorización de los requerimientos de usuarios (Supuesto 3).

Por su parte, y en base a lo que se ha supuesto respecto del proceso de formulación de las preferencias, se ha acordado que los *stakeholders* provienen de una misma comunidad lingüística, y que además, el relevamiento de las preferencias acontece

en un momento dado (Capítulo 3. Sección 3.3). Esto último, trae aparejado una serie de implicancias. En primer lugar, se deberá comprender que la misma comunidad lingüística hace referencia a un conjunto de *stakeholders*, que mínimamente, compartan el mismo idioma (nativo) y que además residan en una misma ciudad o al menos región. Por ejemplo, es muy probable que un colectivo social que resida en una ciudad marítima, pueda tener una producción semántica del concepto “Mar” un tanto diferente si la comparamos con aquella que surja de un colectivo social que resida en una ciudad mediterránea. En este sentido, tal producción semántica se verá incidida por el factor – espacio. Por su parte, es muy probable también que la producción semántica de “software” eliciteda para un colectivo social de Ingenieros en Sistemas en el año 1982, pueda diferir de aquella que se obtenga en el año 2011, inclusive si fuese establecida por el mismo grupo social. Luego, la producción semántica del concepto se vería incidida por el factor temporal.

Como consecuencia de lo descripto, se debe tener en consideración que el presente Modelo deberá ser aplicado, sí y sólo sí, se cuenta con un Proyecto de Software cuyos principales *stakeholders* se encuentren residiendo en espacios territoriales homogéneos (por ejemplo un mismo país / región dependiendo del grado de singularidad que tenga los conceptos claves del dominio del producto de software). Finalmente, y considerando el factor temporal, el modelo podrá ser aplicado sólo en aquellos escenarios en donde el proceso de priorización de requerimientos de software deba ser implementado en el corto plazo.

Otro importante factor a considerar, deviene de las habilidades personales que deba detentar el Ingeniero en Requerimientos. Tal cual lo sugerido por Fricker y Grünbacher (2008) y la Guía SWEBOOK (2004), la habilidad de negociación en el Ingeniero en Requerimientos debe estar especialmente incentivada en él, a los efectos de satisfacer el cumplimiento de los objetivos propuestos transversalmente en el proceso de negociación de requerimientos. Por su parte, y en forma específica en el *MPRUSPS*, el Ingeniero en Requerimiento deberá considerar la importancia que adquiere el aspecto cognitivo dentro del Proceso de Priorización de Requerimientos de Software. Al efecto, Martínez Carod (2007) junto a otros, coinciden en señalar que dicho aspecto aun no se encuentra lo suficientemente desarrollado y formalizado dentro de la Ingeniería en Requerimientos en general.

8.4 Consideraciones finales del Capitulo

El presente Capitulo, se ha propuesto discutir los principales fundamentos metodológicos del MPRUSPS. También, se han considerado una serie de restricciones que deberían tenerse en cuenta, a la hora de aplicar el presente modelo. Luego, se podría presumir que la aplicación del MPRUSPS resultaría práctica y efectiva, en aquellos Proyectos de Software de relativa envergadura (aproximadamente hasta 20 Requerimientos de Usuarios), en donde sus principales *stakeholders* residan en el mismo espacio territorial, conformando un grupo social heterogéneo en términos socio-cognitivos (conocimiento del dominio, intereses, rol desempeñado en el proyecto, entre otros). **En estos términos, el MPRUSPS cobraría vital importancia en aquellos escenarios en donde coexistan las siguientes particularidades: “Relativamente pocos Requerimientos de Usuarios” y “Relativamente gran cantidad de *stakeholders*”**

Capítulo 9

“Conclusión, Recomendaciones y Futuros Trabajos”

9.1 Conclusión

A lo largo de la presente Tesis, se ha analizado y justificado la importancia que adquiere la temática de estudio **“Priorización de Requerimientos de Software”**. En efecto, varios autores la han definido como aquella actividad crítica dentro de la Ingeniería de Requerimientos, que hace referencia a un proceso de decisión tendiente a establecer el orden de implementación de una serie disponible de requerimientos de software, tomando en consideración, algún conjunto definido de variables objetivo, como ser: satisfacción de los *stakeholders*, importancia, costo de implementación, necesidades del cronograma, entre otras [Gomes y Pettersson (2007); Hadad (et. al. 2009); Moisiadis, (2002); Ngo-The y Ruhe (2005) por citar algunos]. Luego, la presente investigación parte de la definición aportada por Firesmith (2004): ***“Proceso que determina el orden de implementación de los requerimientos de un sistema ó como aquel proceso que determina el orden de importancia de los requerimientos en función de la percepción de los stakeholders”***.

Por su parte, se ha podido delimitar el “Problema de la Priorización de Requerimientos Software” de otro problema íntimamente relacionado con él, el cual se lo denomina, **“Problema de la Próxima Versión (NRP)”**. Luego, se ha acordado que las cuestiones inherentes a la planificación del producto, factibilidad de implementación, entre otras, serán inherentes al problema NRP y no al problema de priorización de requerimientos de software.

Con el objeto de introducir la complejidad del fenómeno “Priorización de Requerimientos de Software”, se ha contextualizado el objeto de estudio dentro del Cuerpo de Conocimiento de la Ingeniería de Software (SWEBOK). A partir de allí, se ha constado que el abordaje del problema implicaría un análisis temático de las diversas Unidades de Conocimiento que integran el Área: Requerimientos de Software (*“Software Requirement”*) de la Guía SWEBOK (2004).

Como consecuencia de ello, y una vez advertida la complejidad multidimensional del problema objeto de estudio, se ha emprendido un proceso de revisión bibliográfica tendiente a establecer de qué manera la disciplina Ingeniería de Requerimientos, ha formulado y desarrollado algún tipo de enfoque que permitiese, por un lado, el abordaje del fenómeno, y por el otro, su resolución. Habiéndose procedido de esta forma, se ha **relevado la existencia de tres grandes enfoques de Priorización de Requerimientos de Software: “Enfoques de Negociación”, “Enfoque Cognitivo”, “Enfoque Método”**. Luego y a partir de allí, se ha descrito y desarrollado sus principales líneas de investigación.

