



Universidad
Nacional de
La Plata

TESIS DOCTORAL
“PROCESO DE REQUISITOS
VALIDADO
EMPIRICAMENTE”

Gladys Noemí Kaplan

Director de tesis: *Prof. Ing. Jorge Horacio Doorn*

Año 2022

*Dedicado a la memoria de mi padre, quien
vio en mí un universo infinito.*

Agradecimientos

Cortázar decía que *las palabras nunca alcanzan cuando lo hay que decir, desborda el alma*. Sin lugar a dudas esta frase es lo que mejor representa este momento.

A la primera persona que quiero agradecer es a Jorge Doorn ¡Mi gran maestro! El hombre de la bala de plata. Un gigante que pone su hombro para que todos podamos ver más lejos. Docente e investigador a tiempo completo. *Gracias Jorge por tu inmensa generosidad, por tu extraordinario conocimiento que nunca dudas en compartir, por acompañarme en mi evolución, pero fundamentalmente, por tu afecto.*

A mis padres, les agradezco su confianza y su amor incondicional. Gracias mamá por estar siempre. A mi padre, que nos dejó físicamente hace poco más de un año, le dedico este trabajo realizado con mucho esfuerzo y amor. A él le debo gran parte de lo que soy. Me hizo creer en el respeto, la honestidad y la humildad. Su vocación fue la medicina, a la cual le dedicó su vida. Su entusiasmo por su profesión me incentivo a buscar mi destino, el que finalmente encontré en la docencia. *Gracias viejo por todo lo que me diste y espero que, donde estés, sientas ese orgullo que emana del alma como lo siento por vos.*

A mi hijo Francisco, luz eterna de mi vida, deseo que este trabajo le sirva como ejemplo de esfuerzo, perseverancia, determinación y pasión. No existe estímulo más grande que un hijo para lograr aquello que parece imposible. *Me siento bendecida por tenerte en mi vida y profundamente orgullosa. Sé que lograras todo lo que te propongas.*

Agradezco al Dr. Julio Leite y a la Dra. Graciela Hadad por muchos y valiosos años de trabajo en mis primeros pasos por la investigación. También a infinidad de colegas, docentes e investigadores, que siempre han confiado en mí.

Finalmente, mi profundo agradecimiento a la Universidad Nacional de La Matanza, particularmente al Departamento de Ingeniería e Investigaciones Tecnológicas (DIIT) por su inconmensurable generosidad y apoyo.

Resumen

La construcción del *Proceso de Requisitos basado en Escenarios*, tema central de la presente tesis, ha comenzado hace más de dos décadas. Durante este tiempo, se lo ha revisado en diferentes proyectos de investigación, mientras que simultáneamente se lo ha aplicado en numerosos casos. La detección de algunos problemas ha generado una alerta en la calidad y consistencia de los modelos construidos y, por consiguiente, de los requisitos del software obtenidos. Su estrategia de construcción consta de tres etapas: Comprender el Universo de Discurso actual, Planificar el Universo de Discurso futuro y Explicitar los Requisitos del Software. En la secuencia de ejecución del proceso se utilizan básicamente dos modelos: el Léxico Extendido del Lenguaje (LEL) y los Escenarios, cada uno con sus particularidades. Tanto el proceso como los modelos han sido analizados empíricamente en esta tesis. Parte de los resultados obtenidos consisten en reemplazar la *Actividad Derivar Escenarios Actuales* con un nuevo mecanismo cognitivo que ayuda a construir una primera versión de los Escenarios de mejor calidad, ya que la actividad existente genera Escenarios con problemas de consistencia y completitud. El principal origen de estos problemas ha sido ignorar que el LEL es declarativo y los Escenarios son procedurales. La nueva heurística utiliza un mecanismo procedural e incremental *por proximidad de situaciones*, y, además, se nutre de todo el conocimiento existente en el macrosistema. En forma análoga, se ha detectado que la *Heurística de Construcción del LEL*, la cual es conducida por una lista inicial de símbolos, dificulta la detección de nuevos símbolos, afectando la completitud del glosario. En este caso también se propone reemplazarla por una nueva heurística que utiliza la lista inicial a modo recordatorio y propone identificar los símbolos *por proximidad*, arrojando mejoras significativas en los LEL construidos. Otros cambios son transversales, como es la incorporación de las *jerarquías conceptuales* y de los *puntos de vista del contexto* ya que afectan tanto al LEL como a los Escenarios. De esta manera se incluye información sensible para los requisitos, incrementando el nivel de detalle y la precisión de todas las representaciones. Con el objetivo de reducir los errores involuntarios, durante la descripción del LEL y de los Escenarios, se propone una *Vista Clasificación*, que puede activarse por demanda, con información adicional de cada símbolo del LEL. Las restantes modificaciones son nuevos modelos agregados al proceso, el primero de ellos es la *Construcción del LEL de Requisitos*, lo que ha puesto de manifiesto una importante omisión en el proceso existente al no contemplar la evolución que sufre el léxico durante el proceso de Ingeniería de Requisitos. La utilización del LEL en los Escenarios Futuros y en el documento de especificación de requisitos de software, paradójicamente, es una nueva fuente de ambigüedades ya que el léxico de los clientes y usuarios no es apto para describir el proceso del negocio futuro con el sistema de software en ejecución. Finalmente, el segundo agregado está relacionado con aquella información que aparece espontáneamente, pero que no tiene cabida en el modelo que se está construyendo. Esta *información extemporánea* requiere ser resguardada en el momento que aparece para recuperarla oportunamente, asegurando su comprensión cuando llegue el momento de incorporarla a un modelo. A tal efecto, se describe un mecanismo que ayuda el tratamiento efectivo de este tipo de información en cualquier proceso de requisitos.

Todos los cambios y agregados al Proceso de Requisitos basado en Escenarios contribuyen a obtener una Especificación de Requisitos de Software de la mayor calidad posible, proporcionando modelos más completos y consistentes.

Abstract

The development of the *Scenario-based Requirements Process*, the main subject of this thesis, started two decades ago. Meanwhile, during that time, it has been reviewed in several research projects, while simultaneously it has been applied in many cases. The detection of some problems raised an alert on the quality and the evenness of the constructed models and, therefore, on the quality of the software requirements produced. Their construction process consists of three stages: comprehend the current context in which the future system will be introduced, plan how such context will behave in the future, and make explicit the software requirements. During the process, two models have used the Language Extended Lexicon and the Scenarios, each one with its distinctiveness. Both the process and the models has been empirically tested as a part of this thesis. Some of the obtained results comprise the substitution of the *Derive Actual Scenarios Activity* with a mechanism involving a new cognitive standpoint that helps to construct the first version of the Scenarios with better quality, improving the existing activity, which produces Scenarios with consistency and completeness problems. The main source of these problems has been disregarding that the LEL is declarative and the Scenarios are procedural. The new heuristic uses a procedural and incremental mechanism by *proximity of situations*, in addition besides it is fed by all the existent knowledge in the macro system. Likewise, it has been detected that the *LEL Construction Heuristic*, which is conducted by an initial list of symbols, difficult the detection of new symbols, harming the completeness of the glossary. In this case, it is also proposed to replace it with a new heuristic that uses the initial list as a reminder and recommends identifying the symbols by their proximity, leading to significant improvements in the LEL built. Some other changes introduced such as the incorporation of the *conceptual hierarchies* and the *context points of view* are of a wider scope since they affect the LEL and the Scenarios. Therefore, sensitive information for the requirements is added, incrementing the level of detail and the precision of all representations. In order to reduce unintentional errors during the description of the LEL and the Scenarios, a *Classification View* is proposed, it may be activated by demand giving additional information on each LEL symbol. The rest of the modifications are new models added to the process, the first one is the *Requirements LEL Construction*, which has revealed an important omission in the existent process, by not considering the evolution that suffers the lexicon along the Requirements Engineering process. The use of the LEL in the future Scenarios and the software requirements specification document, paradoxically, is a new source of ambiguities since the client's and user's lexicon is not apt to describe the future business process with the software system running. Finally, the second addition is related to the information that appears spontaneously but does not have room in the model that is currently being built. This *extemporaneous information* requires to be recorded when appears to be able to be recovered opportunely, allowing its comprehension when the moment of incorporating it to a new model comes. To that end, a mechanism, to help the treatment of this type of information in every requirement process, is described.

All changes and additions to the *Scenario-based Requirements Process* contribute to obtaining a Software Requirements Specification of the highest quality possible, providing more complete and consistent models.

Nota Preliminar	18
Capítulo 1. Introducción.....	22
Parte 1. Conceptos Generales	
Introducción	26
Capítulo 2. Análisis de Sistemas y la Ingeniería de Requisitos	
Resumen	28
Análisis de Sistemas.....	30
Ingeniería de Requisitos	35
Diferencia entre el Análisis de Sistemas y la Ingeniería de Requisitos	37
Capítulo 3. Contexto de la Ingeniería de Requisitos	
Resumen	41
Contexto de la Ingeniería de Requisitos.....	42
Relaciones de poder en el Proceso de Requisitos.....	43
Universo de Discurso	44
Evolución del Universo de Discurso	46
Límites del Universo de Discurso.....	48
Conocimiento del Contexto.....	49
Involucrados (Stakeholders).....	50
Fuentes de Información.....	52
Capítulo 4. Ingeniería de Requisitos	
Resumen	56
Requisito y Requerimiento	60
Requisitos del Software	61
Origen de los Requisitos	66

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Índice

El Lenguaje Natural en los Procesos de Requisitos	68
El Documento de Requisitos.....	70
Los Procesos de Requisitos.....	73
Priorización de los Requisitos	78
Rastreabilidad de los Requisitos.....	80
Capítulo 5. Reseña Histórica de la Ingeniería de Requisitos	
Resumen	86
¿La “crisis del software” se ha convertido en una enfermedad crónica?.....	87
Parte 2. Proceso de Requisitos basado en Escenarios	
Introducción	104
Capítulo 6. Proceso de Requisitos basado en Escenarios	
Resumen	108
Estrategia del Proceso de Requisitos.....	109
Verificación con formularios de inspección	112
Capítulo 7. Léxico Extendido del Lenguaje	
Resumen	117
Glosarios en la construcción de software	118
Léxico Extendido del Lenguaje (LEL).....	120
Proceso de Construcción del LEL.....	122
Capítulo 8. Escenarios	
Resumen	136
Estrategias de construcción de Escenarios.....	137
Modelo Escenario	137
Cómo pensar un Escenario	141
Jerarquias de Escenarios	146

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Índice

Operaciones estructurales de Escenarios	147
Integración de Escenarios	157
Capítulo 8.1. Construcción de Escenarios Actuales	
Resumen	164
Proceso de construcción de Escenarios Actuales.....	165
Capítulo 8.2. Construcción de Escenarios Futuros	
Resumen	182
Evolución de Escenarios	183
Proceso de construcción de Escenarios Futuros	184
Proceso Orientado a lo Procedural	188
Proceso Orientado a los Objetivos	189
Proceso Híbrido	190
Validación de Escenarios Futuros.....	191
Capítulo 9. Especificar Requisitos del Software	
Resumen	194
Especificación de Requisitos de Software	195
Estrategia para Asignar Prioridades	204
Parte 3. Estudio Empírico	
Introducción	209
Capítulo 10. Estudio Empírico del LEL	
Resumen	212
Estudios relacionados con la completitud del LEL	213
Factores Internos que afectan la calidad del LEL	219
Lista de símbolos candidatos	219
Limites del UdeD.....	220

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Índice

Empotramiento de símbolos	220
Patrones Sintácticos	225
Factores Externos que afectan la calidad del LEL.....	227
Discursos retóricos o artificiales de los clientes-usuarios	227
Estados	228
Puntos de Vista del Contexto	231
Jerarquías Conceptuales	234

Capítulo 11. Estudio Empírico de los Escenarios

Resumen	241
Factores Internos que afectan la calidad de los Escenarios.....	242
Lista inicial de Escenarios	243
Aumentar la expresividad de los episodios	244
Empotramiento en el título del Escenario	245
Uso correcto del LEL.....	245
Restricciones a los Actores	246
Factores Externos que afectan la calidad de los Escenarios	247
Estados.....	247
Puntos de Vista del contexto.....	249
Jerarquías Conceptuales.....	250

Parte 4. Cambios y agregados al Proceso

Introducción	254
--------------------	-----

Capítulo 12. Cambios y Agregados al LEL

Resumen	258
Síntesis de los Cambios y Agregados	259
Vista Clasificación	260

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Índice

Vista de Empotramiento.....	262
Puntos de Vista.....	264
Sinónimos.....	268
Homónimos.....	269
Tratamiento de estados.....	270
Jerarquías conceptuales.....	276
Evolución de los Patrones Sintácticos.....	282
Capítulo 13. Actualización de la Inspección del LEL	
Resumen.....	291
Cambios a la Inspección del LEL.....	292
Actualización de la Inspección del LEL.....	292
Detección de Homónimos.....	293
Detección de Jerarquías Incompletas.....	294
Trabajo Futuro.....	297
Capítulo 14. Construcción del LEL por Proximidad	
Resumen.....	299
Construcción del LEL por Proximidad.....	300
Heurística por Proximidad.....	303
Capítulo 15. LEL de Requisitos	
Resumen.....	315
Evolución del vocabulario.....	316
Estrategias de Construcción del LEL _R	317
Heurística de Construcción del LEL _R	320
Capítulo 16. Cambios y Agregados a los Escenarios	
Resumen.....	326
Síntesis de los Cambios y Agregados a los Escenarios.....	327

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Índice

Nueva Actividad Derivar	327
Vinculación con el LEL.....	328
Puntos de Vista	332
Importancia de las Jerarquías y los Puntos de Vista en los requisitos	334
Episodios Ubicuos.....	338
Restricciones a los Actores	339
Modificación del Modelo Escenarios	339
Capítulo 17. Actualización de la Inspección de Escenarios	
Resumen	343
Cambios en la Inspección “Consistencia entre Escenarios”	344
Actualización de la Inspección de Escenarios.....	344
Uso de jerarquías contextuales del contexto.....	344
Puntos de Vista en los Escenarios	346
Relación entre la inspección de EA y de los EF.....	347
Trabajo Futuro	348
Capítulo 18. Derivación de Escenarios Actuales por Proximidad	
Resumen	350
Nueva actividad Derivar	351
Heurística de Derivación.....	352
Capítulo 19. Información Extemporánea en los Procesos de Requisitos	
Resumen	366
IE en los Procesos de Requisitos.....	368
Información Adelantada (IA)	369
Información Tardía (IT)	372
IE en el Proceso de Requisitos basado en Escenarios	373
Ficha de Información Extemporánea (FIE)	377

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Índice

Registro, Tratamiento y Cierre de la FIE..... 381

Capítulo 20. Conclusiones

Consideraciones finales388

Trabajo futuro390

Anexo Inspecciones del LEL y Escenarios392

Referencias Bibliográficas436

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Índice de Figuras

FIGURA 1 – ESTRUCTURA DE LA TESIS.....	20
FIGURA 2 – RESOLUCIÓN DE PROYECTOS IT POR TAMAÑO DE PROYECTO (STANDISH GROUP 2011-2015) .	23
FIGURA 3- ESTADÍSTICAS DE EFECTIVIDAD DE PROYECTOS IT (STANDISH GROUP 1979)	33
FIGURA 4 – ESTADÍSTICAS DE EFECTIVIDAD DE PROYECTOS IT (STANDISH GROUP 2014).....	33
FIGURA 5 – ARTEFACTOS GENERADOS POR EL AS Y LA IR.....	38
FIGURA 6 – DISTANCIA AL DISEÑO DESDE EL AS Y LA IR.....	39
FIGURA 7 – RELACIÓN “CLIENTE – PROVEEDOR”	43
FIGURA 8 – RELACIÓN “CLIENTE-PROVEEDOR EXTERNO”	44
FIGURA 9 – CONTEXTO GENERAL Y UDED	46
FIGURA 10 – VARIABLES DEL CONTEXTO GENERAL Y DEL UDED	46
FIGURA 11 – RELACIÓN ENTRE LA EVOLUCIÓN DEL UDED Y LA IR.....	48
FIGURA 12 - PUNTOS DE VISTA Y PERSPECTIVAS DE LOS INVOLUCRADOS.....	53
FIGURA 13 - COSTO DE CORRECCIÓN DE LOS REQUISITOS [BOEHM81]	59
FIGURA 14 – CANALES DE COMUNICACIÓN EN EL PROCESO DE REQUISITOS	68
FIGURA 15 - USO DEL LN EN LOS DOCUMENTOS DE REQUISITOS [BERRY02]	69
FIGURA 16 – CONTENIDO DE UNA ERS [IEEE830]	73
FIGURA 17 – EL PROCESO DEL PROCESO DE REQUISITOS	74
FIGURA 18 – SADT DEL PROCESO DE REQUISITOS [LEITE01].....	77
FIGURA 19 – TIPOS DE RASTREABILIDAD DE REQUISITOS [GOTEL94].....	82
FIGURA 20 – DIMENSIÓN Y DIRECCIÓN DE LA RASTREABILIDAD DE REQUISITOS [WANG19].....	83
FIGURA 21 – EFECTIVIDAD DE PROYECTOS IT ENTRE 1994 Y 2015	88
FIGURA 22– MEDIAS GEOMÉTRICAS DE PROYECTOS IT ENTRE 1994 Y 2015	88
FIGURA 23 – MODELO PROPUESTO POR ROYCE.....	89
FIGURA 24 – MODELO ITERATIVO.....	93
FIGURA 25 – MODELO INCREMENTAL	94
FIGURA 26 – MODELO PROTOTIPADO EVOLUTIVO	95
FIGURA 27 – MODELO EN ESPIRAL.....	96
FIGURA 28 – DIFERENCIA ENTRE CASCADA Y SCRUM.....	98
FIGURA 29 – TASA DE ÉXITO POR TAMAÑO DE PROYECTOS (CASCADA Y ÁGILES)	101
FIGURA 30- ESTRATEGIA DEL PROCESO DE REQUISITOS BASADO EN ESCENARIOS	110
FIGURA 31 – PROCESO DE REQUISITOS BASADO EN ESCENARIOS.....	111
FIGURA 32 – PROCESO DE REQUISITOS BASADO EN ESCENARIOS ACTUALIZADO	113
FIGURA 33 – ACTIVIDAD VERIFICAR	114
FIGURA 34 – META MODELO LEL	121
FIGURA 35– JERARQUÍA INTRÍNSECA DEL LEL	122
FIGURA 36 – SADT DEL PROCESO DE CONSTRUCCIÓN DEL LEL	124

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Índice de Figuras

FIGURA 37- EJEMPLOS DE SÍMBOLOS DEL LEL (CASO SCC).....	127
FIGURA 38 - MODELO ESCENARIOS.....	139
FIGURA 39 – JERARQUÍA DE ESCENARIOS	146
FIGURA 40 – EJEMPLO DE ESCENARIO CON SUB-ESCENARIO (CASO SCC).....	147
FIGURA 41 – OPERACIÓN EMPOTRAR ESCENARIOS	150
FIGURA 42 –OPERACIÓN FACTORIZAR ESCENARIOS	151
FIGURA 43 – CASO ESPECIAL DE LA FACTORIZACIÓN ESCENARIOS	151
FIGURA 44 – OPERACIÓN CONSOLIDAR ESCENARIOS.....	153
FIGURA 45 – OPERACIÓN DIVIDIR ESCENARIOS	153
FIGURA 46 – OPERACIÓN FUSIONAR ESCENARIOS	154
FIGURA 47 – OPERACIÓN SEPARAR ESCENARIOS	155
FIGURA 48 - EJEMPLO DE LA OPERACIÓN FACTORIZAR ESCENARIOS (CASO SMS).....	156
FIGURA 49 - EJEMPLO DE LA OPERACIÓN CONSOLIDAR ESCENARIOS (CASO SMS).....	158
FIGURA 50 - EJEMPLO DE ESCENARIO INTEGRADOR (CASO SCC)	162
FIGURA 51– SADT DEL PROCESO DE CONSTRUCCIÓN DE ESCENARIOS.....	165
FIGURA 52 - SADT DE LA ACTIVIDAD DERIVAR ESCENARIOS	167
FIGURA 53 - EJEMPLO DE DERIVAR ESCENARIOS (CASO SCC)	168
FIGURA 54 - SADT DE LA ACTIVIDAD DESCRIBIR ESCENARIOS ACTUALES.....	170
FIGURA 55 - SADT DE LA ACTIVIDAD ORGANIZAR	172
FIGURA 56- VERIFICACIÓN DE ESCENARIOS	174
FIGURA 57– ABSTRACCIÓN Y CONCRECIÓN EN LA CONSTRUCCIÓN DE ESCENARIOS FUTUROS	184
FIGURA 58 - PROCESO DE CONSTRUCCIÓN DE ESCENARIOS FUTUROS.....	186
FIGURA 59 – EJEMPLO DE ESCENARIOS FUTURO (CASO SCC)	187
FIGURA 60 – PROCESO DE CONSTRUCCIÓN DE EF	190
FIGURA 61 – PROCESO DE CONSTRUCCIÓN DE EF CON INFORMACIÓN EXTEMPORÁNEA.....	191
FIGURA 62 - META-MODELO DE ESPECIFICACIÓN DE REQUISITOS.....	195
FIGURA 63 - SADT DEL PROCESO DE EXPLICITAR REQUISITOS DE SOFTWARE	196
FIGURA 64 - EJEMPLO DE LISTA DE REQUISITOS DE UN ESCENARIO FUTURO (CASO SOP)	199
FIGURA 65– CRONOLOGÍA DE LOS CASOS QUE SUSTENTAN EL ESTUDIO EMPÍRICO	209
FIGURA 66 – LISTA DE PROBLEMAS Y SOLUCIONES PUBLICADA EN 1996	215
FIGURA 67 – EJEMPLO DE SÍMBOLO CON EMPOTRAMIENTO APARENTE (CASO SNOR).....	221
FIGURA 68 - EJEMPLO MODIFICADO DE METONIMIA, PRÉSTAMO Y SINTAGMA	222
FIGURA 69 - EJEMPLO DE SÍMBOLOS CON EMPOTRAMIENTO (CASO SCC)	223
FIGURA 70 – CLASIFICACIÓN DE HOMÓNIMAS.....	225
FIGURA 71 – EJEMPLO DE UN SÍMBOLO BASE Y SUS ESTADOS (CASO SCC).....	229
FIGURA 72 – DER DE SÍMBOLOS ESTADOS	230
FIGURA 73 – EJEMPLO DEL LEL CON PUNTO DE VISTA (CASO SCC)	232