De su análisis, se ha identificado que los enfoques de negociación conllevan un ciclo de interacción entre los *stakeholders* a los efectos de elicitar posiciones, detectar conflictos y promover un marco de negociación tendiente a especificar un esquema de priorización de requerimientos de software, que sea aceptable por el universo de *stakeholders*. No obstante a ello, dicho enfoque implica necesariamente un abordaje cognitivo de los *stakeholders*, ya sea para consensuar una estructura semántica de los conceptos claves del dominio y/o para conocer y justificar la multiplicidad de factores que llevan o no a un principio de acuerdo entre ellos. *Sin embargo, y tal cual se ha mencionado oportunamente, en Martín (et. al, 2003), Martínez Carod y Cebrich (2005) y Martínez Carod (2007) se indica que la mayoría de los enfoques utilizados en la priorización, ignoran en gran parte los aspectos cognitivos de los sujetos intervinientes.*

Desde otra vertiente, se han desarrollado una sucesión de métodos que permiten iniciar un proceso de priorización a partir de la valorización particular o conjunta de una serie de requerimientos implementables. De esta manera, y según Motupally (2008), tales métodos pueden clasificarse en dos grandes grupos, en función de la escala de medida que se utilice para efectuar las ponderaciones de prioridad: Escala Ordinal, Escala de Ratios. De esta manera, el “Enfoque Método” se ha convertido en una alternativa de abordaje para la priorización de requerimientos de software, como por ejemplo se puede apreciar en: Aho (et. al., 1983); Beck (2000); Brackett (1990); Iqbal (et. al., 2010); Karlsson (1996); Karlsson y Ryan (1997); Karlsson (et. al. 1998); Leffingwell y Widring (2000); Motupally (2008) entre otros. Luego, también es apreciable que dicho enfoque ha sido recombinado con diversos *framework* de negociación de requerimientos, a los efectos de formalizar algunos aspectos que se han constituido de manera *ad-hoc* en estos últimos. Al respecto pueden apreciarse los trabajos de: Fricker (2009); Gaur (et. al., 2010); In (et. al., 2002); Ramzan (et. al., 2009); Ruhe (et. al, 2002); Zhang (et. al., 2011), entre otros.

Sin embargo y a pesar del fructífero campo de investigación que ha ofrecido la temática: “Priorización de Requerimientos de Software” en estos últimos años, no se ha vislumbrado al momento de comenzar con el presente Proyecto de Investigación, **aquel**

enfoque capaz de fusionar en forma coherente y formal, los aspectos socio-cognitivos que subyacen de una manifestación de preferencia, el proceso de negociación focalizado para la resolución de conflictos, y finalmente, el establecimiento de un esquema de priorización de requerimientos consensuado por parte del universo de *stakeholders*. Razón por la cual, se ha establecido como Objetivo General de Tesis, *modelar un esquema de priorización de requerimientos de software, a partir de la segmentación de preferencias de los stakeholders*.

Luego y con el fin de satisfacer este propósito, se ha definido conceptual y operacionalmente, cada una de las variables que integran el Modelo de Priorización de Requerimientos de Usuarios a partir de la Segmentación de las Preferencias de los Stakeholders (MPRUSPS). **Dicho modelo, se postula como un enfoque experimental para el tratamiento integral del problema que supone priorizar y negociar los requerimientos de usuario de un determinado Proyecto de Software.**

Desde el punto de vista metodológico, el modelo utiliza el Método: “Proceso de la Jerarquía Analítica (AHP) de Saaty (1980, 1986, 1991) bajo la reformulación propuesta por Karlsson (1996) con el objeto de relevar las preferencias de los *stakeholders*. Luego y por aplicación del supuesto establecido en la Sección 3.3, recurre a la técnica computacional: Mapa Auto-Organizados de Kohonen (Sección 3.4.2) a los efectos de segmentar (agrupar) aquellas preferencias que sean consideradas similares respecto a su patrón de comportamiento. Luego, la calidad de dichos agrupamientos, quedará evaluada mediante la utilización del Índice de Davies-Bouldin (Sección 3.4.3).

A partir de allí, y través de un procedimiento de agregación lineal, se procede a estimar y ordenar los valores medios de preferencia en cada uno de los requerimientos de usuarios para cada grupo en particular. Resulta en consecuencia, el **Esquema Particular de Prioridad Implementable de Requerimientos de Usuarios de cada uno de los grupos (EPPIRU)**. Acto seguido, el Modelo procede a la agregación de cada vector de preferencia grupal, mediante la ponderación del desempeño socio-cognitivo⁴ inherente a cada uno de los grupos consignados (Media Ponderada). Luego, ordena los términos del vector resultante con el fin de hallar el **Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU)**.

Una vez establecido el EGPIRU, se procede a evaluar el grado de alineamiento que ha adquirido cada EPPIRU en relación a dicho esquema general. Este procedimiento, operado mediante la especificación de un **Sistema de Memoria Asociativa Difusa (FAM)**,

⁴ Operado mediante el cociente: $VTV \text{ del Grupo} / \text{Sumatoria } VTV$

permite asistir al Ingeniero en Requerimientos, en la decisión que supone determinar qué grupo deberá ser pasible de un **Proceso de Negociación**⁵.

*En síntesis, con el desarrollo del MPRUSPS, se podrá concluir que se ha satisfecho el objetivo general de investigación, construyendo en consecuencia, un **enfoque de priorización de requerimientos de software a partir de la segmentación de las preferencias de los stakeholders**. Para ello, se ha **fusionado** una serie de **aspectos teóricos de la Priorización de Requerimientos de Software** (Enfoque Negociación y Método) **junto a otros provenientes de la Inteligencia Computacional** (Red Neuronal de Kohonen, Elementos de la Teoría de Conjuntos Difusos) **y a la Psicología Cognitiva** (Definición Finder, Valor Total de las Valorizaciones, Razón de Exclusión, entre otros elementos). **Finalmente, el presente modelo, podría ser aplicado eficazmente en aquellos Proyectos de Software de pequeña envergadura (por ejemplo hasta 20 Requerimientos de Usuarios) caracterizados por un conjunto extenso y heterogéneo**⁶ **de stakeholders, cuya finalidad inmediata, sea determinar un esquema general de prioridad implementable de requerimientos de usuarios (EGIPIRU) consensuado y validado por dicho colectivo social.***

9.2 Recomendaciones y Futuros Trabajos

En términos generales, el Problema de Priorización de Requerimientos de Software, implica el abordaje de un fenómeno de naturaleza compleja. **Con lo cual, la primera recomendación que se efectúa al respecto, consta de especificar explícitamente el criterio de Prioridad en que se basará el enfoque propuesto.** Ya que, como se ha destacado, el criterio de prioridad que se utilice puede condicionar el alcance del proceso de priorización que se formule.

Asimismo, vale la pena destacar que la temática “Priorización de Requerimientos de Software”, ha adquirido y está adquiriendo, una elevada potencialidad en términos de investigación científica. Testimonio de ello, surge de la gran proliferación de publicaciones que se han gestado en estos últimos años, en torno a dicha problemática. **Luego, especial significación debería asignarse a la inserción de componentes cognitivos dentro del proceso de priorización de requerimientos de software, ya que, como se ha indicado, el desarrollo de esta línea de investigación, aún resulta incipiente.**

⁵ Proceso de Negociación Aplicado. Sección 3.5 Capítulo 4.