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Índice de Figuras

FIGURA 74 – EJEMPLO DE ESTADO Y JERARQUÍA DEL CONTEXTO.....	235
FIGURA 75 – ORDENAMIENTO DEL LÉXICO SIN Y CON JERARQUÍAS (CASO SNOR)	236
FIGURA 76 – TIPOS DE JERARQUÍAS CONCEPTUALES DEL CONTEXTO	239
FIGURA 77 – EJEMPLO DE ESCENARIOS CON USOS INCORRECTOS O DUDOSOS DEL LEL (CASO SMAC) ..	246
FIGURA 78 – RELACIÓN ENTRE ESTADOS DEL LEL Y LOS EA	247
FIGURA 79 – RELACIÓN ENTRE LOS ESTADOS DEL UDED ACTUAL Y DEL UDED FUTURO	248
FIGURA 80 – MODELO DER DE LA RELACIÓN “PUNTOS DE VISTA- MODELOS”	249
FIGURA 81 – TRATAMIENTO DE LOS PUNTOS DE VISTA DEL CONTEXTO EN EL PROCESO DE REQUISITOS..	250
FIGURA 82 – EJEMPLO DE UNA JERARQUÍA LÉXICA OMITIDA (CASO SNOR)	251
FIGURA 83 – EJEMPLO DE USO INCORRECTO DE UNA JERARQUÍA LÉXICA (CASO SNOR)	252
FIGURA 84 – SÍNTESIS DE LOS CAMBIOS Y AGREGADOS AL LEL.....	259
FIGURA 85 – EJEMPLO DE IMPACTOS CON PUNTOS DE VISTA.....	266
FIGURA 86 – EJEMPLO DEL USO SUGERIDO DE SINÓNIMOS (CASO SCC).....	268
FIGURA 87 – EJEMPLOS DE USO DE HOMÓNIMOS (CASO SCC)	270
FIGURA 88 – EJEMPLOS DE DEFINICIÓN DE ESTADO (CASO SNOR)	274
FIGURA 89 – EJEMPLO DE UN SÍMBOLO BASE Y SUS ESTADOS (CASO SNOR)	274
FIGURA 90 – EJEMPLO DE UNA JERARQUÍA CONCEPTUAL DE VERBOS (MERELOGÍA).....	277
FIGURA 91 – EJEMPLO DE UNA JERARQUÍA CONCEPTUAL DE OBJETOS (TAXONOMÍA) (CASO SNOR)	277
FIGURA 92 – EJEMPLO DE USO DE JERARQUÍAS EN EL LEL (CASO SNOR).....	278
FIGURA 93 - POSIBLES LECTURAS DE UNA JERARQUÍA CONCEPTUAL EN EL LEL	279
FIGURA 94- IDENTIFICACIÓN DE LA JERARQUÍA CONCEPTUAL (CASO SNOR).....	281
FIGURA 95 – EJEMPLO DE LA VISTA CLASIFICACIÓN PARA JERARQUÍAS CONCEPTUALES (CASO SNOR)..	282
FIGURA 96 – EJEMPLO DE UN SÍMBOLO OBJETO CON DIFERENTES PATRONES SINTÁCTICOS (CASO SNOR)	
.....	284
FIGURA 97 - EJEMPLO DE UN SÍMBOLO SUJETO CON DIFERENTES PATRONES SINTÁCTICOS (CASO SNOR)	285
FIGURA 98 – EJEMPLO DE UN SÍMBOLO VERBO CON DIFERENTES PATRONES SINTÁCTICOS (CASO SNOR)	287
FIGURA 99 - EJEMPLOS DE SÍMBOLOS ESTADOS CON DIFERENTES PATRONES SINTÁCTICOS (CASO SNOR)	287
FIGURA 100 – EJEMPLO DE JERARQUÍAS CONCEPTUALES DEL CONTEXTO (CASO SNOR).....	288
FIGURA 101 – FUENTES DE INFORMACIÓN EN LA CONSTRUCCIÓN DEL LEL POR PROXIMIDAD.....	301
FIGURA 102 – RELACIÓN ENTRE LAS DIFERENTES FUENTES DE INFORMACIÓN	303
FIGURA 103 –HEURÍSTICA DEL CONSTRUCCIÓN DEL LEL POR PROXIMIDAD	305
FIGURA 104 – EJEMPLO DE LA LISTA DE SÍMBOLOS EN LA CONSTRUCCIÓN DEL LEL POR PROXIMIDAD ...	307
FIGURA 105 – EJEMPLO DE MAPA DE RELACIÓN ENTRE LOS ROLES Y LAS PERSONAS	309
FIGURA 106 – EJEMPLO DE LISTA DE SÍMBOLOS DEL LEL POR JERARQUÍAS ORGANIZACIONALES	312
FIGURA 107 – EVOLUCIÓN DEL LÉXICO DEL DOMINIO DURANTE LA CONSTRUCCIÓN DEL SOFTWARE.....	316
FIGURA 108 – EVOLUCIÓN DEL LEL	317
FIGURA 109 – ENFOQUES DE CONSTRUCCIÓN DEL LEL _R	320

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Índice de Figuras

FIGURA 110– SÍMBOLO DEL LEL CON PUNTO DE VISTA	323
FIGURA 111 – CAMBIOS Y AGREGADOS A LOS ESCENARIOS	327
FIGURA 112 – EJEMPLO DE JERARQUÍA DEL CONTEXTO.....	329
FIGURA 113 – EJEMPLO VISTA CLASIFICACIÓN PARA DETECTAR ERRORES DE VINCULACIÓN EN EL LEL .	329
FIGURA 114 – EJEMPLO DE LA VISTA CLASIFICACIÓN PARA IDENTIFICAR ESTADOS (CASO SCC)	331
FIGURA 115 – EJEMPLOS DE IMPACTOS CON PUNTOS DE VISTA	332
FIGURA 116 – REPERCUSIÓN DE LOS PUNTOS DE VISTA EN LOS REQUISITOS DEL SOFTWARE.....	335
FIGURA 117 – EJEMPLO DE JERARQUÍAS OMITIDAS Y SU REPERCUSIÓN EN LOS REQUISITOS (CASA SMAC)	337
FIGURA 118 – EJEMPLO DE SÍMBOLO DEL LEL CON JERARQUÍA NO DETECTADA (CASO SMAC)	337
FIGURA 119 - EJEMPLO DE UN EPISODIO UBICUO	339
FIGURA 120 – MODELO ESCENARIOS ACTUALIZADO.....	341
FIGURA 121 – MARCO CONCEPTUAL DE LA ACTIVIDAD DERIVAR.....	353
FIGURA 122 – ESTRATEGIA DE DERIVACIÓN DE ESCENARIOS POR PROXIMIDAD	353
FIGURA 123 – HEURÍSTICA DE DERIVACIÓN DE ESCENARIOS POR PROXIMIDAD	354
FIGURA 124 – EJEMPLO DE DERIVACIÓN DE EA POR PROXIMIDAD DESDE EL LEL	357
FIGURA 125 – EJEMPLO DE DERIVACIÓN DE EA POR PROXIMIDAD DESDE DOCUMENTOS	359
FIGURA 126 – EJEMPLO SÍMBOLO VERBO QUE INICIA LA DERIVACIÓN POR PROXIMIDAD (CASO SNOR)..	361
FIGURA 127 – EJEMPLO DE UN ESCENARIO DERIVADO POR PROXIMIDAD (CASO SNOR)	362
FIGURA 128 – EJEMPLO DE ESCENARIOS GENERADOS POR PROXIMIDAD (CASO SNOR)	364
FIGURA 129 – EJEMPLO DE ESCENARIO DERIVADO POR PROXIMIDAD MODIFICADO (CASO SNOR)	364
FIGURA 130– CANTIDAD DE IE EN LAS ETAPAS DEL PROCESO DE REQUISITOS	368
FIGURA 131 –IA SEGÚN LA EXPECTATIVA DEL CLIENTE-USUARIO	371
FIGURA 132 – INFORMACIÓN ADELANTADA EN EL PROCESO DE REQUISITOS	373
FIGURA 133 – INFORMACIÓN TARDÍA	374
FIGURA 134 - IA SURGIDA DURANTE LA CONSTRUCCIÓN DEL LEL (CASO SGA).....	375
FIGURA 135 – EJEMPLO DE IA SURGIDA DURANTE LA CONSTRUCCIÓN DE UN EA (CASO SGA)	375
FIGURA 136 – EJEMPLO DE EF GENERADO A PARTIR DE IA (CASO SGA).....	376
FIGURA 137 – IA COMO RESTRICCIÓN DE UN RECURSO DE UN EF Y COMO RNF	377
FIGURA 138 - RNF DETECTADO DESDE UNA IA.....	377
FIGURA 139 - DATOS DE APERTURA DE LA FIE.....	382
FIGURA 140 - DATOS DE CIERRE DE LA FIE	383
FIGURA 141 – EJEMPLO DE FRAGMENTACIÓN DE IE	383
FIGURA 142 – EJEMPLO DE DISPERSIÓN DE IE	384
FIGURA 143 – HEURÍSTICA DE TRATAMIENTO DE LA IA	385
FIGURA 144 – HEURÍSTICA DE CIERRE DE LA FIE	386

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Índice de Tablas

TABLA 1 – PUBLICACIONES RELACIONADAS CON LA PARTE 1	26
TABLA 2 – IMPACTO DEL PARADIGMA ESTRUCTURADO EN EL ANÁLISIS Y DISEÑO	30
TABLA 3 –IMPACTO DEL PARADIGMA OO EN LAS ETAPAS DE ANÁLISIS Y DISEÑO.....	31
TABLA 4 – PONDERACIÓN DE LA DISTANCIA DESDE LA IR Y EL AS AL DISEÑO.....	37
TABLA 5 - MATRIZ DE INTERÉS [GARDNER86].....	54
TABLA 6 – ACTIVIDADES DE LOS PROCESOS DE REQUISITOS	75
TABLA 7– CONTEXTO Y EVOLUCIÓN DE LA CONSTRUCCIÓN DEL SOFTWARE.....	100
TABLA 8 – PUBLICACIONES RELACIONADAS CON LA PARTE 2.....	105
TABLA 9 – PROYECTOS DE INVESTIGACIÓN RELACIONADOS CON LA PARTE 2.....	105
TABLA 10 – PATRONES SINTÁCTICOS DEL LEL [KAPLAN00]	125
TABLA 11– TAXONOMÍA DE DEFECTOS DEL LEL.....	129
TABLA 12 - TIPOS DE REQUISITOS EXTRAÍDOS DE ESCENARIOS FUTUROS	200
TABLA 13 – PROYECTOS DE INVESTIGACIÓN RELACIONADOS CON LA PARTE 3.....	210
TABLA 14 – RESOLUCIÓN DE LOS PROBLEMAS EN EL PROCESO DE CONSTRUCCIÓN DEL LEL EXISTENTE .	216
TABLA 15 – RESULTADOS DEL ESTUDIO [DOORN03].....	217
TABLA 16– RESULTADOS DE LOS LEL DEL SISTEMA DE BIBLIOTECA	218
TABLA 17 - COMPARACIÓN ENTRE LA LISTA INICIAL DE SÍMBOLOS DEL LEL Y LA FINAL	220
TABLA 18 – EJEMPLO DE USO DE EMPOTRAMIENTO (CASO SNOR).....	222
TABLA 19 – EJEMPLOS DE EMPOTRAMIENTOS QUE DEBEN SER ANALIZADOS EN EL UDeD	224
TABLA 20 – EVOLUCIÓN DEL COMPONENTE NOCIÓN	226
TABLA 21 – EVOLUCIÓN DEL COMPONENTE IMPACTO.....	227
TABLA 22 – CAMBIOS Y AGREGADOS AL PROCESO DE REQUISITO.....	254
TABLA 23 – CAMBIOS Y AGREGADOS AL LEL.....	254
TABLA 24 – CAMBIOS Y AGREGADOS A LOS ESCENARIOS	255
TABLA 25 – PUBLICACIONES RELACIONADAS CON LA PARTE 4.....	256
TABLA 26 – PATRONES SINTÁCTICOS DEL LEL ACTUALIZADOS.....	283
TABLA 27 – ACTUALIZACIÓN A LA INSPECCIÓN DEL LEL.....	292
TABLA 28 – RELACIÓN ENTRE EL ESTUDIO EMPÍRICO DE LOS ESCENARIOS Y LAS SOLUCIONES.....	330
TABLA 29 – NUEVOS FORMULARIOS PARA LA INSPECCIÓN DE ESCENARIOS	344
TABLA 30 – CUBRIMIENTO DE LA INSPECCIÓN DE EA EN LOS EF.....	347
TABLA 31 – NUEVOS FORMULARIOS DE INSPECCIÓN PARA EF	348
TABLA 32 - PRINCIPALES SITUACIONES DONDE APARECE IA.....	370
TABLA 33 - PRINCIPALES FACTORES DE CAMBIO QUE INFLUYEN EN LA APARICIÓN DE IA	372
TABLA 34 – DATOS REQUERIDOS PARA LA FIE	379

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Nota Preliminar

El Proceso de Requisitos que se describe en la presente tesis tuvo su punto de partida en trabajos llevados a cabo a fines del siglo anterior y ha sido objeto de varias revisiones desde ese momento. Sobre el mismo proceso se han llevado a cabo numerosos proyectos de investigación en universidades tales como la Universidad de Belgrano (UB), la Universidad Tecnológica Nacional (UTN-FRBA), la Universidad Nacional de La Matanza (UNLaM), la Universidad Nacional del Centro de la Provincia de Buenos Aires (UNICEN), la Universidad Nacional de Luján (UNLu), la Universidad Nacional del Oeste (UNO), la Universidad de La Plata (UNLP) y la Pontificia Universidad Católica de Río de Janeiro (PUC RIO). También se ha realizado transferencia en aula en la UNLaM (grado y posgrado), en la UNSJ (posgrado) en la UNO (grado y posgrado), en la UTN-FRBA (grado y posgrado), en la UNICEN (grado), en la UNLP (posgrado), en la PUC RIO (grado), en la Universidad Católica (UCA) (posgrado) y en la UTN-FRCU (posgrado) desde el año 1995 a la fecha. Del Proceso de Requisitos emergieron varias tesis de maestría y doctorado. La autora de la presente tesis ha participado en varios de los trabajos mencionados junto a otros investigadores, y en otros ha colaborado de manera indirecta. Esta activa participación en la construcción del proceso permite presentarlo como un aporte a la disciplina. Por otro lado, su amplia experiencia en el uso junto al estudio empírico presentado en esta tesis, le ha permitido identificar defectos y debilidades que disminuyen sensiblemente la calidad de los requisitos obtenidos. De esta manera se presenta el Proceso de Requisitos original y los cambios y agregados que permiten obtener un conjunto de requisitos más completos y reales.

El contenido de la presente tesis se ha organizado en cuatro Partes:

- ✓ Parte 1. Conceptos Generales
- ✓ Parte 2. Proceso de Requisitos basado en Escenarios
- ✓ Parte 3. Estudio Empírico
- ✓ Parte 4. Cambios y Agregados al Proceso

La “**Parte 1. Conceptos Generales**”, que incluye la presente nota, contiene un análisis para dilucidar la confusión bastante difundida acerca de las similitudes y diferencias entre la Ingeniería de Requisitos y el Análisis de Sistemas. También, se definen algunos aspectos del contexto de la Ingeniería de Requisitos. Luego, se incluye un resumen sobre el estado del arte de la mencionada disciplina donde se evalúan algunos aspectos poco

considerados en la literatura referidos a la evolución de la construcción del software hasta arribar a la Ingeniería de Requisitos.

La “**Parte 2. Proceso de Requisitos basado en Escenarios**” describe el Proceso de Requisitos completo, con sus etapas, métodos, modelos y heurísticas. Abarca desde la primera actividad hasta la Especificación de los Requisitos del Software (ERS). Esta Parte en particular cumple un doble rol. Por un lado, es marco teórico, pero también es uno de los aportes de la presente tesis.

Luego, en la “**Parte 3. Estudio Empírico**” se presenta un estudio empírico de los modelos Léxico Extendido del Lenguaje (LEL) y Escenarios, descritos en la Parte 2. En el mismo se describen las debilidades existentes en el proceso cuyas mejoras son el núcleo de la presente tesis. Este estudio se sustenta en una revisión de una base de más de 200 casos generados por dos décadas.

En la “**Parte 4. Cambios y Agregados al Proceso**”, se describen las modificaciones al Proceso de Requisitos que solucionan o atemperan las debilidades indicadas. Muchas de estas modificaciones han sido probadas en casos reales mientras que otras son propuestas que se han mostrado eficaces en pruebas preliminares. Entre los cambios más significativos se encuentran dos conceptos transformadores: las jerarquías conceptuales y los puntos de vista¹ del contexto. Otros cambios, como es el caso de la Construcción del LEL y la Derivación de Escenarios Actuales, se han resuelto reemplazando la actividad por otras nuevas. Los agregados al proceso son la Construcción del LEL de Requisitos y el tratamiento de la Información Extemporánea. Este último tema fue presentado con anterioridad en la tesis de Maestría de la tesista.

La Figura 1 muestra la relación de las cuatro Partes, pudiéndose observar la centralidad de la Parte 2, el Proceso de Requisitos basado en Escenarios. Este gráfico se ha incluido en la caratula de cada Parte con el objetivo de orientar al lector. Nuevamente, a efectos de mejorar la comprensión de la presente tesis, se ha incorporado en cada Parte una *Introducción*, la cual contiene información para contextualizar lo que se va a leer (publicaciones, proyectos de investigación, casos de estudio y comentarios generales). También, antes de comenzar cada Capítulo, se incluye un pequeño resumen que otorga al lector una visión rápida de su contenido.

¹ En la presente tesis se utiliza el término para referirse a los *puntos de vista estructurados* ya que hacen referencia a una concepción particular del contexto, como es el punto de vista del “deber ser”, “es”, entre otros.

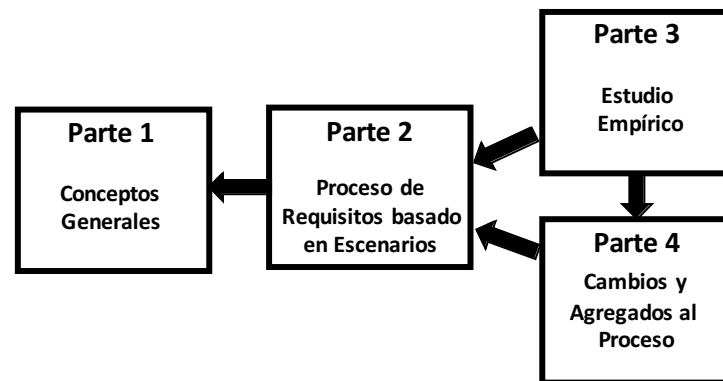


Figura 1 – Estructura de la tesis

Debido a la amplitud del trabajo presentado, es importante aclarar que la presente tesis contiene segmentos tomados de otros documentos, artículos, capítulos, tesis, etc. donde la tesista ha participado. Entre los más destacados se encuentran el *Apunte de Clase* [Kaplan06] y la tesis de maestría [Kaplan14a].

Capítulo 1

Introducción

La Ingeniería de Requisitos es una disciplina ampliamente reconocida por su importancia en el proceso de construcción del software. Este reconocimiento proviene de la necesidad de definir cuidadosamente los servicios que deberá prestar el nuevo Sistema de Software. Aunque el problema de cómo establecer sistemas no es nuevo, la comunidad científica de computación, principalmente la comunidad de Ingeniería de Software, debió prestar una especial atención a la tarea de definición de sistemas de software. Puede observarse que el término “Ingeniería de Requisitos” se acuñó por primera vez en 1977 en IEEE Transactions on Software Engineering. Luego, se realizó la primera conferencia internacional sobre este tema en 1993 (RE 93) y en 1995 se realizó la primera reunión del grupo de trabajo IFIP 2.9 (Ingeniería de Requisitos de Software).

La principal dificultad de la Ingeniería de Requisitos (IR) es la gran diversidad de enfoques existentes para construir sistemas de software. A continuación, para ejemplificar, se mencionan brevemente algunos métodos, técnicas y estrategias.

En 1977, Chuck Morris y Tony Crawford de IMB desarrollaron un método para determinar los requisitos de un sistema de información basado en técnicas de talleres conocido como Diseño de Aplicación Conjunta (JAD por sus siglas en inglés). En estos talleres o reuniones el interés estaba en las personas, enfatizando el desarrollo participativo entre clientes, usuarios y desarrolladores. Para Wood [Wood89] [Wood95] JAD permite diseñar sistemas de calidad en un 40% menos de tiempo.

En 1987, se publicó el método de Sala Limpia (“Cleanroom”) [Mills87] que es un enfoque formal. El propósito es reducir a cero la ambigüedad y maximizar la precisión de la especificación de requisitos de software (ERS). Este enfoque puede asegurar que el Sistema de Software satisface la ERS, pero no aseguran que satisfacen las necesidades de los clientes-usuarios.

Wirfs-Brock et al. propusieron el método dirigido por responsabilidades [Wirfs-Brock95], que es una metodología de análisis y diseño orientada a objetos (ADOO). Esta se diferencia de las metodologías tradicionales orientadas a objetos en que analiza el contexto donde viven los objetos, mientras que las demás analizan directamente los objetos.

Se puede observar que en los métodos descritos existe una clara tendencia a asegurar los servicios del software más que concentrarse en satisfacer las necesidades reales de los clientes y usuarios.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Introducción

Yu et al. propusieron el enfoque i* [Yu95] [Yu97] el cual realiza diferentes modelos orientados a comprender la organización y los objetivos del negocio para definir requisitos que soporten esos objetivos, para asignar responsabilidades en el sistema, para resolver conflictos de puntos de vista, etc.

Otra estrategia es el uso de Escenarios, los cuales describen situaciones del contexto [Zorman95] [Breitman01]. En este modelo, los requisitos del software están empotrados en los mismos Escenarios, por lo tanto, pueden ser incluidos directamente en un documento de requisitos o generar un documento diferente. Una ventaja de este modelo es que utiliza el lenguaje natural para su representación, lo que permite la cooperación de los involucrados durante toda la Ingeniería de Requisitos.

Esta enumeración puede continuar, lo que muestra que a pesar de existir innumerables esfuerzos para mejorar la calidad de los sistemas de software y de la gestión de sus proyectos, los resultados obtenidos se pueden considerar aceptables solo para proyectos pequeños (ver Figura 2). En este caso el cociente entre los proyectos exitosos y fallidos es 6 a favor de los proyectos exitosos, mientras que, en los otros casos, este cociente decrece de manera inversamente proporcional al tamaño del proyecto, siendo de 1 en proyectos moderados, de 1/3 en los medianos, 1/4 en los grandes y de 1/8 para los muy grandes.

	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medlum	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	100%	100%	100%

The resolution of all software projects by size from FY2011–2015 within the new CHAOS database.

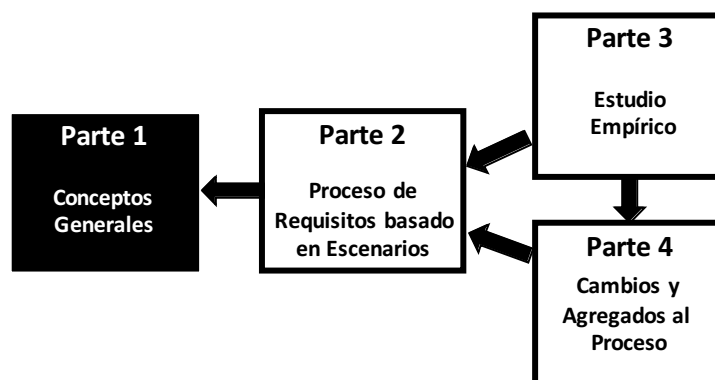
Figura 2 – Resolución de Proyectos IT por tamaño de proyecto (Standish Group 2011-2015)

Es necesario considerar que la Figura 2 no incorpora elementos cualitativos que permitan analizar los orígenes de estos resultados, pero fuerzan a la Ingeniería de Software a analizar estos datos y proponer mecanismos que permitan revertir esta situación. En este marco, el Proceso de Requisitos basado en Escenarios se torna relevante.

Finalmente, considerando que los procesos de requisitos suelen desarrollarse concurrentemente con las mejoras al proceso del negocio, es necesario revisar cómo las organizaciones se están adaptando a los nuevos desafíos. Los cambios tecnológicos y el dinamismo de los contextos están obligando a muchas compañías a modificar la forma de gestionar su organización. En el marco de la presente tesis, es necesario analizar dos tipos básicos de estructuras organizacionales: la vertical y la horizontal. Una *organización horizontal* tiene una gestión por procesos y gestiona equipos. Mientras que una *organización vertical* se basa en la estructura jerárquica piramidal, centrando la responsabilidad en la cima de la misma con una estructura por departamentos funcionales, donde la representación gráfica es escalonada de arriba hacia abajo. El pasaje de una estructura organizacional a otra es complejo, por lo que hoy en día existen organizaciones mixtas, donde se combina la estructura vertical con la horizontal.

Parte 1

Conceptos Generales



Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Introducción

Antes de describir algunos conceptos y particularidades acerca de la IR es necesario comprender dónde y por qué se distancia del Análisis de Sistemas (AS). Estas diferencias resultan esenciales para que el ingeniero/a de requisitos comprenda cuál es su rol y cómo debe afrontar un proceso de requisitos. En la Tabla 1 se incluye una publicación que fue el resultado de haber analizado específicamente este tema y otra que permite comprender las diferentes formas de elicitar conocimiento en el marco de la IR.

Año	Título	Autores	Lugar	Tipo	Cita
2014	Elicitación de Conocimiento Guiada por Modelos	Kaplan G., Doorn J. y Gigante N.	CACIC 14	Congreso nacional	[Kaplan14b]
2019	Similitudes y Diferencias entre el Análisis de Sistemas y la Ingeniería de Requisitos	Kaplan G., Doorn J.	REDDI- UNLaM Revista Digital del Dpto. de Ingeniería e Investigaciones Tecnológicas	Revista nacional con referato	[Kaplan19]

Tabla 1 – Publicaciones relacionadas con la Parte 1

Capítulo 2

Análisis de Sistemas e Ingeniería de Requisitos

Resumen

Una parte importante de las fallas en los sistemas de software se originan por defectos en los requisitos. Muchas de estas fallas provienen de la falta de comprensión acerca de cómo abordan el Análisis de Sistemas y la Ingeniería de Requisitos la construcción de los mismos. Ambas disciplinas afrontan la misma problemática dentro de la construcción del software y comparten algunos métodos, herramientas y técnicas. A la hora de decidir cuál de ellas es la más adecuada para especificar los requisitos del software, se presentan ciertas dudas que se aclaran en el presente Capítulo al analizar sus particularidades.

A pesar del tiempo transcurrido y de toda la actividad académica y profesional referida al desarrollo de sistemas de software y particularmente a la etapa de construcción de los requisitos del software, todavía es difícil encontrar una definición clara y precisa que establezca los límites y las diferencias entre el Análisis de Sistemas (AS) [DeMarco78] [Yourdon78] [Yourdon88] [Seen91] [Kendall97] y la Ingeniería de Requisitos (IR) [Davis93] [Loucopoulos95] [Kotonya98] [Leite94] [Thayer97] [Zave97]. Si bien ambas tienen el mismo objeto de estudio y el mismo objetivo, utilizan diferentes puntos de vista y diferentes formas de representación. Diferentes puntos de vista porque en el Análisis de Sistemas se atienden, por ejemplo, los datos o las entidades participantes, mientras que la Ingeniería de Requisitos presta atención a objetivos, conductas y necesidades. Diferentes formas de representación, porque el Análisis de Sistemas utiliza modelos cercanos al diseño y la Ingeniería de Requisitos utiliza modelos cercanos a los clientes y usuarios. De esta manera el Análisis de Sistemas privilegia la especificación de los servicios del sistema a ser desarrollado utilizando modelos o construcciones que faciliten la comprensión por parte de los desarrolladores. Esto simplifica enormemente la actividad del diseño, pero mutila la comprensión del usuario inhibiendo la percepción de todas las consecuencias que ocurrirán cuando el Sistema de Software se ponga en producción. Por otro lado, la Ingeniería de Requisitos privilegia la especificación de los servicios del sistema a ser desarrollado utilizando modelos o construcciones que faciliten la comprensión por parte de los clientes y usuarios. Esto dificulta en forma sensible la actividad de diseño. Naturalmente que son posibles infinidad de situaciones intermedias, pero siempre existirá la inexorable tensión entre facilitar un punto de vista o el otro.

En este sentido, merece transcribirse lo dicho en [Castro01] donde se afirma que “*el entorno operativo se entiende en términos de actores, responsabilidades, objetivos, tareas y recursos, mientras que el sistema de información en sí mismo se concibe como una colección de (software) módulos, entidades (objetos, agentes), estructuras de datos e interfaces. Esta la falta de coincidencia es uno de los principales factores para la mala calidad de los sistemas de información y la falla frecuente de los proyectos*”. Esta descripción marca una clara diferencia conceptual entre el macrosistema en estudio y los modelos utilizados, pero no profundiza en las causas y consecuencias.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 2 - Análisis de Sistemas e Ingeniería de Requisitos

Análisis de Sistemas

Desde una mirada más metodológica que estratégica, se puede observar que el Análisis de Sistemas ha tomado poca distancia del diseño y del código, provocando una suerte de simbiosis metodológica en pos de una mayor calidad en el producto software a construir, calidad entendida en el contexto del sistema en sí propio, es decir procurando obtener un producto robusto y fácil de mantener.

Dado que el Análisis de Sistemas aspira a facilitar el diseño y la programación, entonces esta última ha ido influyendo sobre aquel en forma progresiva. Esto ha sucedido desde épocas muy tempranas, haciendo que la programación presione metodológicamente a las etapas más tempranas del desarrollo. Esta influencia puede observarse en la Tabla 2 en relación con el paradigma estructurado, en la que sólo se tomaron los hitos más importantes.

Año	Programación	Análisis	Diseño
1966	[Boehm66]		
1967	[Dijkstra67]		
1974	Pascal N. Wirth		
1975			[Yourdon75]
1976		[Ross76]	
1977		[Gane77]	
1978		[DeMarco78]	
1979		[Yourdon79]	
1983		[Davis83]	
1984		[McMenamin84]	
			[Wetherbe84]
1987		[FitzGerald 87]	
1988		[Yourdon88]	[Meilir88]
			[Martin88]
1989		[Marca89]	
...

Tabla 2 – Impacto del paradigma Estructurado en el Análisis y Diseño

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 2 - Análisis de Sistemas e Ingeniería de Requisitos

Nuevamente, con un paradigma utilizado más recientemente como es el OO, sucedió algo similar al estructurado (ver Tabla 3), donde los métodos de programación influenciaron el diseño, el cual lo hizo más tarde con el análisis.

Año	Programación	Análisis	Diseño
1962	[Simula67] ²		
1972	Smalltalk72		
1980	Smalltalk80, C++, ...		
1990			[Wirfs-Brock90]
1991		[Coad91]	
			[Rumbaugh91]
1992	[Sessions92]	[Jacobson92] [Martin92]	
		[Bourne92]	
1993		[Martin93]	
1994		[Booch94]	
		[Graham94]	
1996			[Riel96]
1998		[Rup98]	
...

Tabla 3 –Impacto del paradigma OO en las etapas de Análisis y Diseño

En el paradigma Estructurado se conceptualiza el sistema en términos de funciones y procesos, input y output y en el OO el desarrollador piensa en términos de objetos, clases, métodos, herencia, etc. En ambos existe una distancia conceptual muy importante en el cómo se ve el contexto y en el cómo se lo modela [Castro01] y ambos con la forma de observar el contexto de la Ingeniería de Requisitos.

La propagación de los paradigmas desde las etapas avanzadas del desarrollo hacia las más tempranas, es una consecuencia de la intención del Análisis de Sistemas de facilitar las etapas siguientes sin tener en cuenta las consecuencias que esto provoca. Para ponderar apropiadamente esta inclinación del AS, es necesario comprender en profundidad que significa definir los requisitos de un Sistema de Software. Esta tarea consiste en pensar si

² * http://staff.um.edu.mt/jskl1/talk.html#History_67

un cierto servicio del Sistema de Software es el más apropiado para el desempeño de las actividades de los seres humanos realizando actividades en pos de sus propios objetivos. No es ninguna de las personas que desarrollan el software suficientemente idóneo para evaluar cuan apropiado resulta una conducta de un sistema en el contexto de las actividades de los clientes y usuarios. Son ellos quienes pueden hacer esa evaluación correctamente. El rol del profesional de la informática es el de ofrecer la información acerca de las posibles prestaciones y de su viabilidad en términos de costo y eficiencia. La programación ha presionado tanto al diseño como al análisis para que se adapte a su forma de trabajar, siempre con el argumento que la concurrencia de paradigmas mejora la calidad del software. Esto es cierto cuando el proceso considera más importante construir un software de la mayor calidad posible (en sus propios términos) que en satisfacer lo más apropiadamente las necesidades del cliente, como sucede con el AS. En este contexto surge inevitablemente la siguiente pregunta:

¿Qué habilita a propagar un paradigma exitoso en una etapa a otras?

¡Nada! Aun así, se deben considerar las ventajas y los inconvenientes. La ventaja es que mejora la calidad del proceso en el sentido que el software se construye mejor desde el punto de vista intrínseco y el inconveniente es que los servicios pueden no corresponder con las expectativas del cliente y usuario. Este inconveniente es tan importante que se ha transformado en una dificultad en muchos proyectos de software. Tal es así que varios autores han estudiado este problema y han realizado estadísticas referidas a los fracasos en proyectos de software [Alford79] [Boehm81] [GAO92] [Faulk92] [Lutz93] [CHAOS95] previo a la denominación de Ingeniería de Requisitos como tal (ver Figura 3), y en todos los casos se concluye que el principal origen a estos problemas está referido a requisitos inadecuados, muchos cambios en los requisitos, requisitos mal comprendidos y requisitos incompletos.

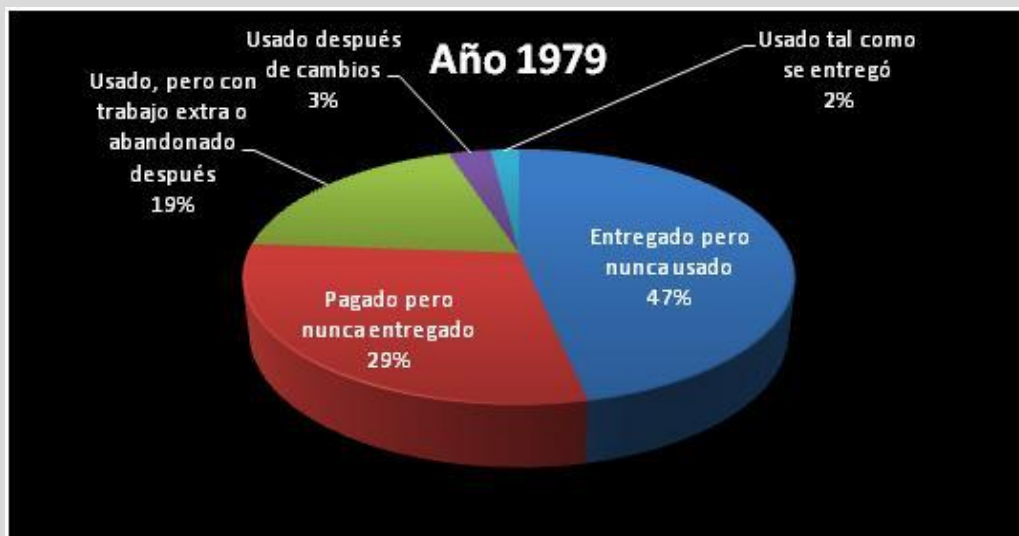


Figura 3- Estadísticas de efectividad de Proyectos IT (Standish Group 1979)

Puede observarse en la Figura 4 que los datos no varían.

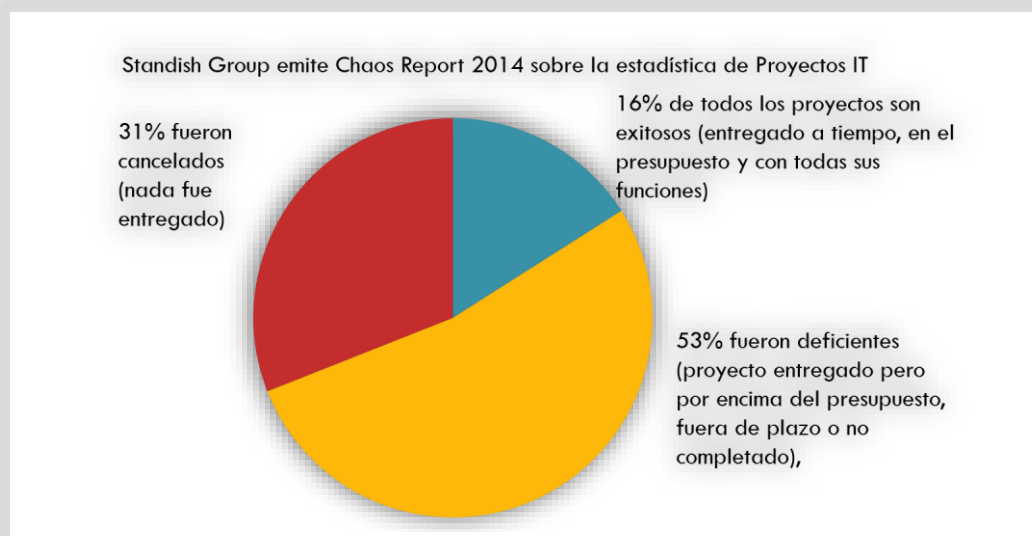


Figura 4 – Estadísticas de efectividad de Proyectos IT (Standish Group 2014)

Cuando se hace referencia a errores en los requisitos, habitualmente se desliza la idea de que el profesional de software o el proceso utilizado han sido el origen de los mismos. Esta suposición probablemente sea parcialmente falsa ya que muchos de los errores, discrepancias y omisiones que aparecen en los requisitos y se propagan hasta el Sistema de Software final, ya existían en la fuente de información. Pero no es una postura adecuada dejarle la responsabilidad a la fuente de información, sino que en realidad ésta

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 2 - Análisis de Sistemas e Ingeniería de Requisitos

está compartida con los procesos de requisitos que tienen la obligación de ayudar al ingeniero/a de requisitos a detectar la mayor cantidad posible de problemas existentes en el macrosistema, principalmente las discrepancias y omisiones.

“Corrige los errores de la especificación cuanto antes. Corregirlos después, costará: un 500% más en la etapa de diseño, un 1000% en la etapa de codificación, un 2000% en la etapa de testing unitario y un 20000% más en la entrega”.

Barry Boehm

Muchos de los fallos en los proyectos de software, vienen por los métodos utilizados en el AS, por no haber capturado todos los requisitos en tiempo y forma. El Análisis de Sistemas expresa los requisitos en términos del producto “el sistema hará un reporte mensual con las ventas que superen las 1000 unidades”. En estos términos la construcción de los sistemas de software se direcciona más al producto que a las necesidades de los clientes, o dicho de otra manera es probable que algunas decisiones de diseño no estén totalmente alineadas a las necesidades de los clientes y sea este último quien deba adaptarse a alguna funcionalidad del software. Los requisitos del software en el Análisis de Sistemas describen el producto software en función a los datos, las funciones, los procesos del software, etc. [Bell76]. A diferencia del AS, la Ingeniería de Requisitos describe los servicios que tendrá el nuevo Sistema de Software desde el punto de vista del cliente.

Para profundizar en ambos paradigmas, a continuación, se presentan algunas definiciones de AS:

[Senn91] *“El Análisis de Sistemas es el proceso de clasificación e interpretación de hechos, diagnóstico de problemas y empleo de la información para recomendar mejoras al sistema. Este es el trabajo del analista de sistemas.”*

[Kendal97] *“El análisis y diseño de sistemas que los analistas de sistemas llevan a cabo busca comprender qué necesitan los humanos para analizar la entrada o el flujo de datos de manera sistemática, procesar o transformar los datos, almacenarlos y producir información en el contexto de una organización específica. Mediante un análisis detallado, los analistas buscan identificar y resolver los problemas correctos. Además, el análisis y diseño de sistemas se utiliza para analizar, diseñar e implementar las mejoras en el apoyo para los usuarios y las funciones de negocios que se puedan llevar a cabo mediante el uso de sistemas de información computarizados.”*

Dictionary Merriam Webster¹ - *“Es el proceso de estudiar un procedimiento o negocio con el fin de identificar sus objetivos y propósitos y crear sistemas y procedimientos que los logren de una manera eficiente.”*

Sólo a modo de observación se muestra una debilidad en la claridad de estas definiciones y está referida a la palabra “sistema” que es utilizada con dos significados diferentes. En la primera y segunda definición, se utiliza para describir el proceso del negocio, mientras que, en la última definición, se utiliza como sinónimo de Sistema de Software.

En resumen, el cliente espera frases donde vea reflejada su problemática y sus particularidades, algo que no se puede lograr sin una ERS.

Ingeniería de Requisitos

La Ingeniería de Requisitos se estableció a mediados de los 70, donde se comenzó a investigar como una práctica individual [Loucopoulos95]. Mucho se ha avanzado en cuanto a los métodos, técnicas y herramientas, pero aún queda mucho por hacer.

A diferencia de las secciones anteriores, esta sección se desarrolla con mayor profundidad en el siguiente Capítulo, por tal motivo, sólo se describe lo que es importante para comprender lo que sigue.

La Ingeniería de Requisitos postula que la importancia de los inconvenientes mencionados para el AS, referidos a no asegurar una clara comprensión de las necesidades de los clientes-usuarios, es notablemente superior a las ventajas enunciadas y que lo que se debe hacer es facilitar la comprensión por parte de los clientes y usuarios de las características del software que van a recibir. Que este postulado sea efectivamente correcto no ha sido completa y extensivamente probado. Tampoco es el objeto de este

trabajo aclarar este punto en gran profundidad. Sí, se pretende clarificar las diferencias entre el Análisis de Sistemas y la Ingeniería de Requisitos y todas sus implicancias.

La Ingeniería de Requisitos comienza cuando el cliente detecta una necesidad que potencialmente se puede satisfacer con un sistema informático. Desde ese momento comienzan a gestarse los requerimientos que a través de un proceso de refinamiento permiten dilucidar la mejor propuesta para el negocio, o sea los requisitos del software. Cabe destacar que, a diferencia del AS, estos requisitos no incluyen aspectos de diseño. Esto le permite a la Ingeniería de Requisitos identificar los servicios que responden a las necesidades reales del cliente desde la mirada del negocio. Por lo tanto, la Ingeniería de Requisitos es responsable de generar una ERS junto al cliente-usuario con el objetivo de construir la solución óptima para la organización que solicitó el Sistema de Software.