⁶ Heterogéneo debe entenderse por un Colectivo Social diverso en términos de intereses, conocimientos, roles, entre otros aspectos

Considerando las Futuras Investigaciones que pueden desprenderse a partir de la presente, resultaría factible proponer una serie de extensiones al MPRUSPS. En primer lugar, y ante el escenario que supone la evaluación de una extensa cantidad de requerimientos de usuarios (por ejemplo más de 20), resultaría interesante analizar algún tipo de clasificación previa de los requerimientos de usuarios. De esta manera, se podría iniciar un proceso de priorización al interior de cada “Grupo de Requerimientos de Usuarios” y luego entre Grupos. El principal beneficio que traería aparejado esta propuesta, consistiría en reducir significativamente la cantidad de comparaciones a pares que debería efectuar el *stakeholder*. Al respecto, se recomienda analizar la propuesta desarrollada en Iqbal (et. al., 2010). De resolver el eventual problema que supondría aplicar la Técnica AHP para la evaluación de un conjunto extenso de requerimientos, el MPRUSPS podría aplicarse de manera efectiva y conveniente, en aquellos Proyectos de Software considerados normalmente como medianos y grandes.

Otra extensión que podría plantearse al MPRUSPS, consistiría luego de invalidar su tercer supuesto: “Cada *stakeholders* tiene el mismo peso relativo en la formulación del plan de priorización de requerimientos (**Supuesto 3**)”. Partiendo de allí, podría plantearse una investigación que pueda responder las siguientes inquietudes:

- ***¿Qué criterios se podrían utilizar a los efectos de imputar una ponderación diferencial en el esquema de prioridad de un stakeholder?***
- ***¿Cómo se podría modelar la incidencia diferencial de las funciones de preferencia dentro de un Esquema General de Prioridad Implementable de Requerimientos de Usuarios (EGPIRU)?***
- ***¿Qué efectos se propagarían dentro del Proceso de Negociación aplicado, si cada stakeholder adquiere un peso diferencial en su preferencia?***

En relación a los Supuestos 1 y 2 del MPRUSPS, los cuales establecen: “Se debe partir de una Especificación de Requerimientos validada por la totalidad de los *stakeholders* que participaron en su construcción” y “Cada Requerimiento de Usuario que se desprenda de dicha Especificación, puede ser priorizado en forma autónoma por parte del *stakeholder* y es independiente del resto” respectivamente, **se considerarán como críticos para el correcto funcionamiento del modelo**. Al efecto, se debe destacar que un Proceso de Priorización de Requerimientos se constituye como una etapa ulterior y sumamente dependiente a la de Especificación y Validación de Requerimientos. Con lo cual, sí se inicia un proceso de priorización cuando dicha especificación no satisfaga sus

atributos de calidad⁷, entonces, los resultados provistos por dicho proceso serán inválidos. Con referencia al segundo supuesto, se considera que priorizar a nivel de requerimientos de usuarios podría mitigar el impacto de la dependencia entre los mismos, comparado por ejemplo, si se hubiese priorizado a nivel de los requerimientos funcionales del sistema. Puesto que, los requerimientos de usuarios representan entidades conceptuales más abstractas, su ordenamiento en términos de prioridad, podría flexibilizar el proceso de planificación de las versiones del producto, por parte del Equipo de Desarrolladores.

Por último, y considerando que la aplicación del **MPRUSPS** excede las incumbencias generales de un Ingeniero de Requerimientos, podría planificarse el desarrollo de una herramienta que lo automatice total o parcialmente.

⁷ Completa, Consistente, Correcta, Modificable, Concisa, Testeable, No-Ambigua, Comprensible, Validable, Verificable, Independiente, entre otras.



Bibliografía Utilizada en la Tesis

- Abran, A; Moore, J (2004) The guide to the software engineering body of knowledge. SWEBOK. En Línea: <http://www.computer.org/portal/web/swebok/htmlformat>
- Aczel, J; Saaty, T (1983) "Procedures for synthesizing ratio judgments". Journal of Mathematical Psychology. 27. Pg: 93-102.
- Aczel, J; Saaty, T (1987) "Synthesizing judgments: a functional equation approach". Journal of Mathematical Psychology. 27. Pg: 311-320
- Aho, A; Hopcroft, J; Ullman, J (1983) Data structures and algorithms. Reading, MA. Addison-Wesley.
- Aly, S; Vrana, I; Vanicek, J (2010) "Logical and Decisive combining criterion for binary group decision making". Journal of Systems Integration. 1. (2). Pg: 43-53.
- Alhoiemi, E; Himberg, J; Parhankangas, J; Vesanto, J (2005) "SOM Toolbox". Laboratory of Computer and Information Science. SOM Toolbox Team.Finland.
- Alvarez Garcia, J (2010) "Clasificación Automática de información en Portales Web mediante técnicas de clustering". Departamento de Informática. Universidad de Salamanca. Tesis Doctoral.
- Anderson, J; Evans, M (1995) "A Generic Simulation System for Intelligent Agent Designs", Applied Artificial Intelligence, 9, 5, October, 1995, pp. 527-562.
- Andriola, W (2008) "Uso da teoria da resposta ao ítem (Tri) para analisar a equidade do proceso de avaliação do aprendizado discente. Revista Iberoamericana de evaluación educativa. 1. Pg: 172-89.
- Attwood, K; Kelly, T; McDermid, J (2004) "The use of Satisfaction Arguments for traceability in Requirements reuse for System families: Position paper". Proceedings of the International Workshop on Requirements Reuse in System Family Engineering, Eighth International Conference on Software Reuse. Pg: 18-21.
- Arbolera, O; Chérrez, R (2006) "Técnicas de Control Difuso aplicadas a la navegación de un robot móvil". Tesis Grado: Ingeniería en Electrónica. Escuela Politécnica Nacional. Quito.
- Aurum, A; Wholin, C (2003) "The fundamentals nature of requirements engineering activities as a decision making-making process". Information and Software Technology. 45. Pg: 945-954.