A continuación, se presentan algunas definiciones de Ingeniería de Requisitos:

[Leite89] *“Es el proceso mediante el cual se intercambian diferentes puntos de vista para recopilar y modelar lo que el sistema va a realizar. Este proceso utiliza una combinación de métodos, herramientas y actores, cuyo producto es un modelo del cual se genera un documento de requisitos.”*

[Zave97] *“Rama de la Ingeniería del Software que trata con el establecimiento de los objetivos, funciones y restricciones de los sistemas de software. Asimismo, se ocupa de la relación entre estos factores con el objeto de establecer especificaciones precisas.”*

[Loucopoulos95] *“La Ingeniería de Requisitos puede ser definida como un proceso sistemático de desarrollo de los requisitos a través de un proceso cooperativo-interactivo de análisis del problema, documentando los resultados observados en una variedad de formatos de representación, y chequeando la comprensión obtenida. Las actividades envueltas en la Ingeniería de Requisitos ayudan a comprender las exactas necesidades de los usuarios de un Sistema de Software y trasladar esas necesidades en sentencias precisas y sin ambigüedad las que luego serán utilizadas en el desarrollo de software.”*

[Sommerville11] *“La Ingeniería de Requisitos es un proceso que comprende todas las actividades requeridas para crear y mantener un documento de requisitos del sistema. Existen cuatro actividades genéricas de alto nivel en este proceso. Estas son el estudio de factibilidad del sistema, la obtención y análisis de los requisitos, la especificación y la documentación, y finalmente la validación.”*

La presente tesis adhiere a la definición de Leite.

Diferencia entre el Análisis de Sistemas y la Ingeniería de Requisitos

Puede observarse que las definiciones de Análisis de Sistemas y de Ingeniería de Requisitos prácticamente coinciden en sus objetivos, sin embargo, como ya se describió, es en la forma en que se concretan las mismas donde reside la gran diferencia. Efectivamente, establecer con mayor precisión esta diferencia requiere abandonar el nivel de objetivos para avanzar sobre las materializaciones que se hicieron de los mismos.

El Análisis de Sistemas produce una descripción del Sistema de Software en términos técnicos esencialmente computacionales, mientras que la Ingeniería de Requisitos lo hace en términos de los servicios perceptibles por los clientes y usuarios. En la Tabla 4 se describen estas características que marcan y definen el comportamiento de cada uno, ya que para la discusión acerca de las ventajas relativas del Análisis de Sistemas sobre la Ingeniería de Requisitos o viceversa, se debe ponderar la importancia relativa de la calidad de la definición de los servicios y la facilidad para el diseño del sistema.

	Calidad de la definición de los servicios del software	Facilidad para el diseño
AS	Pobre	Buena
IR	Buena	Pobre

Tabla 4 – Ponderación de la distancia desde la IR y el AS al Diseño

Como se mencionó, el resultado producido por el Análisis de Sistemas dificulta la percepción de los clientes y usuarios del grado de satisfacción de sus expectativas, pero facilita la tarea de diseñar el sistema. Por su parte el resultado producido por la Ingeniería de Requisitos facilita la percepción de los clientes y usuarios, por lo que contribuye a una mejor definición de los servicios que tendrá el software, pero paga el precio de dificultar el diseño posterior del sistema.

Puede observarse en la Figura 5 que la salida esperada del Análisis de Sistemas se constituye en un conjunto de esquemas, gráficos, diagramas que abstraen el proceso del negocio en conceptos computacionales muy distantes al usuario, lo que dificulta su comprensión. Mientras que la Ingeniería de Requisitos propone modelos lo más cercano al contexto en estudio, donde el aprendizaje para comprender la propuesta se reduzca a la mínima expresión. Este es el motivo central por el cual, aproximadamente el 70% de los modelos utilizados, representan toda la información en lenguaje natural.

Esta cercanía al punto de vista del software le permite al Análisis de Sistemas detectar rápidamente la viabilidad de los pedidos de los clientes y usuarios. Algo que a la Ingeniería de Requisitos le resulta más difícil de percibir. Por lo tanto, cuando un cliente-usuario solicita un servicio o una restricción, el ingeniero/a de requisitos debe estar atento a que ese pedido pueda ser incluido, ya que de no ser posible o genere costos excesivos, se convertirá en un problema al avanzar con la construcción del software. Esto obliga al ingeniero/a de requisitos a estar actualizado para reconocer la viabilidad de los servicios posibles al momento de estudiar cada problema en particular.

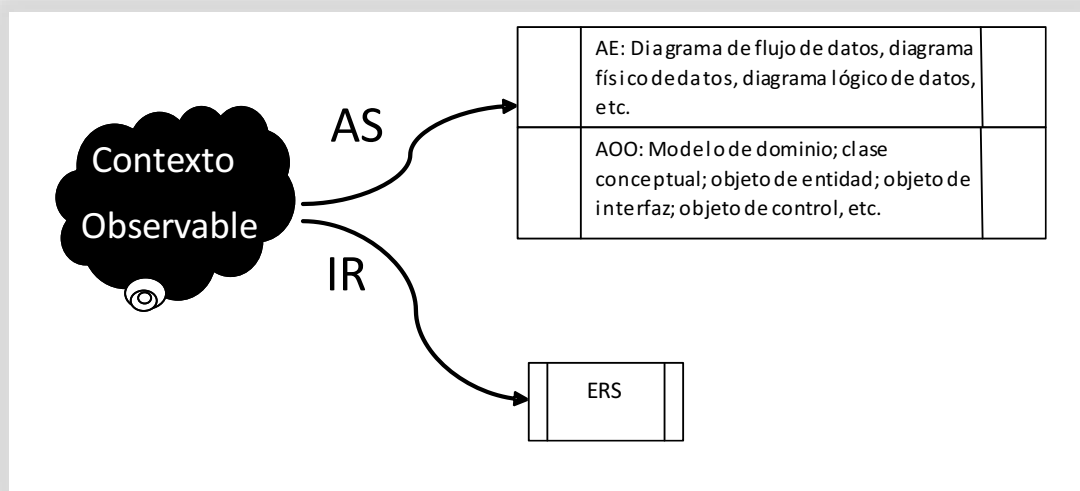


Figura 5 – Artefactos generados por el AS y la IR

El Análisis de Sistemas procura describir la realidad observable utilizando lo más temprano posible técnicas y modelo propios del procesamiento de datos, procurando planificar el proceso del negocio futuro en esos términos, mientras que la Ingeniería de Requisitos procura describir la realidad observable utilizando el lenguaje y los puntos de vista de los clientes y usuarios; planificando el proceso de negocio futuro también en términos del punto de vista de los clientes y usuarios. En la Figura 6 se puede observar que el Análisis de Sistemas genera diagramas que rápidamente se trasladan al diseño, mientras que la Ingeniería de Requisitos está más lejos y necesita un esfuerzo adicional para lograrlo.

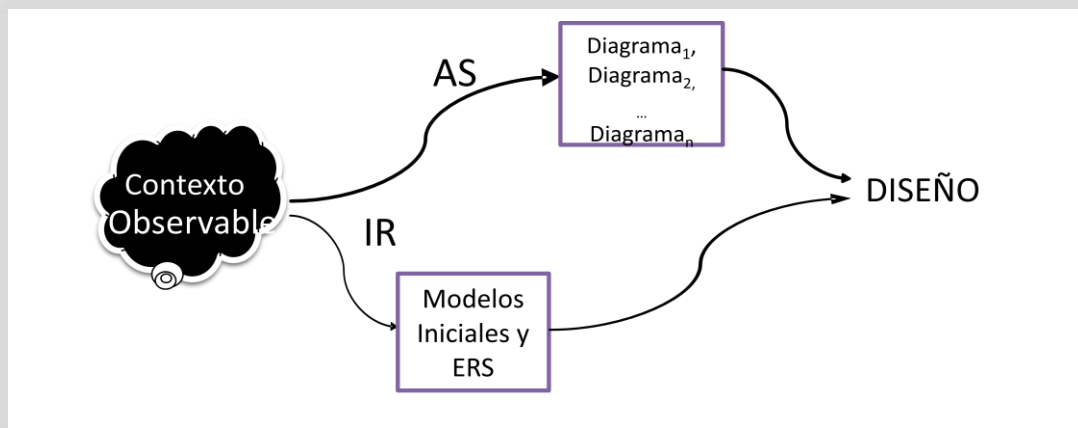


Figura 6 – Distancia al diseño desde el AS y la IR

En la Figura 6 se identifican dos distancias cognitivas. Cabe aclarar que esta distancia incluye sólo la comprensión y el esfuerzo tanto del cliente como del desarrollador. La primera distancia corresponde al camino entre el mundo observable y la obtención de los requisitos del software. La segunda, es el salto hacia el diseño.

Cabe destacar que tanto la Figura 5, como la Figura 6, se han pensado en un entorno de la aplicación de un Modelo de Proceso en Cascada y que deben ser reinterpretadas dependiendo del modelo de proceso que se trate.

En resumen, el precio que se paga por ir por el camino del Análisis de Sistemas es más caro porque no asegura los resultados y eso repercute en la calidad y viabilidad del producto. Es por eso que la comunidad de informática ha evolucionado del Análisis de Sistemas a la Ingeniería de Requisitos procurando un acercamiento al contexto y a las necesidades reales de los clientes y usuarios.

Capítulo 3

Contexto de la Ingeniería de Requisitos

Resumen

La Ingeniería de Requisitos se desarrolla en un contexto particular, el cual determina muchas de las decisiones que se deben tomar. La principal dificultad es que a dicho contexto se lo conoce a medida que se avanza en el Proceso de Requisitos. En el presente Capítulo se abordan algunas de las características más relevantes que deben ser contempladas y comprendidas para construir los requisitos del software de la mayor calidad posible.

Contexto de la Ingeniería de Requisitos

El contexto de la Ingeniería de Requisitos influye sobre la forma de definir los requisitos del software. Ya sea que se obtenga muy rápidamente un conjunto de requisitos relativamente incompleto, posiblemente dudosos y con eventuales contradicciones o que se obtengan requisitos de muy alta calidad, en un plazo apropiado. Por lo tanto, se puede definir al contexto como: todos los factores que afectan la forma de trabajar del ingeniero/a de requisitos. Por tal motivo, el contexto de la Ingeniería de Requisitos, es un concepto ambiguo ya que no se sabe si se refiere a todo aquello que rodea a las actividades del Proceso de Requisitos o si se refiere a todo aquello que rodea al Sistema de Software a desarrollar. Por otro lado, la mayoría de los enfoques de la Ingeniería de Requisitos se desarrollan en un contexto implícito, donde no se toman en cuenta aspectos que condicionan la obtención de los requisitos del software, como ser:

- Costo de oportunidad del Sistema de Software;
- Poco interés de los usuarios en el nuevo Sistema de Software;
- Disponibilidad limitada de los clientes-usuarios;
- Imposibilidad para obtener información del macrosistema;
- Falta de políticas organizacionales claras;
- Acciones erráticas de los usuarios (por ejemplo, resolución de problemas que entran en conflicto con reglas del negocio o normativas);
- Ausencia de procedimientos bien definidos;
- Presiones de la competencia.

Se debe tener en cuenta que la Ingeniería de Requisitos cambia cuando algunos de estos aspectos se hacen presentes, ya que el contexto determina qué Proceso de Requisitos se debe utilizar y define qué actividades de ese proceso se van a realizar y cuáles no.

Esta tesis estudia y hace propuestas para procesos de requisitos que no deben lidiar con estos aspectos. Posiblemente parte de las conclusiones sean extensibles o adaptables a algunas de las situaciones descriptas, pero eso no ha sido analizado en este trabajo.

Finalmente, es necesario aclarar que el contexto se irá refinando durante toda la Ingeniería de Requisitos, aunque prematuramente el Objetivo General del Sistema ya lo delimita.

Relaciones de poder en el Proceso de Requisitos

En los procesos de requisitos existen dos grupos que trabajan cooperativamente, pero diferenciados: el grupo cliente y el grupo proveedor. En el primer grupo se encuentran los involucrados mientras que en el segundo grupo se encuentran los ingeniero/as de requisitos. Estos grupos interactúan fuertemente durante todo el proceso y es notable la forma en que se ve afectado el mismo por la relación de poder entre ambos grupos. En otras palabras, se producen resultados muy diferentes si se está en presencia de un grupo cliente dominante, un grupo proveedor dominante o una relación de poder equilibrada.

La situación de *grupo cliente dominante* ocurre cuando en la organización existe un área de desarrollo de software que depende jerárquicamente del área adonde pertenecen los clientes (ver primer cuadro de la Figura 7). Por ejemplo, un departamento de desarrollo que es parte de una gerencia de comercialización o de una gerencia contable o administrativa. También ocurren situaciones de *grupo cliente dominante* cuando existen numerosos proveedores potenciales y la organización cliente tiene muchas opciones para seleccionar uno de ellos (situación de sobre oferta) (ver primer cuadro de la Figura 8).

Situaciones de *grupo proveedor dominante* se producen cuando el nivel organizacional del grupo de desarrollo es más alto en la jerarquía que el del grupo cliente (ver tercer cuadro de la Figura 7). En el caso de proveedores externos, el caso de *grupo proveedor dominante* se produce cuando existe sobredemanda de desarrollos (ver tercer cuadro de la Figura 8).

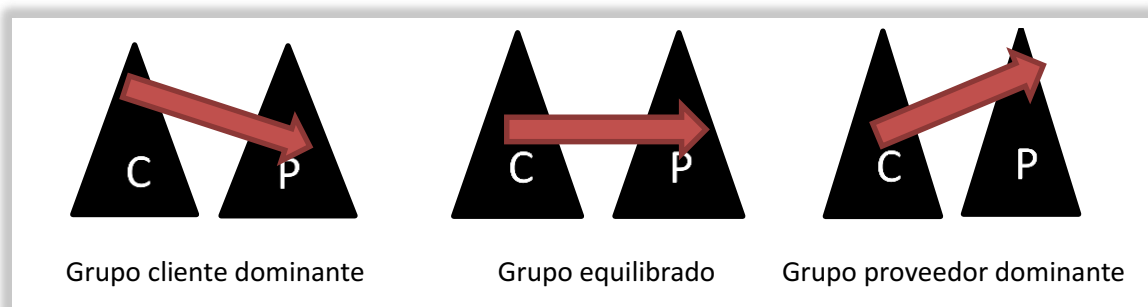


Figura 7 – Relación “Cliente – Proveedor”

Finalmente, una situación equilibrada tiene lugar cuando el grupo clientes y el grupo de desarrolladores tienen una ubicación jerárquica similar en el organigrama de la organización (ver segundo cuadro de la Figura 7). En el caso de proveedores externos, esta

situación ocurre cuando la oferta y la demanda están equilibradas (ver segundo cuadro de la Figura 8).

Cuando se está en presencia de *grupo clientes dominantes*, resulta difícil o prácticamente inviable realizar una Ingeniería de Requisitos, por lo tanto, es mejor utilizar modelos de procesos aptos para desenvolverse en contextos de alta volatilidad de los requisitos, como ser procesos de prototipado [Floyd84] o métodos ágiles [Cockburn02] [Highsmith02] [Strode05] [Visconti07].

Cuando se trata de una situación con un *grupo proveedor dominante*, se debe poner especial énfasis en atender apropiadamente las necesidades de los clientes-usuarios de manera de evitar que la mayoría de las decisiones sean tomadas unilateralmente por el ingeniero/a de requisitos.



Figura 8 – Relación “cliente-proveedor externo”

Universo de Discurso

El contexto específico para el desarrollo del nuevo Sistema de Software se denomina Universo de Discurso (UdeD). Es importante comprender “*donde*” las actividades de la Ingeniería de Requisitos tendrán lugar y de qué manera se deben situar, atendiendo, no sólo al proceso de construcción del software sino también al ambiente en donde el software se ejecutará. En el UdeD conviven una gran cantidad de variables. Parte de la complejidad de la Ingeniería de Requisitos consiste en esta convivencia que se debe contemplar a la hora de especificar los requisitos del software. Estas variables son:

- El proceso del negocio.
- Las necesidades/deseos de los clientes-usuarios.

- La planificación del proceso de negocio futuro.
- El vocabulario del macrosistema.
- Las fuentes de información.
- Los involucrados.

Para algunos autores como Bjørner [Bjørner06] el UdeD está compuesto por el grupo de personas interesadas en comunicarse y en los artefactos de requisitos y diseño. Determina un único UdeD para todo el proceso de desarrollo. Mientras que otros autores definen UdeD como el espacio organizacional donde reside toda la información necesaria para construir el nuevo Sistema de Software [Leite94] [Jackson94]. La presente tesis adhiere a la definición de Leite y describe el UdeD de la siguiente manera:

“Es todo el contexto en el cual el software se desarrolla e incluye todas las fuentes de información. Es la realidad acotada por el conjunto de objetivos establecidos por quienes demandan una solución de software”

Aunque el macrosistema no esté bien definido, siempre se puede y se deben establecer los límites de la actuación del ingeniero/a de requisitos. Cuanto mejor este delineado el UdeD, mayores serán las oportunidades de construir un software completo y correcto. Pero cabe destacar que este UdeD se va precisando y mejorando sus límites durante todo el Proceso de Requisitos. Esto implica que algunas fronteras pueden estar algo confusas en las primeras etapas del proceso, pero deben quedar bien definidas al finalizar dicho proceso.

El UdeD no es estático. Va cambiando a medida que el tiempo transcurre. Estos cambios alteran el proceso del negocio y, por lo tanto, los requisitos del software. Estas transformaciones que sufren los requisitos, producidos en el tiempo, se denomina **evolución de los requisitos** y debe ser reflejado en los modelos construidos y gestionado apropiadamente.

Si bien el UdeD es el contexto particular para el desarrollo del nuevo Sistema de Software, existe otro contexto más general de la organización, el cual puede influir en las decisiones que se deben tomar durante la construcción del nuevo software. La Figura 9 muestra ambos contextos y un área de intersección que representa los aspectos del contexto general que influyen sobre el UdeD.

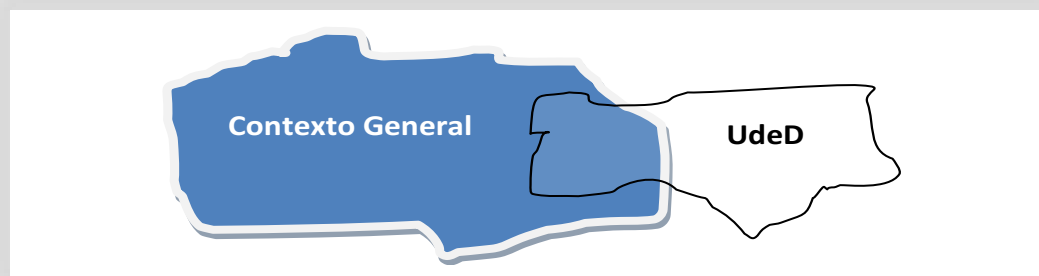
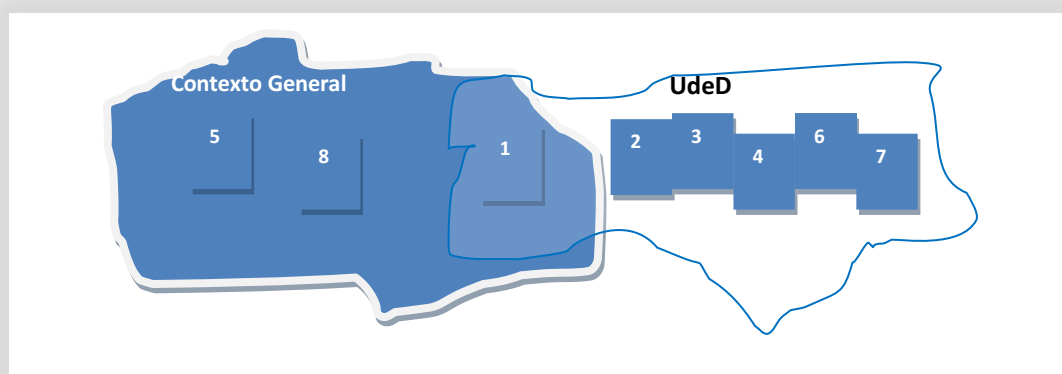


Figura 9 – Contexto general y UdeD

En la Figura 10 se puede observar que existen variables propias del contexto general, otras de la intersección y las propias del UdeD.



1. Costo de oportunidad del Sistema de Software;
2. Poco interés de los usuarios en el nuevo Sistema de Software;
3. Disponibilidad limitada de los clientes-usuarios;
4. Imposibilidad para obtener información del macrosistema;
5. Falta de políticas organizacionales claras;
6. Acciones erráticas de los usuarios (ej., resolución de problemas que entran en conflicto con reglas del negocio o normativas);
7. Ausencia de procedimientos bien definidos;
8. Presiones de la competencia.

Figura 10 – Variables del contexto general y del UdeD

Como ya se mencionó, la presente tesis no analiza todos los ítems de la Figura 10, en diferentes Capítulos se analizan las variables numeradas como 2, 3, 4, 6 y 7 en el contexto del Proceso de Requisitos basado en Escenarios.

En la presente tesis, los términos “contexto” y “universo de discurso” se utilizarán en adelante como sinónimos.

Evolución del Universo de Discurso

Antes de describir la *evolución del UdeD* es necesario aclarar cuál es la acepción de la palabra evolución utilizada en el presente documento. Existe una controversia en la bibliografía, acerca del uso de este término. Para algunos autores el concepto está ceñido

a las transformaciones que sufren los modelos durante todo el proceso de construcción del software, desde la IR, pasando por el diseño, el código y siguiendo con otras actividades hasta llegar a la puesta en servicio. Este es el caso de Brietman et. al [Brietman05] donde se denomina evolución a las transformaciones de los Escenarios y otros documentos durante todo el ciclo de vida del desarrollo de software. Otros, se refieren a la evolución como los cambios que sufre el macrosistema producto de planificar la inserción de un nuevo Sistema de Software. La presente tesis adhiere a la última definición.