- Azar, J; Smith, R; Cordes, D (2007) "Value-Oriented Requirements Prioritization in a Small Development Organization". IEEE Software. January/February 2007. Pg: 32-37.
- Becker, S (1991)"Unsupervised learning procedures for neural networks". The International Journal of Neuronal System. Vol 1 &2. Pg: 17-33.
- Back, T (1996) "Evolutionary Algorithms in Theory and Practice". Oxford Univerty Press. New York.
- Bagnall, A; Rayward-Smith, V; Whittle, I (2001). "The next release problem". Information and Software Technology. 43. (14). Pg: 883-890.
- Bai, C; Sarkis, J (2010) "Green supplier development: analytical evaluation using rough set theory". Journal of Cleaner Production. 18. Pg: 1200-1210.
- Barragán Piña, A (2009) "Síntesis de Sistemas de Control Borroso Estables por Diseño". Tesis Doctoral. Universidad de Huelva.
- Barzilai, J (1997) "Deriving weights from pairwise comparison matrices". J Opl res Soc. 48. (12) Pg: 1226-1232.
- Beck, K (2000) Extreme programming explained. Reading, MA: Addison-Wesley.
- Berander, P (2004) "Prioritization of Stakeholder Need in Software Engineering". Department of Systems and Software Engineering. Blekinge Institute of Technology. Thesis degree Licentiate of Technology in Software Engineering.
- Bergman, B; Klefsjo, B (2003) Quality: From customer needs to customer satisfaction. Lund, Sweden, Studentlitteratur.
- Bidgoli, H. (1996). "Group decision support system". Manage.14.Pg: 56-62.
- Blanc, X; Mounier, I; Mougnot, A; Mens, T; (2008) "Detecting Model Inconsistency through Operation-Based Model Construction". ICSE'08, May 10-18, Leipzig, Germany.
- Boehm, B (1988) "A Spiral Model of Software Development and Enhancement". Computer, May, pp. 61-72.
- Boehm, B ; Ross, R (1989) "Theory W Software Project Management: Principles and Examples," IEEE Trans. Software Engr., July 1989.
- Boehm, B., Bose, P., Horowitz, E., and Lee, M.(1995) "Software Requirements Negotiation and Renegotiation Aids: A Theory-W Based Spiral Approach", in 17th International Conference on Software Engineering (ICSE- 17). Seattle: IEEE Computer Society Press.
- Boehm, B. and In, H. (1996) "Identifying Quality- Requirement Conflicts", IEEE Software, Vol. 13, No. 2, March.
- Boehm, B (2003) "Value – Based Software Engineering". Software Engineering Notes. 28.(2). Pg: 1-13. March.

- Bolshakova, N ; Azuaje, F (2002) "Cluster validation techniques for genome expression data". Department of Computer Science, Trinity College, Dublin 2, Ireland.
- Bonissone, P.P. (1999); Yu-To Chen; Goebel, K.; Khedkar, P.S.; Hybrid soft computing systems: industrial and commercial applications. Proceedings of the IEEE . Volume 87, Issue 9, Sept. 1999 Page(s):1641 – 1667.
- Brooks, F (1987) "No Silver Bullet". Computer Magazine. April.
- Brugha, C (2004). "Phased Multicriteria preference finding". European Journal of Operational Research. 158. (2). Pg: 308-316.
- Burg, J (1997) "Linguistic Instruments in Requirements Engineering". Amsterdam : IOS Press.
- Burgos, F (2003) "Tipos de Redes Neuronales". Capítulo 3 de Herramientas en GNU/Linux para estudiantes universitarios: Redes Neuronales con GNU/Linux. En línea:
http://softwarelibre.unsa.edu.ar/docs/descarga/2003/curso/htmls/redes_neuronales/x152.html [Consulta 3/3/2011].
- Carnevale, F (2010) "Orientación de un objeto 3D: Implementación de Redes Neuronales artificiales utilizando lógica programable": Tesis de Maestría en Ingeniería: Instituto Balseiro. CNEA.
- Carretero, M (1997) Introducción a la Psicología Cognitiva. Buenos Aires. Aique.
- Casellas i Serra, L (2009) "La Gestión de Documentos Electrónicos: Normas de Referencia y Contexto Tecnológico". ASARCA. Forma: 5.Pg: 1-36.
- Chang, A; Han, T (1995) "Design of an argumentation-based negotiation support system". System Science. 4. Pg: 3-6.
- Charles, D; Fyfe, C. (2000) "Kernel factor analysis with Varimax rotation". En: Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on.
- Coello-Coello, C (2004) Introducción a la Computación Evolutiva. Departamento de Ingeniería Eléctrica. CINVESTAV-IPN . En línea:
<http://delta.cs.cinvestav.mx/~ccoello/compevol/apuntes.pdf.gz> [Consulta: 3/3/2011].
- Damian, D; Lanubile, F; Mallardo, T (2008). "On the need for mixed media in distributed requirement negotiations". IEEE Transactions on Software of Engineering. Vol: 34, 1, January.
- Davis, A. (1993). Software Requirements: Objects, Functions and States, Prentice Hall.
- Davis, A. (2003). "The Art of Requirements Triage". IEEE Computer Society. March.

- Davies, D; Bouldin, D (1979) "A cluster separation measure," IEEE Trans. Patt. Anal. Machine Intell., vol. PAMI-1, pp. 224–227, Apr.
- Devlin, J. T., Gonnerman, L. M., Andersen, E. S.; Seidenberg, M. S. (1998). "Category-specific semantic deficits in focal and widespread brain damage: A computational account". Journal of Cognitive Neuroscience. 10. Pg: 77-94.
- Desanctis, R; Gallupe, R (1987). "A foundation for the study of group decision support systems. Manage Science. 33. Pg: 589-609.
- Document Lifecycle Management Forum (2008) "MoReq2 Specification". En Linea: http://www.dlmforum.eu/index.php?option=com_content&view=category&layout=blog&id=901&Itemid=20&lang=en [Consulta: 9/09/2010]
- Dick, J (2000) "Rich Traceability". In Automated Software Engineering Conference Edinburgh, Scotland.
- Díez Ruano (2003) "Técnica de Agrupamiento para identificación y control por modelos locales". Departamento de Ingeniería de Sistemas y Automática. Universidad Politécnica de Valencia. Tesis Doctoral.
- Doerr, J; Hartkopf, S; Kerkow, D; Landamann, D; Amthor, P (2007) "Built-in user satisfaction – Feature appraisal and prioritization with Amuse". 15th IEEE International Requirements Engineering Conference. Pg: 101-110.
- Duda, R. O ; P. E. Hart (1973). Pattern Classification and Scene Analysis. John Wiley & Sons. New York.
- Duda, R. O., P. E. Hart and D. G. Stork (2000). Pattern Classification. John Wiley & Sons. New York.
- Durrillo, J; Zhang, Y (2009) "A study of Multi-Objective Next Release Problem". In Proceedings of the 2009 1st International Symposium on Search Based Software Engineering. IEEE Computer Society. Pg: 48-58.
- Easterbrook ,J S; Chechik, M (2001) "A Framework for Multi-Valued Reasoning over Inconsistent Viewpoint" In Proc. of 21st International Conference on Software Engineering. Pp: 411–421.
- Engelbrecht, A (2002). Computational Intelligence: An Introduction. John Wiley. Firth Edition.
- Espín Andrade, R et. al.(2004) "Compensatory Logic: A fuzzy approach to decision making ". Proceeding of 4th International Symposium on Engineering of Intelligence System, 2004.
- Firesmith, D (2004) "Prioritizing Requirements". Journal of Object Technology. 3. (8). Pg: 35-47.
- Fogel, L; Owens, A; Walsh, M (1965) "Artificial Intelligence through a Simulation of Evolution". In M. Maxfield, A. Callahan, and L. J. Fogel, editors, Biophysics and Cybernetic Systems: Proceedings of the Second Cybernetic Sciences Symposium, pages 131–155. Spartan Books, Washington, D.C., 1965.
- Fogel, L (1966). Artificial Intelligence through Simulated Evolution. JohnWiley, New York, 1966.