Ya se mencionó que el UdeD evoluciona. Esta evolución comienza con la mera decisión de incorporar un nuevo Sistema de Software, la cual habilita una nueva línea de pensamiento. Algo parecido sucede con el propio Proceso de Requisitos, el cual facilita la generación de nuevas ideas al tener que describir aspectos no observables o alterar, en diferente grado, el proceso del negocio existente. Otro aspecto que hace evolucionar el UdeD es que las organizaciones cambian; a veces por factores internos (tecnológicos, políticas organizacionales, sociales, eventualmente culturales) y en otros casos por factores externos (legislación, competencia, etc.). Los cambios del UdeD, en algunos casos, son concurrentes y asincrónicos, esto implica que cuando uno comienza a desarrollarse el otro puede estar concluyendo. Cuando un proceso del negocio se ve afectado por diferentes cambios, se genera una inminente necesidad de evolucionar los requisitos de dicho software [Brietman05].

Puede observarse en la Figura 11 que existe un UdeD actual y otro UdeD futuro. Esta división en “actual” y en “futuro” sólo existe a efectos de mejorar la conceptualización del tema, ya que el UdeD es único. El *UdeD actual* describe el proceso del negocio como existe al momento de comenzar con el desarrollo del software y el *UdeD futuro* modela la planificación del proceso del negocio con el Sistema de Software incluido.

El gráfico de la Figura 11 muestra la evolución del contexto y se lo dividió en dos secciones, A y B, con el objetivo de diferenciar la evolución real del UdeD (parte A) y la “evolución” que se genera explícitamente dentro de la Ingeniería de Requisitos, lo que se ha denominado “evolución aparente” (parte B). La evolución real se efectiviza un tiempo después de que el Sistema de Software se encuentra en ejecución, a diferencia de la evolución aparente que se genera con el objetivo de obtener una ERS. Esta evolución es conceptual ya que dichos cambios sólo se pueden observar en los artefactos producidos

en el mismo Proceso de Requisitos y de los cuales se desconoce que parte será aceptada en el UdeD y que parte se rechazará. Esta evolución se lleva a cabo en un lapso relativamente corto de tiempo.

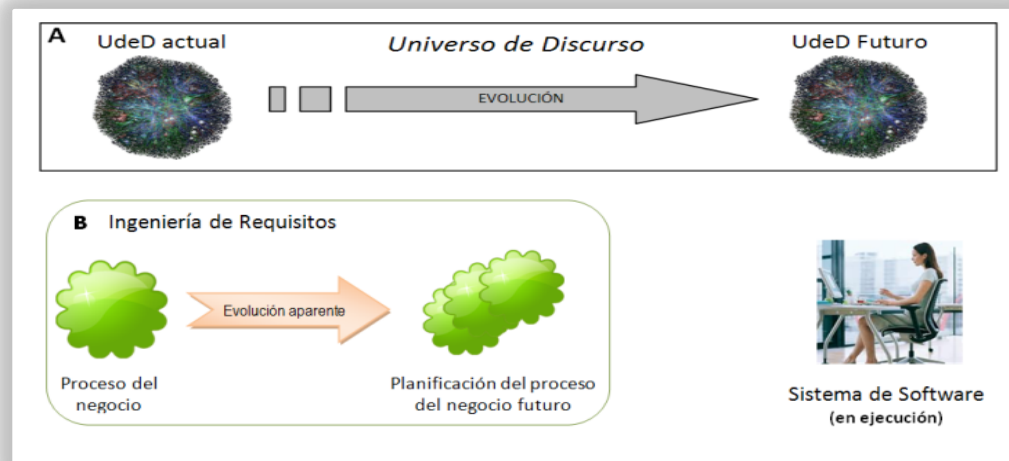


Figura 11 – Relación entre la evolución del UdeD y la IR

La evolución del contexto genera la necesidad de construir ERS dinámicas que deben ser registradas correctamente para realizar el seguimiento de cada requisito desde cualquier punto del proceso de construcción del software hacia sus orígenes (backward traceability o rastreabilidad hacia atrás) o desde sus orígenes hacia el código (forward traceability o rastreabilidad hacia adelante) [Wieringa95].

Límites del Universo de Discurso

Es sustancial recalcar que el UdeD no determina los límites del sistema. Es necesario ahondar en el UdeD, comprenderlo casi en su totalidad y conocer las necesidades de los clientes-usuarios para definirlo. Parfraseando a Jackson [Jackson95], si bien la definición de los límites es lo primero que se debe hacer, es casualmente ese el momento en que menos se sabe al respecto. No sólo se sabe menos, sino que no se tiene acceso a esa información, ya que la misma se irá completando y refinando durante toda la Ingeniería de Requisitos. Experiencias del mismo estudio replicado por diferentes personas, han demostrado que los límites que establecen esas distintas personas suelen ser diferentes. Esta diferencia puede representar solamente un innecesario exceso de trabajo, pero también puede indicar haber ignorado aspectos importantes del problema.

Aunque el alcance del sistema no esté correctamente definido, siempre se puede y se deben establecer los límites de la actuación del ingeniero/a de requisitos. Cuanto mejor este delineado el UdeD, mayores serán las oportunidades de construir un Sistema de Software al menor costo y en los tiempos apropiados.

Conocimiento del Contexto

La RAE define *conocimiento* como la “acción y efecto de conocer” donde conocer es “entender en un asunto con facultad legítima para ello”. Este entendimiento actúa como un marco referencial que evita cometer errores en desmedro de la organización. O sea, conocer las reglas del negocio, los procedimientos formalizados, los estándares utilizados, los manuales, entre otros, obliga a mantenerse dentro los límites organizacionales. Esta información tiene el punto de vista sobre *cómo* se debe realizar el proceso del negocio para que esté alineado a la organización (punto de vista “Deber ser”). Para obtener esta información es necesario el *compromiso* del cliente, el cual se debe efectivizar en un Proceso de Requisitos altamente *cooperativo*, indispensable para realizar una Ingeniería de Requisitos exitosa.

Generalmente, las personas son las poseedoras del conocimiento más importante para un Proceso de Requisitos. Esto se debe a que pueden tener varias perspectivas simultáneas, como ser el marco referencial de la organización, la operacionalización de los objetivos, la cultura organizacional, etc. Pero este conocimiento no es tan fácil de obtener ya que puede estar presente de forma explícita o implícita.

El *conocimiento explícito* es aquel que se encuentra en un nivel “consciente”, es fácil de percibir ya que se utiliza para realizar las tareas. Se obtiene a través de la observación de las acciones de las personas, de las descripciones en la literatura del macrosistema, en los sistemas informáticos o en las entrevistas (conversaciones) con los usuarios o brainstorming. No todo el conocimiento explícito es visible [Lapeña19]. Existe una parte tácita de este conocimiento, que es aquel que no se expresa. En general es aquel conocimiento rutinario que se da por sobreentendido o por conocido.

El *conocimiento implícito* es el tipo de conocimiento que permanece en un nivel “inconsciente”, se encuentra desarticulado y se utiliza de una manera mecánica sin que la fuente de información se dé cuenta de su contenido. Este conocimiento suele ser difícil de detectar y se refiere a aspectos culturales de la organización, aspectos culturales,

intereses propios de los usuarios, interpretaciones sobre la literatura del macrosistema, etc.

Esta diversidad de conocimiento también puede encontrarse en el ingeniero/a de requisitos, por ejemplo, cuando tiene experiencia en dominios similares.

Los procesos de requisitos deben ser un canal de estimulación para visibilizar todo el conocimiento existente. Las técnicas de elicitación [Goguen93] permiten obtener la información del macrosistema mientras que los modelos, los métodos y las herramientas la representan para que esté disponible.

Involucrados (Stakeholders)

Involucrados es un término muy utilizado en la Ingeniería de Software (IS) [Sharp99] [Gonzalez10], pero su origen es del área de la Administración. Por su uso tan diverso, existen muchos homónimos en diferentes contextos y en cada caso se incluyen grupos particulares de personas, siendo en algunos casos muy sutil la diferencia. Por tal motivo, se presenta una breve historia del uso del término.

El Instituto de Investigación de Stanford introdujo el término stakeholder por primera vez en la literatura en 1963 [Freeman84] para referirse a un reducido grupo de personas estrechamente comprometidas con el funcionamiento de la empresa, básicamente: accionistas, empleados, clientes y aquellos con capacidades técnicas esenciales para las operaciones de la compañía. Fue así como, en un primer momento, los stakeholders se identificaron con “aquellos grupos sin cuya contribución la corporación dejaría de existir”. Estos stakeholders podían no participar en la propiedad de la empresa, pero eran fundamentales para la planificación empresarial y el éxito de la firma. Siguiendo esta línea, Freeman define stakeholder como “todo grupo o individuo que puede afectar o verse afectado por la consecución de los objetivos de la organización”.

En el contexto de las actividades apoyadas por el Banco Mundial, los stakeholders son “los afectados por el resultado, negativo o positivamente o que puedan afectar el resultado de una intervención propuesta” [ESD96] “Los stakeholders pueden incluir: Prestatarios (funcionarios electos, personal de línea de agencia, los funcionarios del gobierno local, y así sucesivamente), los grupos directamente afectados (incluyendo pobres y desfavorecidos), los grupos afectados indirectamente (por ejemplo, las ONG y

organizaciones del sector privado), y la administración del Banco Mundial, el personal y los accionistas” [ESD98].

El Business Dictionary define stakeholders como: “una persona, grupo u organización que tenga interés directo o indirecto en una organización, ya que puede afectar o verse afectados por las acciones de la organización, los objetivos y las políticas. Las partes interesadas en una organización empresarial incluyen los acreedores, clientes, directores, empleados, gobierno (y sus agencias), los propietarios (accionistas), proveedores, sindicatos y la comunidad de la cual la empresa obtiene sus recursos.”³

Sternberg describe la responsabilidad de los interesados como una participación consiente y define a los stakeholders como “aquellos en quienes la organización tiene un interés” [Sternberg99]. Alonso por otro lado, afirma que es los stakeholders son “todos aquellos que tienen un interés en la organización” [Alonso08].

En el campo de la informática el consenso se encuentra en cuanto a que los stakeholders son aquellas personas de la organización que tienen una relación directa o indirecta con el nuevo Sistema de Software, pero la controversia se mantiene en cuanto a si incluyen a los desarrolladores. Se presentan algunas definiciones:

[Leite94] *“Las personas involucradas en el proceso de desarrollo son principalmente: usuarios, clientes, ingenieros de software, expertos del dominio, las cuales son denominadas en la literatura con la palabra ‘stakeholders’”*

[Pouloudi97] *“son los participantes en el desarrollo de procesos junto con cualquier otro individuo, grupos u organizaciones cuyas acciones pueden influir o ser influido por el desarrollo y uso del sistema ya sea directa o indirectamente”.*

[Sommerville11] *“todos aquellos que se benefician en una forma directa o indirecta del sistema que está en desarrollo”.* Entre los stakeholders se encuentran los usuarios finales que interactúan con el sistema y todos aquellos en la organización que se pueden ver afectados por su instalación. Otros stakeholders del sistema pueden ser los ingenieros que desarrollan o dan mantenimiento a otros sistemas relacionados, los gerentes del negocio, los expertos en *el dominio del sistema* y los representantes de los trabajadores.

³ <http://www.businessdictionary.com/definition/stakeholder.html>

La presente tesis adhiere a la definición de Sommerville, dejando expresamente aclarado que se excluye del grupo de los stakeholders a los ingenieros/as de software.

Entre los stakeholders que sobresalen se encuentra el cliente y el usuario. Cabe aclarar que a menudo no son la misma persona [IEEE830] pero pueden ser roles intercambiables. El *usuario* es quien se ocupa de los detalles operativos y se expresa en términos muy concretos, mientras que el *cliente*, es quien demanda el nuevo Sistema de Software, se ocupa del Objetivo General del Sistema y se expresa en un nivel de abstracción más alto. Estos roles no son estáticos y pueden ser intercambiados cuando el cliente menciona detalles del proceso del negocio o cuando el usuario justifica algún aspecto referido con el nuevo Sistema de Software. Con el objetivo de evidenciar este intercambio de roles, que a menudo es transparente para el ingeniero/a de requisitos, en la presente tesis se utiliza el término “*cliente-usuario*”.

Fuentes de Información

Se denomina Fuente de Información (FI) a todo aquello que contiene, de diferente manera, la base de conocimiento de la organización y en particular del macrosistema específico que se desea estudiar. Las FI son parte del UdeD y se concentran en dos grandes grupos: personas y recursos.

La **FI personas** son aquellas que tienen alguna influencia directa o indirecta sobre los requisitos del software. Como se mencionó en la sección anterior, estas personas se denominan aquí “involucrados”.

En el contexto de la Ingeniería de Requisitos, como se mencionó, las FI más importante son las personas [Antonelli02]. Este hecho se debe a que son poseedoras del mayor conocimiento del negocio y esto enriquece significativamente el Proceso de Requisitos, pero a su vez es la FI más compleja [Cross01]. Estas características se deben a:

- La posibilidad de generar nuevo conocimiento.
- La aportación de experiencias previas en contexto similares y diferentes.
- El enriquecimiento del macrosistema con aspectos socio-culturales propios.

Algunos de estos ítems se generan y transforman a través de los vínculos establecidos entre las personas generando en muchos casos, nuevo conocimiento. Cross [Cross01] marca que en general, cuando se piensa en información, suele ponerse poca atención a las

relaciones interpersonales, aun siendo que estas relaciones pueden ser críticas para obtener información y para resolver problemas y aprender cómo hacer su trabajo. Cada stakeholder ve el proceso del negocio desde una perspectiva propia o desde el rol que desempeña en la organización. Un mismo stakeholder puede cambiar de punto de vista dependiendo del momento. A su vez, distintos stakeholders pueden ver un proceso o tarea desde el mismo punto de vista o el contrario. Puede observarse en la Figura 12 que la FI personas pueden tomar diferentes perspectivas o tener diferentes puntos de vista del contexto, dependiendo de su rol en la organización, de sus interés, deseos, expectativas, miedos, etc.



Figura 12 - Puntos de vista y perspectivas de los involucrados

Otra variante para analizar los stakeholders se basa en considerar su poder dentro de la organización y el dinamismo de cada persona dentro de ella. Estas variables pueden ser utilizadas como indicadores para seleccionar los stakeholders del Proceso de Requisitos. Como se ve en la Tabla 5, existen cuatro grupos. Los del grupo A y B suelen ser fáciles de tratar; los del grupo C suelen tener pocas expectativas y por lo tanto se pueden manejar; y los del grupo D son los de mayor atención ya que tiene poder y son poco predecibles. La **FI recursos** es todo aquello necesario para que las personas realicen sus tareas. Entre los recursos se encuentran:

- Toda la literatura del macrosistema (organigrama, políticas de la organización, manuales de usuario, manuales de procedimientos, normativas, contratos, formularios, libros, artículos en revistas especializadas, estándares nacionales e internacionales en uso o para aplicar, etc.);

- Sistemas de software propios y de terceros (COTS).
- Legislación vigente que afecta al macrosistema en estudio.

Poder dentro de la Organización	Dinamismo	
	BAJO	ALTO
BAJO	A Pocos problemas	B Poco predecibles pero manejables
ALTO	C Poderoso pero predecibles	D Peligrosos u Oportunistas

Tabla 5 - Matriz de interés [Gardner86]

La importancia de las FI se determina, cuantitativa y cualitativamente, por la información intrínseca que contienen.

Capítulo 4

Ingeniería de Requisitos

Resumen

Siendo la Ingeniería de Requisitos la etapa que proporciona los lineamientos para el resto del proceso de construcción del software y, además, siendo el marco de referencia de la presente tesis, se describe en el presente Capítulo algunos conceptos generales que permiten entre otras cosas, unificar criterios.

Para obtener software de calidad, se ha requerido de un enfoque sistemático, disciplinado y cuantificable para su desarrollo, operación y mantenimiento. A esta disciplina se la ha denominado Ingeniería de Software y se encarga de todo lo referido a la producción de software [Sommerville07], desde el comienzo hasta después de que el mismo se encuentra en uso. A continuación, se presentan un par de definiciones:

[Boehm76] *“es la aplicación práctica del conocimiento científico al diseño y construcción de programas de computadora y a la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software.”*

[Pressman05] *“se trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales.”*

En el marco de la Ingeniería de Software, la obtención de los requisitos del software es una de las actividades más tempranas que se deben realizar. En este momento es donde se definen los servicios que el nuevo Sistema de Software brindará y las restricciones que deberá contemplar, tomando mayor relevancia cuando se está en presencia de un cliente-usuario que desconoce el impacto que producirá la incorporación del Sistema de Software en los procesos de negocio. Esta etapa se denomina Ingeniería de Requisitos y se encarga de la manipulación sistemática de los requisitos del software [Sawyer01] [Wohlrab18] con el objetivo de mejorar la forma en que comprendemos y definimos los sistemas de software. Siendo que en 1993 se realizó la primera conferencia internacional sobre este tema en RE 93 y en 1995 se llevó a cabo la primera reunión del grupo de trabajo IFIP 2.9 (Ingeniería de Requisitos de Software), el término “Ingeniería de Requisitos” ya había sido publicado formalmente en 1977 en IEEE Transactions on Software Engineering. Se presentan a continuación dos definiciones de Ingeniería de Requisitos:

[Loucopoulos95] *“puede ser definida como un proceso sistemático de desarrollo de los requisitos a través de un proceso cooperativo-interactivo de análisis del problema, documentando los resultados observados en una variedad de formatos de representación, y chequeando la comprensión obtenida. Las actividades envueltas en la Ingeniería de Requisitos ayudan a comprender las exactas necesidades de los usuarios de un Sistema de Software y trasladar esas necesidades en sentencias precisas y sin ambigüedad las que luego serán utilizadas en el desarrollo de software.”*

[Sommerville07] *“es un proceso que comprende todas las actividades requeridas para crear y mantener un documento de requisitos del sistema. Existen cuatro actividades genéricas de alto nivel en este proceso. Estas son estudio de factibilidad del sistema, la obtención y análisis de los requisitos, la especificación y documentación, y finalmente la validación.”*

Usualmente, los requisitos del software se agrupan en un documento denominado **Especificación de Requisitos del Software** (ERS o SRS en inglés), principal artefacto producido por la Ingeniería de Requisitos. Según la [IEEE830], una ERS debe tener las siguientes características: debe ser correcta; sin ambigüedad; completa; consistente; ordenada por importancia y/o estabilidad; verificable; modificable; y rastreable. Pero no todos logran obtener una ERS que cumpla con todos estos atributos [Firesmith03], ya que los requisitos de baja calidad han sido a lo largo del tiempo, una de las principales causas de proyectos que han fracasado [Standish94]. Una ERS de baja calidad o pobre, o en el peor de los casos su inexistencia, puede incrementar significativamente el costo de desarrollo. Como se muestra en la Figura 13, cuanto más tarde es la detección de un error, más alto es el **costo de corrección**, con un impacto negativo dentro de la organización que puede afectar la confianza y la cooperación para completar exitosamente el proceso. La incidencia del costo de corrección de los requisitos del software ha sido ampliamente estudiada por autores como [Boehm76] [Bell76] [Daly77] [Basili81] [Davis93] entre otros.

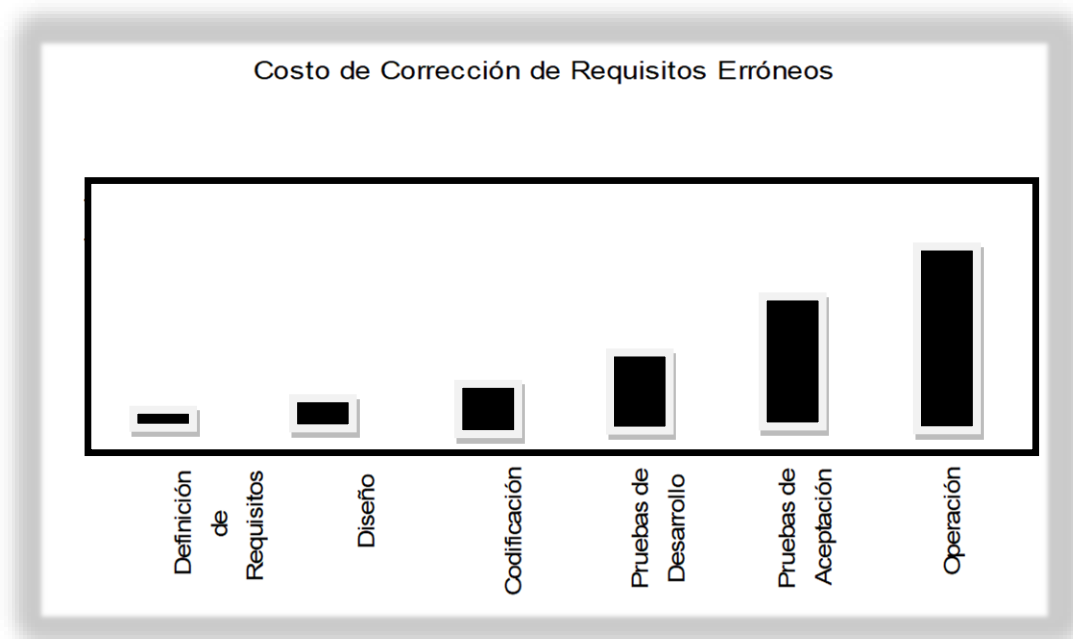


Figura 13 - Costo de corrección de los requisitos [Boehm81]

Habitualmente cuando se hace referencia a errores en los requisitos de software, se desliza en forma tácita la idea que el profesional de software o el proceso de captura han sido la fuente y origen de los mismos. Esta suposición, si bien muy difundida, probablemente sea parcialmente falsa ya que muchos de los errores, discrepancias y omisiones que aparecen en los requisitos y naturalmente se propagan al sistema final, ya existían en la fuente de información. Jackson afirmaba que “*algunos proyectos han fracasado porque sus requisitos son inadecuadamente explorados y definidos*” [Jackson95]. Por otro lado, algunos errores tienen su origen en la falta de compromiso de los clientes-usuarios. En ocasiones los clientes-usuarios dan por aprobada una ERS sin su correspondiente análisis, aludiendo falta de tiempo o no comprender los modelos presentados. La cantidad de errores relacionados con los requisitos del software es muy significativa, y muchos errores se detectan tardíamente cuando podrían haber sido identificados mucho antes en el proceso de construcción del software [Davis93]. En respuesta a la baja calidad de los productos de software con altos costos de corrección y mantenimiento, muchos desarrolladores de software e investigadores han profundizado su estudio en el proceso de definición del software, específicamente en la etapa de Ingeniería de Requisitos. Ellos apuntan a garantizar la producción de software de alta calidad y bajo costo basándose en el establecimiento de un punto de partida de alta confiabilidad.

Requisitos y Requerimiento

Existe una controversia terminológica en el dominio de la Ingeniería de Requisitos relacionada con los términos Requisito y Requerimiento. Estos términos suelen ser utilizados como sinónimos, tanto en la literatura como en la jerga de los desarrolladores, pero no lo son. La etimología de estos términos ya presentaba esta diferencia. Es así que en latín “petitio” tiene su traducción al castellano como “requerimiento” mientras que “requires” como “requisito”. Las definiciones de la Real Academia Española - edición 22⁴ dice lo siguiente:

Requerimiento.

1. m. Acción y efecto de requerir.

Requerir.

1. tr. Intimar, avisar o hacer saber algo con autoridad pública.
2. tr. Reconocer o examinar el estado en que se halla algo.
3. tr. necesitar.
4. tr. Dicho de una persona: Solicitar, pretender, explicar su deseo o pasión amorosa.
5. tr. inducir (|| instigar).

Requisito

- m. Circunstancia o condición necesaria para algo.

Puede observarse que la definición de “requisito” contempla el concepto de obligatoriedad, necesario cuando se debe especificar lo que el software “debe” hacer. El uso incorrecto de la palabra “requerimiento”, puede ser originado en lo que se conoce como “falsos amigos” entre idiomas, que consiste en tener palabras que a pesar de tener un significado diferente se las escribe o pronuncian de una manera similar. Este caso de “amigos falsos” se presenta en la transliteración del término en inglés “requirements” al término en castellano “requerimientos”. Desde la semántica del término “requirements” su traducción se corresponde con la semántica de “requisitos” y para demostrarlo se presentan a continuación las definiciones en inglés⁵:

⁴ <http://www.rae.es>

⁵ <http://dictionary.reference.com>

Requirement

noun

1. that [which](#) is [required](#); a thing demanded or obligatory: One of the requirements of the [job](#) is accuracy.
2. an act or instance of [requiring](#).

Require

verb (used with object)

1. to have need of; need: He requires medical care.
2. to call on authoritatively; order or enjoin to do something: to require an agent to account for money spent.
3. to ask for authoritatively or imperatively; demand.
4. to impose need or occasion for; make necessary or indispensable: The work required in finite patience.
5. to call for or exact as obligatory; ordain: The law requires annual income-tax returns.

Requisite

adjective

1. required or necessary for a particular purpose, position, etc.; indispensable: the requisite skills of an engineer.

noun

2. something requisite; a necessary quality, thing, etc.

Request

noun

1. the act of asking for something to be given or done, especially as a favor or courtesy; solicitation or petition: At his request, they left.
2. an instance of this: There have been many requests for the product.
3. a written statement of petition: If you need supplies, send in a request.
4. something asked for: to obtain one's request.
5. the state of being asked for; demand.

Con las precisiones indicadas más arriba, requerimiento se refiere a las solicitudes, las necesidades y los deseos de los clientes-usuarios que aún no han sido analizados desde su viabilidad, oportunidad, costo, u otra característica del producto software. Por lo tanto, se desconoce si finalmente se convertirán en requisitos del software. Mientras que un requisito es todo aquello que se ha definido y consensuado que el Sistema de Software “debe” satisfacer. Tal es su importancia que este conjunto de requisitos puede ser utilizado como un contrato entre el cliente y el proveedor del software.

Requisitos de Software

Un requisito de software es “un atributo necesario en un sistema, una sentencia que identifica una capacidad, características o un factor de calidad de un sistema en función de tener valor y utilidad a un cliente o usuario” [Young04]. Por lo tanto, los requisitos expresan las necesidades y las restricciones impuestas a un Sistema

de Software para dar solución a un problema o necesidad del mundo real [Swebok04] [Kotonya98] [Jackson95]. Según la [IEEE Std. 610-1990], un requisito de software es:

- “1. Una condición o capacidad que necesita un usuario para resolver un problema o alcanzar un objetivo.*
- 2. Una condición o capacidad que debe poseer un sistema o componente de un sistema para satisfacer un contrato, un estándar, una especificación, u otro documento formalmente impuesto.*
- 3. Una representación documentada de una condición o una capacidad como se describe en 1 y 2.”*

Los requisitos del software tienen **propiedades**, siendo estas las características que cada requisito debe cumplir. Para la [IEEE1233] estas propiedades son:

- Abstracto: cada requisito debe ser independiente de su implementación.
- No ambiguo: cada requisito debe ser expresado de tal manera que tenga una única interpretación.
- Rastreadable: se debe poder determinar la relación entre cada requisito y su fuente.
- Validable: debe existir la manera de probar que el sistema satisface los requisitos.

El estándar también hace referencia a la siguiente categorización de los requisitos:

- Identificación: cada requisito debe ser identificado de forma única.
- Prioridad: el cliente debe especificar la prioridad de cada requisito.
- Criticismo: es determinado por el ingeniero/a de requisitos en colaboración con el cliente.
- Viabilidad: es determinada por el cliente junto al ingeniero/a de requisitos para identificar la viabilidad de incluir cada requisito en el sistema.
- Riesgo: se deben utilizar técnicas de análisis de riesgo para calificar los requisitos.
- Fuente: cada requisito debe tener una identificación de su origen (autor). Sea esta única o múltiple.
- Tipo: cada requisito también se debe poder clasificar [Davis03] por uno o más de los siguientes tipos: entrada, salida, confiabilidad, disponibilidad, capacidad de

mantenimiento, desempeño, accesibilidad, condiciones ambientales, ergonomía, seguridad, transportabilidad, formación, documentación, interfaces externas, pruebas, calidad, políticas de regulación, compatibilidad con sistemas existentes, normas y políticas técnicas, conversión, capacidad de expansión.

Los requisitos del software también tienen **atributos** asociados, que permiten su identificación, clasificación o selección, y el control del mismo proceso de definición de requisitos. Algunos ejemplos de atributos:

- Kotonya y Sommerville [Kotonya98] proponen un meta-modelo para *especificar* requisitos con los siguientes datos: el identificador del requisito, descripción del requisito, fecha de creación, fecha de modificación, origen del requisito (persona, documento u otro requisito), fundamento, estado, lista de requisitos dependientes, lista de requisitos de los que depende, vínculos a modelos y un comentario.
- Davis [Davis99] propone para la *gestión* de requisitos los siguientes atributos: beneficio del cliente, esfuerzo de implementación, prioridad de desarrollo, estado, autores, parte responsable, fundamento, fecha de creación o modificación, versión y relación con otros requisitos. Donde *autores* corresponde al responsable de escribir la ERS o quién identificó la necesidad; y la *parte responsable* es quien asegura la satisfacción del requisito.
- SWEBOK [Sawyer01] propone tanto para la *gestión* como para la *interpretación* de los requisitos, incluir los siguientes atributos: la clasificación en distintas dimensiones (RF o RNF, derivado o impuesto, sobre el producto o el proceso, prioridad, alcance, volatilidad - estabilidad), el método de verificación o plan de prueba de aceptación, el fundamento, el origen del requisito, la historia de cambios y un identificador único para cada requisito.

Asegurar la calidad de los requisitos del software requiere en primer lugar, de una buena comunicación y comprensión entre el ingeniero/a de requisitos y los clientes-usuarios, poniendo énfasis en desarrollar o mejorar los procedimientos de obtención de requisitos que faciliten la cooperación y el compromiso de todos aquellos que participan de manera activa en el proceso.

Desde el punto de vista del conjunto de los requisitos, para mejorar la claridad de la ERS, los mismos suelen ser clasificados por algún criterio funcional, por su origen, por algún aspecto temporal o por algún otro criterio que se adapte mejor al contexto. Sommerville [Sommerville07] define dos tipos de requisitos que se dan en diferente momento del proceso. En primer lugar, los Requisitos de Usuario, destinados al usuario final y que son declaraciones de los servicios que se espera que el sistema realice y sus restricciones. Luego, los Requisitos del Sistema, o la denominada ERS, que contiene el detalle de los servicios y restricciones de los requisitos de usuarios, con el agregado de aspectos técnicos. Por otro lado, [Young04] define cuatro tipos de requisitos: Requisitos del negocio, Requisitos del usuario, Requisitos del producto y Requisitos del ambiente. Los requisitos del negocio son la base del desarrollo del software y son derivados de los objetivos del negocio. Mientras que los de usuario capturan la visión del cliente, el alcance del sistema, la estimación de costos, etc. necesarios para construir el Sistema de Software. Los requisitos del producto son los del desarrollo. Los del ambiente son los que se determinan por aspectos sociales, físicos y culturales en el cual el Sistema de Software será utilizado.

La clasificación más consensuada por la comunidad informática es la de tipificar los requisitos en funcionales y no funcionales [Loucopoulos95] [Swebok04] [Aurum05] [Sommerville07].

Los **requisitos funcionales** (RF) son conocidos y aceptados como capacidades o servicios que el Sistema de Software deberá realizar. Los **requisitos no funcionales** (RNF) sufren de una gran dispersión en las definiciones existentes. Se los identifica por ciertas cualidades-atributos-propiedades del nuevo Sistema de Software, como ser restricciones de la solución y algunos aspectos de calidad. En resumen, un requisito no funcional es un aspecto del nuevo Sistema de Software aplicado a la seguridad, eficiencia, rendimiento, disponibilidad, accesibilidad, usabilidad, estabilidad, portabilidad, interoperabilidad, escalabilidad, concurrencia, mantenibilidad, confiabilidad, etc., muchos de los cuales serán operacionalizados cuando sean aplicados a uno o varios RF en alguna etapa de la construcción del software.

A continuación, se describen algunas definiciones de RNF:

[Davis93] *“son atributos requeridos del sistema, incluyendo portabilidad, confiabilidad, eficiencia, la ingeniería humana, capacidad de prueba, compresión y capacidad de modificación”*

[Kotonya98] *“son requerimientos que no se refieren específicamente a la funcionalidad de un sistema. Se imponen restricciones sobre el producto que se está desarrollando y el proceso de desarrollo, y que especifican restricciones externas que el producto debe cumplir”.*

[Jacobson99] *“es un requerimiento que especifica propiedades del sistema, tales como restricciones de implementación y de ambiente, performance, dependencias de plataforma, mantenimiento, capacidad de extensión y confiabilidad. Un requerimiento que especifica restricciones físicas sobre un requerimiento funcional”*

[Cysneiros04] *“son requerimientos de calidad, que representan restricciones o las cualidades que el sistema debe tener tales como: Precisión, usabilidad, seguridad, rendimiento, confiabilidad, performance, entre otras”.*

A continuación, se dan ejemplos de RF y RNF:

- (RF) El sistema debe mostrar el stock real disponible, stock físico menos pedidos, cada vez que producto es solicitado.
- (RF) El sistema debe permitir operaciones con tarjetas de créditos sólo de los bancos autorizados.
- (RF) El sistema debe alertar al usuario cada vez que se produzca un cambio en el estado de cuenta de algunos de los clientes de su cartera.
- (RNF) El sistema debe cumplir con normas IEEE para la conexión de los equipos.
- (RNF) El sistema debe tomar en cuenta las necesidades, requerimientos, reglas, política, misión, objetivos de cada área y estar disponible el 80% del día tomando en cuenta que cada área de la empresa tiene un horario diferente; mientras que el 20% del tiempo restante es para tareas administrativas sobre el Sistema de Software.

Una pregunta subyace cuando se habla de requisitos del software:

¿Los requisitos se definen o son algo así como pepitas de oro que hay que encontrar?

Algunas visiones sostienen que los requisitos ya existen en el macrosistema de la aplicación. En cambio, otras aluden a que los requisitos deben ser construidos. Sin lugar a dudas no se puede estar en una única postura ya que algunos requisitos preexisten y otros deben ser definidos. En este último caso, algunos requisitos los genera el cliente-usuario, otros serán generados por el ingeniero/a de requisitos y finalmente otro grupo serán construidos en forma conjunta en una actividad de negociación.

Origen de los Requisitos

En la sección anterior se describió como se definen algunos requisitos del software, pero en este contexto surge otra pregunta, quizás la más básica y elemental, pero a su vez la más significativa de la Ingeniería de Requisitos:

¿Quién define los requisitos del software?

En el dominio de los sistemas de información, una respuesta posible es “los clientes y usuarios”. Es posible que esto ocurra, pero probablemente estas personas no tengan la información suficiente como para concebir el conjunto de prestaciones posibles del Sistema de Software. Otra respuesta posible es “los desarrolladores del sistema”. Esto puede ocurrir, pero probablemente estas personas no tengan la información suficiente acerca del proceso del negocio y menos aún del impacto que el Sistema de Software tendrá en el mismo. Parece que la mejor solución es que los requisitos sean definidos en conjunto por ambos grupos de actores (involucrados y desarrolladores). Esto determina con precisión qué actitud debe adoptar el ingeniero/a de requisitos frente a un Proceso de Requisitos. No debe preguntar qué es lo que quieren los clientes-usuarios porque de esa manera incentivan la definición de los requisitos por parte de ellos. La pregunta correcta es “*¿qué hacen?*” para promover que los desarrolladores busquen información sobre las características del proceso del negocio y esto los habilite a participar en la definición de los requisitos del sistema junto a los clientes-usuarios. Como consecuencia de todo lo anterior ocurre que no se debe pensar en elicitación de los requisitos, sino que se debe **elicitación de información**. Todavía queda un aspecto algo oscuro en relación a cómo se deben pensar los requisitos que se definirán en conjunto. Aquí es muy importante comprender que hubo un acercamiento entre los clientes-usuarios con el equipo de desarrollo, porque fueron los clientes-usuarios quienes manifestaron la necesidad de incorporar un Sistema de Software

para un objetivo específico, que es central en el Proceso de Requisitos ya que define la intención de los clientes-usuarios, o sea, el Objetivo General del Sistema. Lo próximo que se debe hacer es planificar el proceso del negocio futuro, suponiendo la existencia de ese software deseado. Esta actividad la deben desarrollar en conjunto entre los clientes-usuarios y el ingeniero/a de requisitos. Una vez hecho esto, los requisitos habrán quedado definidos en un documento, en un modelo o lo que sea que se haya construido para describir ese proceso de negocio planificado. Naturalmente que el peso de construcción de los modelos recae sobre el ingeniero/a de requisitos y la participación de los clientes-usuarios se materializa respondiendo preguntas aclaratorias, seleccionando alternativas, validando los modelos confeccionados, etc. Para que los clientes-usuarios puedan cumplir acabadamente su rol en estas actividades, es necesario que comprendan con precisión los modelos que se construyen. En este sentido los modelos se transforman en el medio más importante de comunicación. Es necesario enfatizar esto, tanto como sea posible, ya que, al seleccionar opciones, al responder consultas, o al validar el proceso del negocio planeado, los clientes-usuarios toman decisiones importantes que afectarán en forma directa al proceso de negocio cuando el software este en servicio. Es responsabilidad del ingeniero/a de requisitos una descripción clara y precisa y es responsabilidad de los clientes-usuarios asegurarse que comprenden los modelos presentados para poder analizar la viabilidad en función de los objetivos de la organización.

El punto clave de todo lo anterior es la claridad y facilidad de lectura de estos modelos. Por lo que se deben evitar construcciones sintácticas o gráficas complicadas procurando lograr una narrativa lo más simple posible. Aquí hay un compromiso claro entre el nivel mínimo de estructuración necesaria para que el documento sea de utilidad a los desarrolladores manteniendo la claridad y simplicidad que se debe exhibir a los clientes-usuarios. Todo lo anterior deriva en la construcción de descripciones en lenguaje natural con una sintaxis mínima que no entorpezca su lectura y facilite su comprensión.

Se puede concluir que tanto la actitud del equipo de requisitos como la de los clientes-usuarios son determinantes para asegurar la correcta comprensión del proceso del negocio. De esta manera, con una buena colaboración entre las partes interesadas y procesos que ayuden a identificar inconsistencias, omisiones y errores se incrementa sensiblemente la calidad de los requisitos del software.

Uso del Lenguaje Natural en los Procesos de Requisitos

Durante muchos años la frase “una imagen vale más que mil palabras” justificó la existencia de modelos gráficos en los procesos de construcción del software. Esta frase fue dejándose de lado al vislumbrar que muchos proyectos de software fracasaban debido a ERS incorrectas e incompletas, muchas de las cuales tenían su origen en la utilización de modelos que los clientes-usuarios no podían comprender. Por ejemplo, en el Análisis Estructurado se ha hecho referencia a “la vaguedad y embrollo del lenguaje corriente” [Gane77] con el cual se justificó el uso excesivo de herramientas gráficas durante el Análisis de Sistemas.

Promover que los participantes de un Proceso de Requisitos comprendan todos los modelos construidos mejora significativamente la comunicación y aumenta el compromiso. La comunicación es un componente de gran importancia en un Proceso de Requisitos y se presenta en diferentes direcciones. Como se muestra en la Figura 14 el canal principal se da entre los clientes-usuarios y el ingeniero/a de requisitos, siendo además un espacio bidireccional. Asegurar este canal es de suma importancia ya que de él depende la calidad de los requisitos del software. Paralelo a este canal, existen otros dos canales que funcionan dentro del propio grupo de los clientes y usuarios y dentro del propio grupo de los ingeniero/as de requisitos, mejorando la consistencia interna de cada grupo.

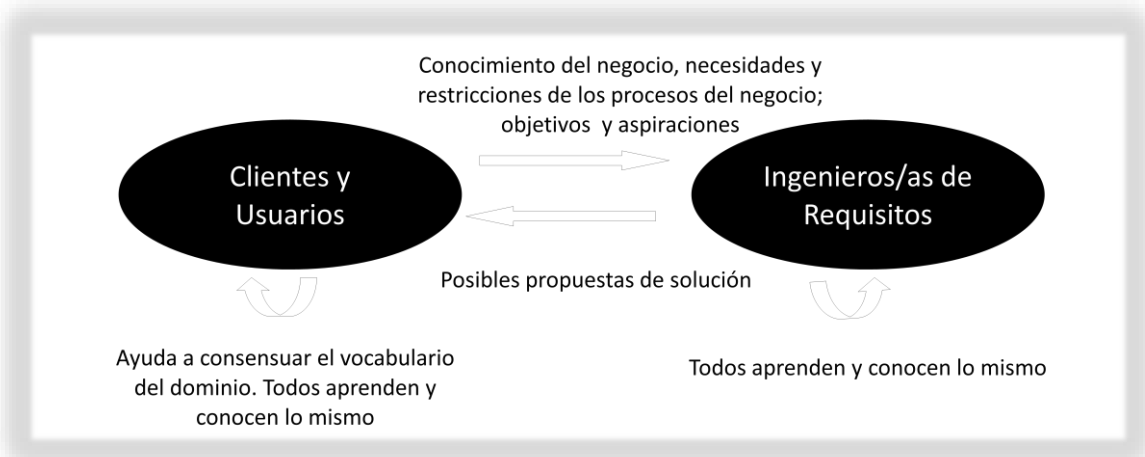


Figura 14 – Canales de comunicación en el Proceso de Requisitos

El lenguaje natural (LN) es un denominador común entre el equipo de desarrollo y los clientes-usuarios que puede ser utilizado con un mínimo esfuerzo de aprendizaje previo. Dicho esfuerzo se debe a la necesidad de estructurar mínimamente el uso del lenguaje

natural para garantizar que se cumplan los objetivos mencionados [Mille13] [Kłosowski18]. Son ampliamente conocidas las desventajas del uso del lenguaje natural, pero pueden ser mitigadas. Entre las desventajas más importantes se encuentra la ambigüedad y la imprecisión [Berry02], además de otras inconsistencias [Gervasi05] [Gervasi98]. Estos aspectos negativos han sido una de las justificaciones para no utilizarlo en los procesos de requisitos, pero se ha comprobado que pueden ser controladas sin que esto afecte la legibilidad y simplicidad del texto, retomando la ventaja más significativa del uso del lenguaje natural que es asegurar una buena comunicación. A continuación, se describen las recomendaciones de Berry para utilizar lenguaje natural:

1. Aprender a escribir con menos ambigüedad e imprecisión.
2. Aprender a detectar la ambigüedad y la imprecisión.
3. Usar un lenguaje natural restringido para que sea menos ambiguo y más preciso.

El uso del lenguaje natural en los documentos de requisitos está largamente difundido [Mille13], llegando al 93% del total de todos los documentos existentes, y dejando sólo un 7% a otras representaciones. En la Figura 15 se muestra como se distribuye ese 93% según un estudio realizado en la Università di Trento en Italia y disponible en eprints.biblio.unitn.it.