- Fricker, S; Grünbacher, P (2008) "Negotiation Constellations – Method Selection Framework for Requirements Negotiation". REFSQ 2008, Berlin Heidelberg. Pp: 37-51.
- Fricker, S (2009) "Specification and Analysis of Requirements Negotiation Strategy in Software Ecosystems". In Proceedings of the first International Workshop on Ecosystem Software. CEURS –WS. Pg: 19-33.
- Fung, G (2001) A Comprehensive Overview of Basic Clustering Algorithms. En línea: <http://pages.cs.wisc.edu/~gfung/clustering.pdf>
- Garrard, P., Lambon Ralph, M. A., Hodges, J. R., & Patterson, K. (2001). "Prototypicality, distinctiveness, and intercorrelation: Analyses of the semantic attributes of living and nonliving concepts". Cognitive Neuropsychology. 18. Pg: 125-174.
- Gaur, V; Sonny, A; Bedi, P (2010) "An application of Multi-Person Decision-Making Model for Negotiation and Prioritizing Requirement in Agent-Oriented Paradigm". International Conference on Data Storage and Data Engineering. Pg: 164-168.
- Giesen, J ; Völker, A (2002) "Requirements interdependencies and stakeholders preference". Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering. Pg: 206-212.
- Gomes, A; Pettersson, A (2007) "Market Driven Requirements Engineering Process Model –MDREP". Karlskrona: Blekinge Institute of Technology.
- González, M; Comesaña, A (2009) "La evaluación semántica objetiva como contribución a la evaluación de los aprendizajes en la enseñanza universitaria". En: Evaluación de Redes Semánticas, Instrumentos y Aplicaciones. Ediciones Suarez. 1era. Ed. 2009. Capítulo VI.
- Hadad, G; Doorn, J; Ridao, M; Kaplan, G (2009) "Facilitando la asignación de prioridades a los requisitos". WER'09. Aceptado para publicación. Valparaiso, Chile. Julio.
- Harman, M (2007) "The current state and future of Search Based Software Engineering". In FOSE 2007.
- Harman, M; Jones, F (2001) "Search Based Software Engineering". Information and Software Technology. 43. (14). Pg: 833-839.
- Hatton, S (2007) "Early prioritization of goal. In Advances in conceptual modeling. Foundations and applications. Pg: 235-244.
- Hatton, S (2008) "Choosing the right prioritization method". 19 th Australian Conference on Software Engineering. Pp: 517-526.
- Haykin, S (1994) Neuronal Networks: A comprehensive foundations. Macmillan College Publishing Company.

- Hebb, D (1949) "The organization behavior : A neurophysiological Theory". Wiley.
- Henry, J; Henry, S (1993) "Quantitative assessment of software maintenance process and requirements volatility. In Proceeding of ACM Conference on Computer Science. Pg: 346-351.
- Herrera-Umaña, M; Osorio-Gómez, J (2006) "Modelo para la gestión de proveedores utilizando AHP difuso". Estudios Gerenciales. 99. (4). Pg: 69-88.
- Herrera-Viedma, E., Herrera, F., Chiclana, F; Luque, M. (2004) "Some issues on consistency of fuzzy preference relations". European Journal of Operational Research, Vol.154, No.1.Pp. 98-109.
- Hertz, J (1991) " Introduction to the Theory of Neural Computation". Perseus Publishing.
- Holland, D (1962) "Outline for a logical theory of adaptative system". Journal of Association for Computing Machinery. (9). Pg: 297-314.
- Huapaya, R; Lizarralde, F; Arona, G (2009) "INFOSEM: Un sistema informático para evaluar conocimiento basado en la distancia semántica". En: Evaluación de Redes Semánticas, Instrumentos y Aplicaciones. Ediciones Suarez. 1era. Ed. 2009. Capitulo IV.
- Hull, E; Jackson, K; Dick, J (2005) Requirements Engineering. Ed. 2^{da}.Springer-Verlag. London.
- IEEE, Std 610-1990 (1990). Standard glossary of software engineering terminology, IEEE.
- IEEE, Std 830-1998 (1998) IEEE Recommended Practice for Software Requirements Specifications, IEEE.
- In, H., Boehm, B., Rodgers, T., and Deutsch, M (2001) "Applying WinWin to Quality Requirements: A Case Study", IEEE International Conference on Software Engineering (ICSE 2001), IEEE Computer Society Press, Toronto, Canada, May 12-19, 2001, pp. 555- 564.
- In, H; Olson, D; Rodgers T (2002) "Multi-Criteria Preference Analysis for Systematic Requirements Negotiation. Proceedings of the 26 th Annual International Computer Software and Applications Conference (COMPSAC'02).
- Iqbal, M; Zaidi, A; Murtaza, S (2010) "A new Requirements Prioritization Model for Market Driven Products using Analytic Hierarchy Process". In: International Conference on Data Storage and Data Engineering 2010. Pg: 142-149.
- Isaacs, J; Foo, S; Meyer-Baese, A (2007) "Novel Kernels and Kernel PCA for Pattern Recognition". Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on
- Jackson, M (2001) Problem Frames: Analyzing and Structuring Software Development Problems. Addison-Wesley. London.
- Jang, J.S.; Gullet, N. (1997) "Fuzzy logic toolbox user's guide". MATLAB (v .1).