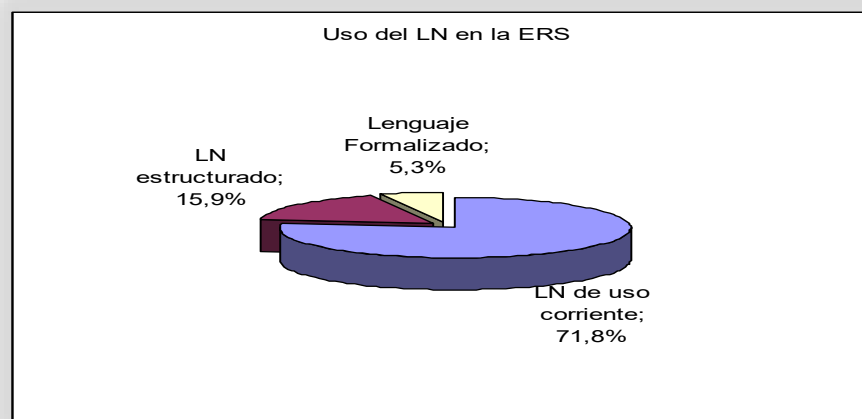


Figura 15 - Uso del LN en los documentos de requisitos [Berry02]

El lenguaje natural es utilizado en modelos como los Casos de Uso [Jacobson92] que son descripciones narrativas de la interacción entre un actor y el sistema, en los Escenarios [Booch92] [Potts94] [Wirfs-Brock95] que son descripciones de situaciones del contexto,

en los glosarios como el LEL [Leite93] u otros glosarios como [Whitenack94] [Oberg98] [Kovitz98] [Rolland98] [Regnell99] [Alspaugh99], en modelo de Reglas de Negocio [Ross97] [Leite98] [Gottesdiener99], en Historias de Usuarios (User-Stories) [Beck00], entre otros.

El Documento de Requisitos

El documento más relevante para el cliente que se genera en la etapa de Ingeniería de Requisitos es la ERS. Su importancia radica en asegurar que la realidad que se planifica para cuando el nuevo Sistema de Software este en uso, cumpla con las necesidades de los clientes-usuarios.

Existen diferentes formas de representar los requisitos. En primer lugar, como Especificación de Requisitos de Software [Hull05] o en plantillas donde se pueden definir atributos según las necesidades de cada caso en particular [Failey97] [Hull05] [Kerkow05] [Davis03] [Whitten03].

En general, las ERS son narrativas explícitas en lenguaje natural que permiten comunicar a todos los participantes e interesados del proyecto acerca de los servicios del software. También puede servir como contrato entre los clientes y el equipo de desarrollo. Las ERS utilizan diferentes estructuras [Kotonya98], aunque existen estándares internacionales que los especifican, tal es el caso de la IEEE Std 830-1998 [IEEE830] (ver Figura 16), IEEE Std P1233/D3-1995 [IEEE1233], DOD 5000.2-R [DOD5000] entre otros.

Si bien la ERS es suficiente para establecer qué hará el Sistema de Software, no siempre lo es para quienes deben realizar el diseño del software. Por tal motivo, la ERS puede estar acompañada por otros artefactos con el objetivo de mostrar diferentes puntos de vista de la solución. En general estos documentos (gráficos, esquemas, etc.) adicionales se realizan con el objetivo de mejorar la comprensión de los requisitos, pero es necesario destacar que no deben incorporar información, sino que deben mostrarla desde diferentes ángulos.

La ERS es un documento donde se definen las bases para un acuerdo entre el cliente y el proveedor (equipo de desarrollo) donde se establece lo que el producto de software debe realizar [Jalote05]. También puede ser utilizado, como ya se mencionó, como contrato entre las partes. Su función es de columna vertebral para todo el proceso de construcción

de software, dando los lineamientos de lo que se debe hacer. La ERS es utilizada para validar cada etapa posterior del proceso de construcción del software.

Existen dos maneras de encarar una ERS:

1. Como un modelo autónomo, su construcción puede estar relacionada con modelos anteriores (si existen) pero su generación es un proceso independiente.
2. Como el resultado de un proceso [Booch99] [Leite04], los requisitos están alojados en un modelo previo a la ERS. Por ejemplo, en los Casos de Uso o en los Escenarios Futuros (estos últimos se describen en el **Capítulo 3**).

En el primer caso, cuando la ERS es autónoma del proceso, se requiere de una exhaustiva actividad de V&V sobre el documento de requisitos, con el objetivo de asegurar que su contenido es correcto, razonablemente completo, sin ambigüedad, consistente, verificable, modificable y rastreable. Mientras que en el segundo caso (en el que se encuadra el proceso utilizado en esta tesis), la actividad de V&V de la ERS se reduce a un mínimo, debido a que todos los requisitos se alojaban en modelos previos a la ERS que ya han sido verificados y validados oportunamente.

En ocasiones, los requisitos están representados en los mismos modelos utilizados, como sucede con los Casos de Uso. En otros casos, los modelos son el medio para construir los documentos de requisitos, o sea son previos a la ERS.

Como ya se mencionó, el estándar [IEEE830] describe cómo crear la ERS y cuál debe ser su contenido para mantener su calidad. Según este estándar, una buena ERS debe cumplir con las siguientes propiedades:

- a) **Correcta:** Una ERS es correcta si, y sólo si, cada requisito declarado es uno que el software debe cumplir.
- b) **No ambigua:** Una ERS es no ambigua si, y sólo si, cada requisito declarado tiene sólo una interpretación. Como un mínimo, se requiere que cada característica del producto final se describa usando un único término.
- c) **Completa:** Una ERS está completa si, y sólo si, todos los requisitos significativos están relacionados a la funcionalidad, rendimiento, restricciones del diseño, atributos o interfaces externas. En particular todo requisito externo impuesto por una especificación del sistema debe tratarse.

- d) **Consistente:** una ERS consistente se refiere a la consistencia interna del documento. Si una ERS no está de acuerdo con algún documento del nivel superior, como una especificación de requisitos de sistema, entonces no es correcta
- e) **Ordenada por importancia y/o estabilidad:** una ERS es priorizada por importancia y/o estabilidad si cada requisito tiene un identificador para indicar la importancia o estabilidad de ese requisito en particular.
- f) **Verificable:** Una ERS es verificable si, y sólo si, cada requisito declarado es verificable. Un requisito es verificable si, y sólo si, existe algún proceso de costo efectivamente finito con el que una persona o una máquina puede verificar que el producto de software reúne el requisito. En general cualquier requisito ambiguo no es verificable.
- g) **Modificable:** Una ERS es modificable si, y sólo si, su estructura y estilo son tales que puede hacerse cualquier cambio a los requisitos fácilmente, completamente y de forma consistente mientras conserva la estructura y estilo.
- h) **Rastreable:** Una ERS es rastreable si el origen de cada uno de sus requisitos está claro y si facilita las referencias de cada requisito en el desarrollo futuro o documentación del mismo.

Al analizar las propiedades antes descriptas, aparece un interrogante referido al ítem e) acerca del concepto de “completitud”: ¿Cómo saber cuándo se tienen “todos” los requisitos significativos? ¿Existe algún mecanismo para determinar el grado de cubrimiento de los requisitos? Otro aspecto que genera duda está relacionado con la rastreabilidad, ya que no siempre se conoce el origen de cada requisito. Si bien es lo deseable, en la realidad pueden existir requisitos originados, por ejemplo, de una charla casual no documentada.

La Figura 16 es un resumen del contenido de la ERS. La estructura que sugiere este estándar es una guía general, dando las pautas del contenido de cada sección: la sección 1 describe los objetivos y el alcance del sistema, e incluye un glosario de términos utilizados en el documento; la sección 2 describe la funcionalidad general del sistema y los factores que afectarán al sistema y al proceso de desarrollo; en la sección 3 se describe en detalle los requisitos del software. El estándar ofrece ocho patrones para organizar los requisitos alternativos (ver ítem “3.7 Organizar los requisitos específicos” del estándar)

de la manera más clara según las necesidades o tipo de información particular de cada proyecto.

La ERS puede ser usada para registrar y gestionar cómo los requisitos son satisfechos y qué fragmento del Sistema de Software fue originado en un cierto requisito, como un ancla para el rastreo hacia adelante y hacia atrás, como ancla para la gestión de cambios en los requisitos, como herramienta de validación de los requisitos con los usuarios, como puente de comunicación entre el equipo de analistas y el equipo de diseñadores, como instrumento para probar el Sistema de Software.

-
- 1. Introducción
 - 1.1. Propósito del documento de requisitos
 - 1.2. Alcance del proyecto
 - 1.3. Definiciones, acrónimos y abreviaturas
 - 1.4. Referencias
 - 1.5. Resumen del resto del documento
 - 2. Descripción General
 - 2.1. Perspectiva del producto
 - 2.2. Funciones del producto
 - 2.3. Características de los usuarios
 - 2.4. Limitaciones generales
 - 2.5. Suposiciones y dependencias
 - 3. Requisitos Específicos
 - Requisitos funcionales, no funcionales y de interfaces
 - Apéndices
 - Índice

Figura 16 – Contenido de una ERS [IEEE830]

Los Procesos de Requisitos

Un *proceso* es un “conjunto de actividades relacionadas o que interactúan, las cuales transforman elementos de entrada en resultados” [ISO9001].

Como se describe en la Figura 17, la entrada al Proceso de Requisitos proviene del contexto organizacional, mientras que la salida se divide en dos partes, que por su heterogeneidad generan un conflicto a la hora de decidir los métodos, estrategias, formas de representación en los procesos de requisitos, provocando un nuevo interrogante:

¿Quién es el destinatario de la Especificación de Requisitos de Software?



Figura 17 – El proceso del Proceso de Requisitos

La primera respuesta es el cliente. Pero no se puede desconocer que la Ingeniería de Requisitos se parte de un proceso mayor. La Visualización de Requisitos [Duan06] [Gotel07], que transforma un requisito en lenguaje natural a una representación gráfica, puede ser una de las otras salidas de la Ingeniería de Requisitos. Estas técnicas permiten analizar los requisitos del software desde diferentes perspectivas, pero no se conoce aún su destinatario final. Considerando el salto conceptual que sugiere pasar de un texto a un gráfico, el análisis de estas visualizaciones con los clientes-usuarios deberá estar limitado a ocasiones muy concretas. En general, su uso se encuadra dentro de la propia Ingeniería de Requisitos como una técnica para mejorar la negociación y durante la gestión de requisitos o adjuntados a la ERS para el Diseño.

Los procesos de requisitos conforman un conjunto de actividades relacionadas entre sí que a su vez pueden contener sub-actividades dependientes. En la Tabla 6, se describen algunos procesos de requisitos y se presentan las actividades que los componen. Puede observarse que existen varias coincidencias, a pesar de que cada autor contempla una perspectiva propia de cómo se debe llevar a cabo el proceso. Por ejemplo, la actividad Analizar de Leite corresponde a negociar, verificar y validar los modelos construidos, mientras que para Kotonya & Sommerville, Analizar es construir los modelos y verificarlos, y para Thayer & Dorfman es equivalente a la actividad Modelar de Leite.

[Leite01]	[Loucopoulos95]	[Sawyer01]	[Kotonya98]	[Thayer97]
<ul style="list-style-type: none"> • Elicitar • Modelar • Analizar • Gestionar 	<ul style="list-style-type: none"> • Elicitar • Especificar • Validar • Gestionar 	<ul style="list-style-type: none"> • Elicitar • Analizar • Especificar • Validar 	<ul style="list-style-type: none"> • Elicitar • Analizar y Negociar • Documentar • Validar • Gestionar 	<ul style="list-style-type: none"> • Elicitar • Analizar • Especificar • Verificar • Gestionar

Tabla 6 – Actividades de los Procesos de Requisitos

El Proceso de Requisitos [Leite01], en el que se basa el estudio de la presente tesis, consta de cuatro actividades, como puede observarse en la Figura 18. Las tres primeras se presentan, por lo general, en forma secuencial, pero pueden ser realizadas en forma concurrente. En cambio, la actividad Gestionar comienza cuando se tiene la primera versión de la ERS y continúa hasta que el Sistema de Software muere. A continuación, se describe cada actividad:

Elicitar: es una actividad donde se identifican las FI que contienen la información del macrosistema. Estas FI deben ser priorizadas teniendo en cuenta, entre otras características, el tipo de información que proveen y la disponibilidad para acceder a ella.

El tipo de información adquirida tiene una fuerte relación con su origen y puede ser relacionado con la pirámide organizacional. Cuando la información proviene del nivel más alto de la pirámide o este nivel es el destino del mismo, se refiere por lo general, a objetivos y políticas organizacionales. Mientras que en un nivel medio pueden convivir distintos niveles de detalle. Finalmente, en la base de la pirámide la información es puntual sobre tareas específicas. La disponibilidad de la FI es un elemento para determinar la técnica de elicitación más adecuada. Muchas de estas técnicas tienen su origen en las Ciencias Sociales [Goguen93] [Bertolin98]. Una vez seleccionada la técnica de elicitación comienza la recolección de hechos y estos deben ser representados en algún modelo.

Modelar: es una actividad donde se representa en uno o varios modelos los hechos recolectados durante la elicitación. Existen diferentes modelos que permiten resguardar la información del macrosistema, como son los Casos de Uso [Jacobson92], los Escenarios [Booch92] [Potts94] [Wirfs-Brock95], UML (*Unified Modeling Language*) [Daoust12], LEL (Léxico Extendido del

Lenguaje) [Leite94], entre otros.

Analizar: es donde se verifica, negocia y valida todo lo modelado. Se trata de descubrir y resolver los problemas que se presentan en los modelos, entre los que se pueden citar: inconsistencias, omisiones, ambigüedad, contradicciones, errores, etc.

La verificación y la validación pueden ser realizadas concurrentemente o en actividades por separado. En la etapa de Ingeniería de Requisitos existen diversas técnicas de verificación y validación (V&V) que en su mayoría son adaptaciones que provienen de técnicas utilizadas en la etapa de diseño, codificación y prueba. Las técnicas de V&V se pueden clasificar en estáticas, dinámicas y formales [Wallace96]. Las estáticas se aplican en general a un documento y analizan la forma y su estructura. Algunas técnicas estáticas son las revisiones, las inspecciones, las auditorías, etc. Las técnicas dinámicas envuelven una ejecución o simulación, como ser un prototipo. Las técnicas formales utilizan rigurosas técnicas matemáticas y sus especificaciones utilizan algún lenguaje formal como el Z y el VDM. Algunas técnicas de verificación se basan en máquinas de estado, inspecciones, análisis de consistencia, etc. Mientras que técnicas de validación son Storyboard, Prototipos, Reviews o Walk-throughs, Casos de prueba, etc.

Según la [IEEE610] la verificación y la validación es:

“el proceso de determinar si los requisitos para un sistema o componente son completos y correctos, si los productos de cada fase de desarrollo cumplen con las condiciones impuestas por la fase anterior, y si el sistema final o componente final cumple con la ERS.”

Existen en la literatura dos preguntas [Boehm84] [Pressman05] muy útiles que delimitan claramente estas dos actividades. Cuando se Verifica se debe interrogar sobre si “¿estamos construyendo el producto correctamente?”, mientras que para validar la pregunta apunta a si “¿estamos construyendo el producto correcto?”.

Es durante la sub-actividad Negociar, donde se discute la resolución de conflictos, las diferentes propuestas de solución, la asignación de prioridades a

los requisitos, etc. Pero en la definición de los RNF es donde se encuentra la mayor cantidad de conflictos y es aquí donde la negociación incrementa su influencia. Entre estos conflictos se pueden citar como ejemplo los siguientes pares de conflictos: mantenibilidad - eficiencia o seguridad – disponibilidad [Kotonya98]. Si bien esta actividad encuentra un momento en el proceso de desarrollo del software donde se despliega a pleno, puede ser concurrente y/o espontánea en el momento de elicitar y de validar.

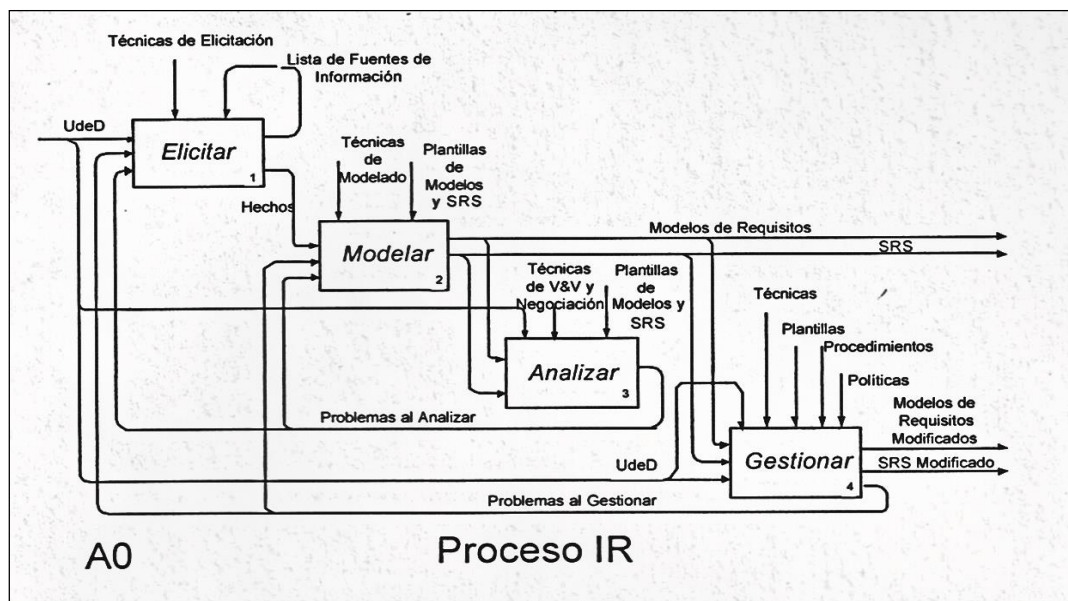


Figura 18 – SADT⁶ del Proceso de Requisitos [Leite01]

Gestionar: es la actividad donde se identifican, analizan y realizan los cambios en los requisitos. Estos cambios pueden ser generados por requisitos nuevos, modificaciones a requisitos existentes (errores, mejoras, etc.) o requisitos que evolucionan. Una vez identificado el cambio se debe establecer la dependencia con otros requisitos y el costo que este cambio producirá. Cuando el cambio es aprobado se deben actualizar los documentos y modelos de requisitos para luego hacer las modificaciones sobre el diseño y el código.

Durante la Gestión de Requisitos es necesario poder rastrear los requisitos del software [Gote194] [Wieringa95] [IEEE830] [Pinheiro04] en ambas direcciones,

⁶ Structured Analysis and Design Technique (SADT): es un método gráfico que adopta un enfoque puramente jerárquico del problema con una sucesión de planos que se refinan hasta lograr una solución. La caja representa una actividad (Actigramas) o datos (Datagramas).

a través de todos los períodos de continuo refinamiento e iteración en cualquiera de sus fases.

A continuación, se presentan dos ejemplos de empresas de software que promocionan la gestión de requisitos como una ventaja competitiva:

IBM: *“Ayuda a reducir las tareas de remodelación y acelera el tiempo de comercialización con los accionistas, a aumentar la productividad mediante el control y la gestión de los cambios en los requisitos, a minimizar los costos y los riesgos midiendo el impacto de los cambios a medida que se producen y a demostrar la conformidad gracias a la rastreabilidad completa de los requisitos. Aumenta la productividad con un estricto control.”*

Siemens: *“Gracias a la perfecta integración con la gestión de requisitos podrá recabar la documentación de requisitos desde diversas fuentes, adaptarla a requisitos individuales y mantener dichos requisitos en el mismo entorno empleado para la gestión del ciclo de vida del producto (PLM). Esto implica que podrá incorporar requisitos a los procesos basados en flujos de trabajo que emplee para las decisiones de ingeniería, aprovisionamiento, planificación de la fabricación, calidad y cambios”.*

Priorización de Requisitos

Como se expresa en [Hadad09b], la tarea de asignar prioridades [Karisson89] [Park99] [Greer99] requiere de la participación de clientes y usuarios con cierto nivel de decisión y puede realizarse de diversas maneras, tales como reuniones, cuestionarios y otras. Se debe determinar la importancia relativa que tiene un requisito para los clientes y usuarios, y organizar aquellos requisitos que deben implementarse inicialmente frente a aquellos que pueden postergarse.

Al asignar prioridades, se deben tener en cuenta la dependencia entre requisitos, la multiplicidad de intereses de los clientes y usuarios, las limitaciones de recursos, las necesidades del negocio, las imposiciones del mercado y los costos de implementación, entre otros factores. Por ende, los ingeniero/as de software deberán orientar a los clientes y usuarios respecto a estas contingencias.

La asignación de prioridades ha sido muy estudiada por una variedad de autores que han definido estrategias simples o complejas, que van desde una asignación que califica al requisito como importante o no, como obligatorio o diferible hasta un ranking de

importancia de los requisitos. Dentro de estas técnicas de asignación de prioridades a requisitos se encuentran: Asignación Numérica [Brackett90] que divide los requisitos en tres grupos: obligatorios, deseables y no esenciales; Ranking se basa en una escala ordinal donde al requisito más importante se le da el valor 1 y al menos importante el valor N siendo N la cantidad de requisitos, y la lista de ranking se obtiene aplicando por ejemplo el método de ordenamiento burbujeo o árbol binario de búsqueda; AHP (Analytic Hierarchy Process) [Saaty80] [Karisson89] basada en la técnica de “pairwise comparison” estimando un valor relativo de importancia de los requisitos; QFD (Quality Function Deployment) [Zultner92] basada en la asignación numérica de prioridades respecto a una escala absoluta; Método de 100 Puntos [Leffingwell03] donde a cada participante se le dan 100 puntos para aplicar votando a favor de los requisitos más importantes para él; Teoría W (también denominada Win-Win) [Boehm89] [Park99] donde cada involucrado da un ranking privadamente a los requisitos considerando aquellos que está dispuesto a abandonar y luego se realiza la negociación, Requirements Triage [Davis03] incluye dar prioridades relativas a los requisitos, estimar los recursos necesarios para satisfacerlos y seleccionar un subconjunto de requisitos para optimizar la probabilidad de éxito; Juego de Planeamiento [Beck04] usado en Programación Extrema sobre las historias de usuarios donde se agrupan los requisitos en grupos y luego se dan rankings dentro de cada grupo; y SERUM [Greer99] usa estimaciones de costo, beneficio, riesgo de desarrollo y reducción de riesgo operacional para dar prioridades.