- Jewer, J; McKay, K (2008) "A software release planning methodology for developers". Proceedings of the Fourteenth Americas Conference on Information Systems, Toronto, ON, Canada, August 14th- 17th .
- Jung, H (1999) "Optimizing value and cost in Requirements Analysis". IEEE Software. July/August.
- Kacprzyk, J; Fredizzi, M; Nurmi, H (1992) "Fuzzy Logic with Linguistic Quantifiers in Group Decision Making". In "An Introduction to Fuzzy Logic Applications in Intelligent Systems", ed. R.R. Yager and L.A. Zadeh, pp. 263-280, Kluwer Academic Publishers, Norwell, MA, 1992.
- Karlsson, J. (1996). "Software Requirements Prioritizing". Proceedings of ICRE'96. IEEE Software.
- Karlsson, J; Ryan, K (1997) "A cost-value approach for prioritizing requirements". IEEE Software. September/October. Pg: 67-74.
- Karlsson, J; Wholin, C; Regnell, B (1998) "An evaluation of methods for prioritizing software requirements". Information and Software Technology. (39). Pp: 939-947.
- Karlsson, L; Berander, P; Regnell, B ;Wholin, C (2004) "Requirements Prioritisation: An experiment exhaustive pair-wise comparisons versus Planning Game Partitioning". Proceedings 8th Conference on Empirical Assessment in Software Engineering. Edinburg. UK.
- Karlsson, L., Höst, M., & Regnell, B. (2006). "Evaluating the Practical Use of Different Measurement Scales in Requirements Prioritisation". ACM. Pp: 326-335.
- In, H; Roy, S. (2001). "Visualization Issues for Software Requirements Negotiations". IEEE. Transactions on Software Engineering, July. pp: 10-15. pp: 10-15.
- Karlsson, L; Thelin, T; Regnell, B; Berander, P; Wholin, C (2007) "Pair-wise comparisons versus Planning Game Partitioning- experiments on requirements prioritisation techniques". Empir Software Eng. 12. Pg: 3-33.
- Kaiya, H; Shinbara, D; Kawano, J; Saeki, M(2005) "Improving the detection of requirements discordances among stakeholders" Requirements Engineering, vol.10, no.4, December. Pp.289–303.
- Keeney y Raiffa (1976) "Decisions with multiple objectives preferences and values trade-offs". J Wiley and Sons. New York.
- Kim, S; Ahn, B; (1999). "Interactive Group decision making procedure under incomplete information. European Journal of Operational Research. 116. (1). Pg: 498-507
- Kohonen, T. (1982) Self-organized formation of topologically correct feature maps. Biological Cybernetics, 43. Pg: 59-69.
- Kohonen, T(1990). "Self-Organizing Maps" Proceedings of the IEEE. 78.

- Kohonen, T(2001). Self-Organizing Maps. 30. Springer Verlag.
- Kumar, S; Vaidyaa, O. (2006) "Analytic hierarchy process: An overview and applications". European Journal of Operational Research. 169. (1). Pg: 1-29.
- Lamsweerde, A (2001) "Goal – Oriented Requirements Engineering: A guided Tour". Paper presented at the Proceedings 5th IEEE International Symposium on Requirements Engineering, Toronto, Canada.
- Leffingwell, D; Widring, D (2000) Managing Software Requirements – A unified approach. Addison-Wesley. Upper Saddle River, Nj.
- Lehtola, L; Kauppinen, M. (2004). "Empirical Evaluation of Two Requirements Prioritization Methods in Product Development Projects". Springer-Verlag. Berlin Heidelberg, 3281. Pp: 161-170.
- Lehtola, L; Kauppinen, M; Kujala, S (2004). "Requirements Prioritization Challenges in Practice". Proc. of 5th Int'l Conf. On Product Focused Software Process Improvement (PROFES), Kansai Science City.
- Lehtola, L; Kauppinen, M. (2006) "Suitability of Requirements Prioritizations Methods for Market Driven Software Product Development". Software Process: Improvement and Practice.11. (1). Pg: 7-19.
- Leskinen, P y Kangas, J (2005) "Rank reversal in multi-criteria decisión analysis with statistical modeling of ratio- scale pairwise comparisons". J Opl res Soc. 57. (6) Pg: 682-691.
- Li, R (1999). "Fuzzy method in goup decision making. Mathematic Applied. 38. Pg: 91-101
- Li, Y; Tang, J; Luo, X; Xu, J (2009). "An integrated method of rough set, Kano's model and AHP for rating customer requirements' final importance". Expert Systems with Applications 36. Pg: 7045-7053
- Lootsma, F (1993) " Scale sensitivity in the multiplicative AHP and SMART". J Multi-criteria Decis Anal. 2. (2). Pg: 87-110.
- López, J ; Dolado, J (2009) "Combinación de distribuciones de probabilidad con AHP". Actas de los Talleres de las Jornadas de la Ingeniería del Software y Bases de Datos. 3. (1). Pg: 37-49.
- Loucopoulos, P; Karakostas, V (1995). System Requirement Engineering. 1^{era}. Ed. McGraw. New York. USA.
- Lu, S (1999) "Modeling collaborative design process with a socio-technical framework". Proceedings of Sixth ISPE Int'l Conf. Concurrent Engineering, Bath, UK.

- Lu, S; Jiang, N; (2009). "A socio-technical negotiation approach for collaborative design in software engineering". *Int. J. Collaborative Engineering*. 1. (1/2). Pg: 185-209.
- Ma, Q (2009). "The Effectiveness of Requirements Prioritization Techniques for a Medium to Large Number of Requirements: A Systematic Literature Review". Thesis of Master: "Computer and Information Science". Auckland University of Technology. School of Computing and Mathematical Sciences. November.
- Mallo, P; Artola, M; Zanfrillo, A et. al. (2010) "Una propuesta de selección de entidades aseguradoras a partir de un Modelo de Lógica Difusa Compensatoria". *Cuadernos del CIMBAGE*. (12). Pg: 85-111.
- Marjaie, S; Kulkarni, V (2010) "Recognition of Hidden Factors in Requirements Prioritization Using Factor Analysis". En: *Computational Intelligence and Software Engineering (CiSE), 2010 International Conference on*.
- Martín, A; Martínez, N; Martínez-Carod, Aranda, G; Cechich, A (2003). "Classifying Groupware Tools to Improve Communication in Geographically Distributed Elicitation". *IX Congreso Argentino en Ciencias de la Computación, CACIC 2003, La Plata, Octubre*. Pp: 942-953.
- Martinez Carod, N. (2005). "Priorización de Requerimientos de Software utilizando una estrategia cognitiva". Departamento de Ciencias de la Computación. UNComa. Argentina.
- Martinez Carod, N ; Cechich, A (2005). "Classifying Software Requirement Prioritization Approaches". *XI Congreso Argentino en Ciencias de la Computación, CACIC 2005, Entre Ríos, Octubre*.
- Martinez Carod, N (2007). "Gestión de Preferencias de Requerimientos basadas en Técnicas Cognitivas". Departamento de Ciencias de la Computación. UNComa. Argentina.
- Martinez Carod, N and Cechich, A. (2007) "A Cognitive Psychology Approach for Balancing Elicitation Goals", Accepted In the Sixth IEEE International Conference on Cognitive Informatics (ICCI'07), California, USA.
- Maravall, D. (1993). *Algoritmos de agrupación de clases: clustering*. RA-MA. Barcelona.
- McNamara, T (2005) "Semantic priming: Perspectives from memory and word recognition". Nueva York: Psychology Press.
- Mendel, J (1995) "Fuzzy logic systems for engineering: A tutorial". *Proceedings of the IEEE*, 83(3). Pg: 345-377.
- Mizumoto, M (1988) "Fuzzy controls under various fuzzy reasoning methods. *Information Sciences*, 45(2). Pg:129-151.

- Moisiadis, F (2002) "The fundamentals of prioritizing requirements". Systems Engineering, Test Evaluation Conference, Sydney. Australian. October.
- Motupally, P (2008) "Using Satisfaction Arguments and Rich Traceability in Requirement Priorization". Thesis of Master of Computer and Information Sciences. Auckland University of Technology.
- Murphy, G. L. (2002). "The big book of concepts". Massachussets: MIT Press.
- Ngo-The, A; Ruhe, G (2005) "Decision support in requirement engineering". In Aurum & Wohlin (Eds). Engineering and managing software requirements. Pg: 267-286. Springer Berlin Heidelberg.
- Nuseibeh, B; Easterbrook, S (2000) "Requirements Engineering: A Roadmap". Proc. 22nd Int'l Conf. Software Eng., pp. 35-46, 2000.
- Nuseibeh, B; Kramer, J; Finkelstein, A (2003). "Viewpoints: Meaningful Relationships are Difficult!" In Proc. of 25th International Conference on Software-Engineering. Pp 676–681.
- Pascual Montano, A (2002) "Redes neuronales autoorganizativas basadas en optimización funcional. Aplicación en bioinformática y biología computacional". Departamento de Ingeniería Informática. Universidad Autónoma de Madrid. Tesis Doctoral.
- Pawlak, Z. (1982). "Rough sets". International Journal of Computer and Information Science. 11. Pg: 341-356.
- Pawlak, Z. (1997). "Rough set approach to knowledge – based decision support". European Journal of Operational Research. 99. (1). Pg: 48-57.
- Pazgón, E; Favarotto, V; Calcopietro, M; Azzolini, M; Vivas, J (2010) "Escala normalizada de Atributos". Congreso Nacional de Lingüística. Argentina, Abril de 2010.
- Pedrycz, W; Poland, W (2002) "Computational Intelligence as a Emerging Paradigm of Software Engineering". SEKE'02. Pg: 7-14.
- Peraita Adrados y Grasso (2010) "Corpus Lingüístico de definiciones de categorías semánticas de personas mayores sanas y con la enfermedad de Alzheimer". Documento de Trabajo.3. Fundación BBVA.
- Perini, A; Ricca, F; Susi, A (2009) "Tool-supported requirements prioritization: Comparing the AHP and CBRank methods". Information and Software Technology. 51. (6). Pg: 1021-1032.
- Poole, D; Mackworth, A; Goebel, R (1998) "Computational Intelligence: A logical approach". Computación y Sistemas. 2. (2) Pg: 146-9.
- Posner, M (1993). "Foundations of Cognitive Science". MIT Press.

- Quirós Hernández, M (2003) “Aplicación de Lógica Borrosa en Climatología”. Departamento de Geografía Facultad de Filosofía y Letras. Universidad de Valladolid. En línea: <http://www.cexeci.org/IX%20CONFIBSIG/Comunicaciones/Metodolog%EDa/Quir%F3s%20hern%E1ndez.pdf>. [Consulta: 28/02/2010].
- Ramzan, M; Jaffar, A; Iqbal, A; Anwar, S (2009) “Value Based Fuzzy Requirement Prioritization and Its Evaluation Framework”. icicic, pp.1464-1468, 2009 Fourth International Conference on Innovative Computing, Information and Control.
- Regnell; B; Host, M; Natt och Dag, J; Beremark, P; Hjelm, T (2001) “An Industrial case study on distributed prioritization in market –driven Requirements Engineering for Packaged Software. Requirements Engineering. 6. (1). Pg: 51-62.
- Rodríguez Bello, S. (2007). “Toma de decisión Multicriterio con AHP, ANP y LD”. Maestría en Ingeniería de Sistemas y Computación. Universidad Nacional de Colombia.
- Rogers, T. T. y McClelland, J. J. (2004) Semantic cognition: A parallel distributed approach. Massachussets: MIT Press.
- Rousseeuw, P.J.; Kaufman, L. (1990). Finding Groups in Data: An Introduction to Cluster Analysis. Wiley.
- Ruhe, G; Eberlein, A; Pfahl, D (2002) “Quantitative WinWin – A New Method for Decision Support in Requirements Negotiation” . Proceedings of the 14th international conference on Software engineering and knowledge engineering, July 15-19, 2002, Ischia, Italy. ACM.
- Ruhe, G (2003) “Software engineering decision support- A new paradigm for learning software organizations”. Advances in learning software organization. 260. Pg: 104-115.
- Rumelhart y Zipser (1985) “Feature discovery by competitive learning”. Cognitive Science. (1).
- Saaty, T (1980). The Analytic Hierarchy Process. NY. McGraw- Hill.
- Saaty, T,L. (1986). “Axiomatic foundations of the analytic hierarchy process”. Management Science. 32. Pp: 841-855.
- Saaty, T. (1990). “How to make decision : The Analytic Hierarchy Process” . European Journal of Operational Research. (48). North Holland. Pg: 9-26.
- Saaty, T,L. (2001) “Decision-Making whit the AHP: Why is the principal eigenvector necessary ?”. ISAHP 2001, Berne, Switzerland, August 2-4.
- Sadiq, M; Ghafir, S; Shahid, M (2009) “An approach for eliciting software requirements and its priorization using AHP”. International Conference on Advances in Recent Technology in Communication and Computing. IEEE Computer Society. 2009.

- Sadiq, M; Ahmed, J; Asim, M; Qureshi, A; Suman, R (2010) "More on Elicitation of Software Requirements and Prioritizations using AHP". In: International Conference on Data Storage and Data Engineering 2010.
- Sartori, G., Mameli, F., Polezzi, D. y Lombardi, L. (2006). "An ERP study of low and high relevance semantic features". *Brain Research Bulletin*. 69. Pg: 182–186.
- Sartori, G. y Lombardi, L. (2004). "Semantic relevance and semantic disorders". *Journal of Cognitive Neuroscience*. 16. Pg: 439–452.
- Shiliang, W; Shibo, W. (2004) "Application of AHP Method into synthetic evaluation of venture capital investment projects". *Enterprise Economy*. (4). Pp: 72-75
- Siddiqi, J; Shekaran , M (1996). "Requirement Engineering: The emerging wisdom". *IEEE Software*. 2. Pg: 15-19.
- Sierra, C., Jennings, N.R., Noriega, P. and Parsons, S. (1998) "A framework for argumentation-based negotiation". En Singh, R.A. and Wooldridge, M. (Eds.): *Intelligent Agent IV: 4th, International Workshop on Agent Theories, Architectures and Languages (ATAL – 1997)*, Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, Vol. 1365, pp.177–192.
- Sillence, J; Saeedi, M (1999) "Computer- mediated communication: problems and potentials of argumentation support systems". *Decision Support System*. 26. Pg: 287-306.
- Skowron, A; Ramanna, S; Peters, J (2006) "Conflict Analysis and Information Systems: A Rough Set Approach". *Proceedings of RSKT 2006*. Pg:233-240.
- Smith, E (1976) *Theories of semantic memory*. En W. K. Estes (Ed), *Handbook of learning and cognitive process*. 4. Pg: 67-8.
- Solomatine, D (2002) "Computational Intelligence Techniques in modeling water systems: some applications". *IEEE*. 6 (2). Pg: 1853-8.
- Sommerville, I (1996). *Requirement Engineering*. Addison-Wesley. Ed. 5ta. England, Wokingham.
- Sommerville, I ; Sawyer, P (1997) *Requirement Engineering – A good practice guide*. John Wiley and Sons, Chichester, UK.
- Straszecka, E. (2000). *Fuzzy systems in medicine*. En: Cap. *Defining Membership functions*. P. Szczepaniak, G. L. P.J y J. Kacprzyk, Ed. New York: Physica-Verlag. Pp: 32-47.
- Suarez Fernández, M (2007) "Estimación de la Información Mutua en problemas con datos imprecisos". Tesis Doctoral. Universidad Oviedo.
- Toulmin, S (1958) "The uses of arguments". Cambridge University Press, London.
- Triantaphyllou, E (2001) "Two new cases of rank reversal when the AHP and some of its additive variants are used that do not occur with the multiplicative AHP". *J Multi- criteria Decis Anal*. 10. (1) Pg: 11-25.

- Tulving, E (1972) Episodic and semantic memory. En Tulving y Donaldson (Eds). Organization of memory. New York: Academic Press. Pg: 381-403.
- Turksen, I; Tian, Y (1993) "Combination of rules of their consequences in fuzzy expert systems". Fuzzy Sets and Systems. 58. (1). Pg: 3-40.
- Vesanto, J ; Alhoniemi, E (2000) "Clustering of the Self-Organizing Map". IEEE Transactions on neural networks.11. (3). May.
- Vivas, J. (2009a). "Modelos de Memoria Semántica". En J. Vivas (Comp.). Evaluación de redes semánticas. Instrumentos y Aplicaciones". Mar del Plata: Eudem. ISBN 978-987-05-5903-0 (En Prensa).
- Vivas, J. (2009b). "La Centralidad Sociocognitiva y la evolución de las Redes Semánticas en la construcción de significados y la formación de consensos". En A. López Alonso (Comp.). Psicología Cognitiva y Ecológica: Principales cuestiones y distintas aplicaciones. Bs. As.: 2010. (En prensa).
- Vivas, J., Huapaya, C., Lizarralde, F., Arona, G., Comesaña, A. (2009). "La evaluación semántica en entornos de nuevas tecnologías de la información y la comunicación". En M. Concepción Rodríguez y V. Padilla Montemayor (Comp.) Interacción y medición en ambientes de aprendizaje electrónicamente mediados. Monterrey: Ed. Trillas. 92-111.
- Vivas, J.; Ledesma, R. (2009). "Evaluación de la memoria semántica. El método Distsem". En J. Vivas (Comp.). Evaluación de redes semánticas. Instrumentos y Aplicaciones. MDP: Eudem. ISBN 978-987-05-5903-0 (En Prensa).
- Vivas, J; Azzolini, M; Vivas, L (2010) "Buscador de Definidoras y VTV. Utilidad de algunas técnicas de generación de normas para la producción de atributos semánticos y la definición del campo semántico. VII Congreso Iberoamericano de Psicología. Oviedo. 22 de Julio de 2010.
- Vivas, J; Pazgón, E; Azzolini, M. (2010). "El significado de las emociones. Instrumentos de visualización y análisis". XII CONGRESO ARGENTINO DE NEUROPSIQUIATRÍA y Neurociencia Cognitiva 8° CONGRESO LATINOAMERICANO DE NEUROPSIQUIATRÍA XIII Jornadas de la Enfermedad de Alzheimer. 1 a 3 de septiembre de 2010.
- Wang, T; Chen, Y (2005). "A new method on decision-making using fuzzy linguistic assessment variables and fuzzy preference relations", The 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2005), July 10-13, 2005 - Orlando, Florida.
- Wiegers, K (1999). Software Requirements. Microsoft Press. Redmont. Washington.

- Wu, S; Banzhaf, W (2010) "The use of computational intelligence in intrusion detection systems : A review. Applied Soft Computing. 10. Pg: 1-35.
- Xie, G; Zhang, J; Lai, K; Yu, L (2008). "Variable Precision Rough Set Model for group decision – making: An application". International Journal of Aproximate Reasoning. 49. Pg: 331-343.
- Yager, R (1983)"Quantifiers in the Formulation of Multiple Objective Decision Functions". Information Sciences, (31). Pp: 107-139.
- Zadeh, L, (1965). "Fuzzy Sets". Information and Control (8).Pg:338-353.
- Zadeh, L, (1983). "A Computational Approach to Fuzzy Quantifiers in Natural Languages". Computers and Mathematics with Applications (9). Pp: 149-184.
- Zadeh, L.A. (1998) Soft computing, fuzzy logic and recognition technology. Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence. Volume: 2.pp: 1678 - 1679
- Zave, P; Jackson, M (1997) "Four Dark Corner of Requirements Engineering". ACM Transactions on Software Engineering and Methodology. 6. (1). Pg: 1-30.
- Zhang, Y; Harman, M; Finkelstein, M ; Mansouri, A (2011) "Comparing the Performance of Metaheuristics for the Analysis of Multi-Stakeholder Tradeoffs in Requirements Optimisation". Information and Software Technology. doi:10.1016/j.infsof.2011.02.001.
- Ziarko, W. (1993). "Variable Precision Rough Set Model". Computer System Science. 46. (1). Pg: 39-59.
- Ziarko, W. (2002). "Set approximation quality measures in the variable precision rough set model". In: Proceeding of Second International Conference on Hybrid Intelligence System (HIC´S 02). Soft Computing System. 87. Pg: 442-452.

Páginas WEB visitadas

- <http://www.wfmc.org/>
- <http://www.moreq2.eu/>
- <http://www.dlmforum.eu/>