Otro aspecto a considerar adicional a asignar prioridades, es la compatibilización de las mismas debido a la dependencia entre requisitos. En este caso también se han propuesto varias técnicas, como por ejemplo: el método de Wiegers [Wiegers05] se basa en calcular una prioridad para cada requisito en base a cuatro atributos: beneficio para el cliente, penalidad si el requisito no se incluye, costo de implementación y riesgo técnico, asignando a cada atributo un valor entre 1 y 9, antes de calcular las prioridades se determina que todos los requisitos tengan el mismo nivel de abstracción y se considera si hay requisitos vinculados para incluir sólo el requisito dominante; EVOLVE [Greer05] basada en un algoritmo genético que considera distintos puntos de vista de los involucrados, restricciones de esfuerzo, restricciones de riesgo y dependencias entre requisitos; Win-Win [Ruhe02] que a diferencia de la técnica Win-Win de Boehm incluye métodos cuantitativos pues usa AHP para determinar las preferencias de los involucrados,

y estos resultados se combinan para evaluar la factibilidad de subconjuntos de requisitos alternativos considerando sus esfuerzos de implementación relativos; y Requirements Prioritization Framework [Moisiadis00] [Moisiadis01] donde los involucrados dan una tasa de importancia a los requisitos y a los objetivos del negocio, luego se buscan las dependencias de los requisitos para dar prioridades más eficientemente y finalmente se aplican técnicas de análisis de riesgo para descubrir valoraciones subjetivas o corporativas de los involucrados.

Rastreabilidad de los Requisitos

Muchos autores han propuestos diferentes formas de rastrear el software [Gotel94] [Wieringa95] [Palmer96] [Jarke98] [Leite03] [Pinheiro04] [Winkler10] [Wohlrab20]. La importancia de la rastreabilidad es asegurar que se estén construyendo los productos correctos en cada fase del ciclo de vida de desarrollo de software y que los cambios en los requisitos sean correctamente incorporados.

La rastreabilidad no sólo se utiliza para administrar los cambios en los requisitos, sino que también es de fundamental ayuda para la verificación y validación de los requisitos y para el control del proceso de desarrollo [Palmer96] [Davis99], pues facilita detectar conflictos utilizando los vínculos establecidos entre los elementos rastreables, posibilita asegurar que decisiones tomadas avanzado el desarrollo sean consistentes con decisiones tempranas, y permite verificar que todos los requisitos han sido implementados en el software, entre otros usos.

Entre algunas de las definiciones existentes se encuentran:

[Gotel94] *Trazabilidad de requisitos: se refiere a la capacidad de describir y seguir la vida de un requisito, tanto hacia adelante como hacia atrás.*

[IEEE610] *"El grado en que se puede establecer una relación entre dos o más productos del proceso de desarrollo, especialmente productos que tienen una relación predecesor-sucesor o maestro-subordinado entre sí; por ejemplo, el grado en que los requisitos y el diseño coinciden en un componente de software".*

[IEEE830] *La ERS es trazable si se conoce el origen de cada requisito y se facilita la referencia de cada requisito a los componentes del diseño y de la implementación. La trazabilidad hacia atrás indica el origen (documento, persona, etc.) de cada requisito. La trazabilidad hacia delante de un requisito R indica que componentes del sistema son los que realizan el requisito R.*

[Clarke02] *"La habilidad para determinar realmente cómo una pieza de un artefacto de software (requisito, diseño, código) afecta a otras. La rastreabilidad hace posible buscar un cambio a un requisito y encontrar sus partes en los detalles del diseño y el código que son afectados por el cambio. Es esencial para mantener los documentos de requisitos y diseño al día con respecto a la evolución del código".*

[ISO9001] *"Aquellos procedimientos preestablecidos y autosuficientes que permiten conocer el histórico, la ubicación y la trayectoria de un producto o lote de productos a lo largo de la cadena de suministros en un momento dado, a través de herramientas determinadas".*

Mantener un proceso de rastreabilidad en un proyecto, ayuda a:

- Identificar el origen del requisito, ya sea emitido por una persona, documento o grupo de personas.
- Realizar un análisis de impacto, rastrea qué otros componentes podrían verse afectados en respuesta a un cambio en un requisito en particular.
- Verificar los casos de prueba.
- Seguir el progreso general del proyecto

Cabe destacar que son Gotel and Finkelstein [Gotel94] (ver Figura 19) quienes introducen la distinción entre la rastreabilidad previa a la especificación de requisitos (pre-RS) y la rastreabilidad posterior a la especificación de los requisitos (post-RS). Un estudio posterior [Uzair08], en el cual se analizó qué definiciones se habían utilizado en un conjunto de trabajos publicados, determinó que el 80% habían citado a Gotel and Finkelstein, quienes describen dos tipos de rastreabilidad:

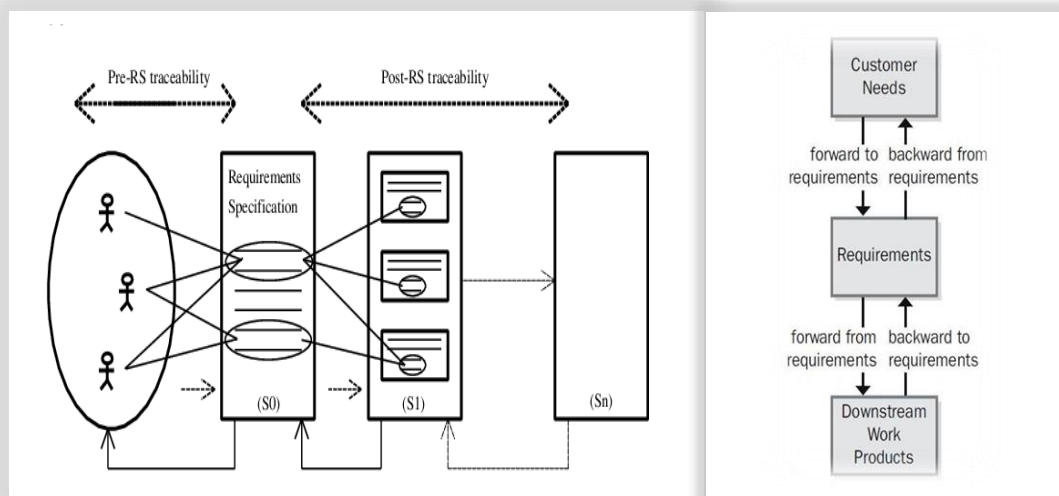


Figura 19 – Tipos de rastreabilidad de requisitos [Gotel94]

Pre-RS traceability: se refiere a aquellos aspectos de la vida de un requisito antes de su inclusión en la ERS.

Post-RS traceability: se refiere a aquellos aspectos de la vida de un requisito que resultan de la inclusión en la ERS

Para otros autores [Wieringa95] [IEEE830] [Pinheiro04] [Davis] [Sommerville98] (ver Figura 20) la clasificación más extensa pero incluye la de Gotel:

Backward-from traceability: Vincula los requisitos con sus fuentes, que se encuentran en otros documentos o de personas. Backward-from y Pre-RS traceability pueden ocurrir al mismo tiempo. Por ejemplo, cuando se desea conocer la fuente de un cambio en un requisito, se rastrea desde el requisito hacia la FI que aceptó el cambio.

Forward-from traceability: Vincula los requisitos con los componentes de diseño e implementación. Forward-from traceability y Post-RS traceability pueden ocurrir al mismo tiempo. Por ejemplo, cuando se detecta un error en un requisito y se rastrea hacia el código.

Backward-to traceability: Vincula los componentes de diseño e implementación con los requisitos.

Forward-to traceability: Vincula otros documentos a los requisitos relevantes. Estos otros documentos pueden ser, por ejemplo, manuales de funcionamiento que describen la funcionalidad del sistema.

En el caso de Pinheiro, además de los cuatro tipos mencionados, incluye dos tipos adicionales:

Inter-requirements traceability: se refiere a las relaciones entre requisitos de un mismo documento.

Extra-requirements traceability: se refiere a las relaciones entre los requisitos y otros artefactos.

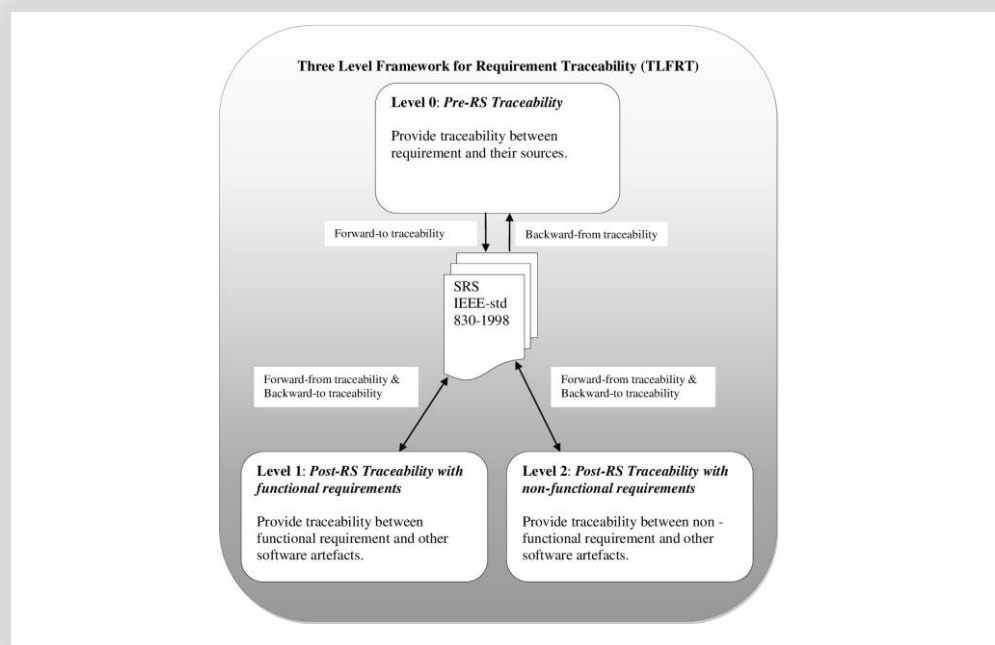


Figura 20 – Dimensión y dirección de la rastreabilidad de requisitos [Wang19]

Estrictamente los inter rastreabilidad no son rastros sino dependencias. Sin embargo, por cuestiones de costos se las suele tratar conjuntamente con los rastros porque se representan con las mismas estructuras.

Se debe tener en cuenta que la rastreabilidad de requisitos se puede tornar inevitable, ya que en sistemas medianos a complejos la cantidad de información que se debe manejar es muy grande y se hace necesario contar con un modelo de trazabilidad para gestionarla. En estos sistemas existe una red de relaciones bastante compleja. Es común que varios

requisitos provengan de la misma fuente, así como que un solo requisito tenga más de una fuente. También es común tener un requisito derivado de varios otros, así como varios requisitos que se colapsan en uno solo. La diversidad y la gran cantidad de información que existe cuando se desarrolla un Sistema de Software de gran tamaño, genera la necesidad de contar con un soporte automatizado para todo el desarrollo, incluida la rastreabilidad.

Puede observarse que la rastreabilidad descrita está centrada en la IR. Esto es esperable ya que, como se mencionó, es una actividad de la Gestión de Requisitos. Pero se debe tener en cuenta que la rastreabilidad debe ser un mecanismo independiente y que cubra todo el proceso de construcción del software. O sea, se generan rastros durante todo el proceso.

Capítulo 5

Reseña Histórica de la Ingeniería de Requisitos

Resumen

Revisar la historia reciente de la construcción del software permite apreciar algunos acontecimientos y hechos que han justificado la aparición de la IR. En el presente Capítulo se realiza una somera reseña que permite apreciar rápidamente cómo ha evolucionado el tratamiento de los requisitos del software en los últimos 50 años.

¿La “*crisis del software*” se ha convertido en una *enfermedad crónica*? Grady Booch afirma que “el desarrollo de software ha sido, es y probablemente será, fundamentalmente difícil”. Diferentes autores han escrito acerca de la historia de la Ingeniería de Software [Larman03], con el objetivo de marcar los avances y retrocesos producidos. Sólo a efectos de comprender esta evolución, se analiza la crisis del software [Dijkstra72] como uno de los desencadenantes de estos cambios. Cabe aclarar que en 1968, en la primera conferencia de la OTAN (Organización del Tratado del Atlántico Norte), Friedrich L. Bauer habló por primera vez del conjunto de dificultades o errores ocurridos en la planificación, estimación de costos, productividad y calidad del software, coincidiendo con lo que luego se conoció como la crisis del software. Algunos de los problemas se referían a proyectos de software que sobrepasaban el presupuesto; resultados poco confiables; grandes sistemas de software que resultaban difíciles y caros de mantener; software que no satisfacía las crecientes necesidades del cliente; dependencia del software con el hardware; aumento de la demanda de nuevo software que superaba la capacidad de generarlo; proyectos de software que en promedio sobrepasaba el 50% de su cronograma original; software, que en muchas ocasiones, nunca se entregaba; etc. A efectos de ejemplificar esta crisis, se utilizan las estadísticas de CHAOS Report by The Standish Group, los cuales han analizado 50.000 proyectos, clasificándolos en “Challenged” (con problemas), “Failed” (que nunca se han terminado) y Successful (exitosos, terminaron en tiempo y costo). En la Figura 21 se puede observar que los proyectos exitosos han tenido avances y retrocesos, existiendo dos momentos donde se puede observar un cambio significativo, en 1996 y en 2006. El techo de los proyectos exitosos ha sido del 39%, una cifra realmente alarmante. La media geométrica de los casos exitosos, descrita en la Figura 21 y representada en la Figura 22, es de apenas un 30,99. Se puede observar también que los proyectos fallidos han disminuido, pero dicha merma parece haber sido trasladada a los proyectos con problemas más que a los exitosos.

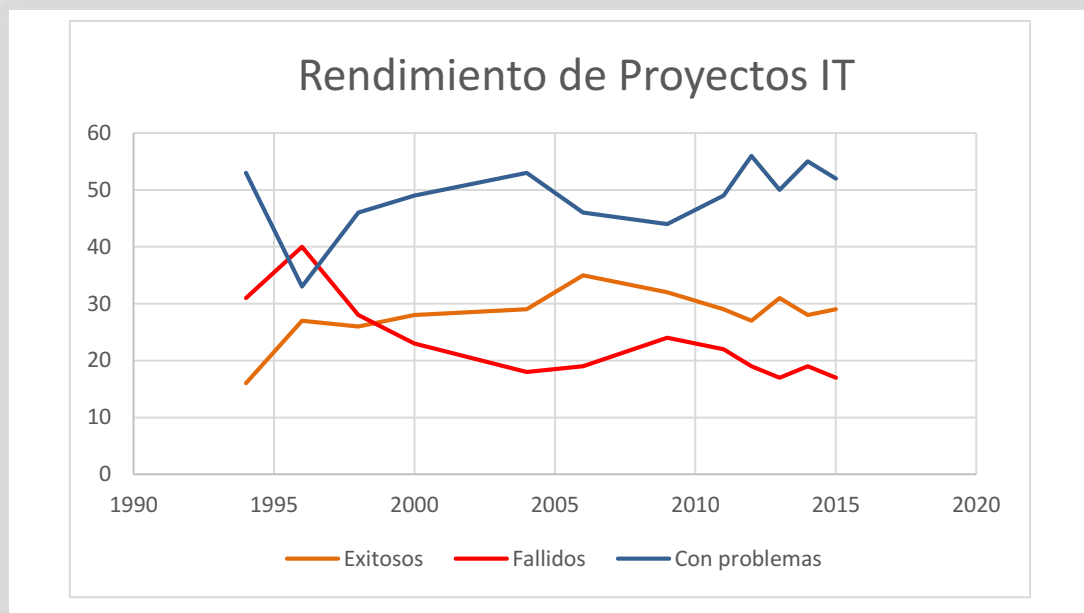


Figura 21 – Efectividad de proyectos IT entre 1994 y 2015

Tipo Proyecto TI	MEDIA GEOM
Exitosos	30.99
Con problemas	42.37
Fallidos	22.29

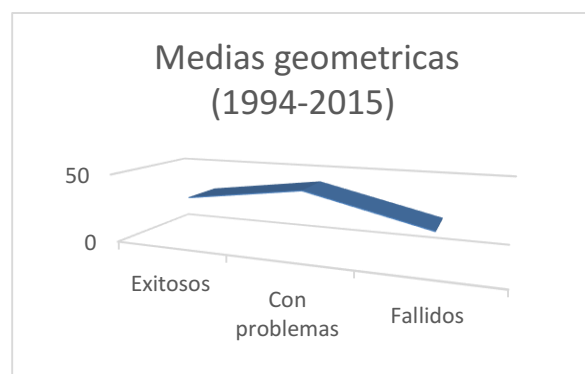


Figura 22– Medias geométricas de proyectos IT entre 1994 y 2015

Esta síntesis histórica, comienza en 1970, y toma como primer hito la publicación de Royce [Royce70]. Hasta ese momento la construcción de software se realizaba en pequeños ciclos de “Analysis and Coding” (análisis y codificación) el cual reemplazó al ciclo “Code and Fix” (codificar y corregir) utilizado durante la década 60. Royce propone abrir la construcción de software en más etapas, y define un ciclo abierto con las siguientes etapas⁷: Software Requirements, Analysis, Program Design, Coding, Testing and Operations. La propuesta es que cada etapa se complete antes de pasar a la siguiente. Royce considera que la mejor forma de construir software es sin retroalimentación entre

⁷ La primera mención conocida que describe el uso de las fases de la Ingeniería de Software fue realizada por Herbert D. Benington en el “Symposium on advanced programming methods for digital computers” en 1956.

fases, pero reconoce que es “arriesgada e inevitable al fracaso”. Por tal motivo, en el mismo artículo describe la manera de iterar entre etapas, pero haciendo mención a la incidencia de la retroalimentación en el costo del software indicando que retornar a etapas anteriores puede volver el proceso prácticamente imposible de sostener. La propuesta real de Royce se puede observar en la Figura 23, donde además de abrir las etapas del proceso, agrega un ciclo de prueba inicial de todo el proceso denominado *Diseño Preliminar del Programa* y lo ubica entre las fases de Requisitos del Software y la de Análisis. El objetivo de este ciclo es controlar el costo de posibles errores u omisiones en las etapas siguientes. De esta manera recomienda hacer dos veces el mismo trabajo, la primera vez para comprobar la viabilidad del proceso con un prototipo utilizable y la segunda para construirlo. Todo el conocimiento obtenido en esta fase es utilizado para alimentar las fases posteriores.

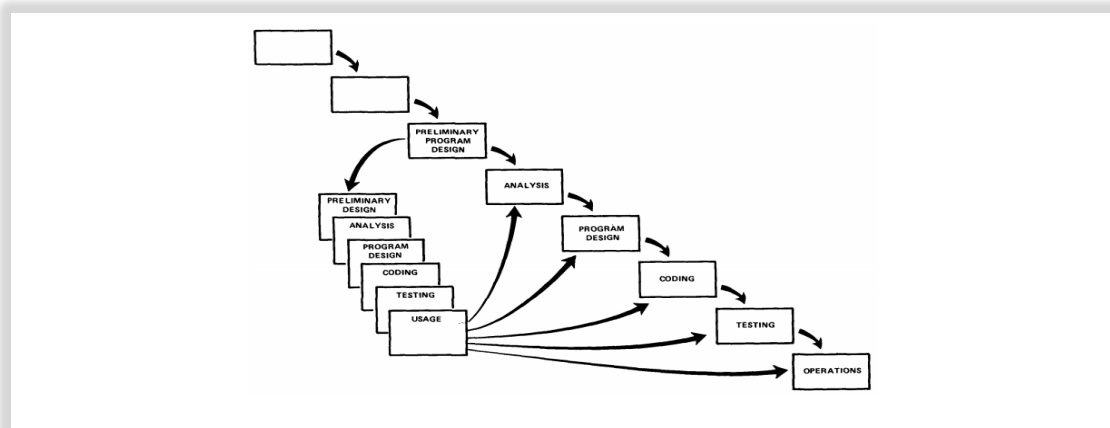


Figura 23 – Modelo propuesto por Royce

Las etapas propuestas por Royce fueron conocidas en 1975 como **SDLC** (Software Develop Life Cicle) o ciclo de vida del desarrollo del Software, el cual también ha sido fuertemente cuestionado, aludiendo que en vez de ayudar perjudica a la construcción del software al hacer muy dificultosa la incorporación de cambios en los requisitos [Gladden82]. McCracken y Jackson también argumentaron en contra del modelo de Royce haciendo referencia a una "cascada sofocante" [McCracken80] y subrayando la posición subordinada de los modelos iterativos. Brooks dejó muy claro su punto de vista acerca del modelo en cascada en el discurso de apertura en la Conferencia Internacional de Ingeniería de Software de 1995 al decir que "¡El modelo de cascada está mal!". Royce nunca menciona el tan conocido “modelo en cascada”, esta denominación (waterfall

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 5 - Reseña histórica de la Ingeniería de Requisitos

model) surgió con posterioridad, cuando se relacionó el trabajo de Royce con una “cascada” de actividades secuenciales [Bell76]. Luego, el modelo en cascada como tal, se popularizó a través de la norma estadounidense DoD-STD-2167 en el año 1985, donde se lo utilizó con una sola iteración, siendo la duda si se debía completar en una sola iteración o si podían sucederse otras. Este estándar influenciaría a varios otros estándares significativos de la industria, como el JSP-188 (Gran Bretaña), V-Model (Alemania), GAM-T-17 (Francia), entre otros. Según Larman y Basilli [Larman03] a pesar de la rigidez del modelo en cascada, se puede ver en el artículo de Royce indicios de desarrollo iterativo, retroalimentación y adaptación y, citan un fragmento de la conversación que tuvieron con su hijo, Walker Royce:

"Él siempre fue un defensor del desarrollo iterativo, incremental, evolutivo. Su artículo describe la cascada como la descripción más simple, pero eso no funcionaría para todos los proyectos, excepto aquellos más sencillos. El resto de su trabajo describe las [prácticas iterativas] en el contexto del modelo de contratación del gobierno de los 60s/70s".

Las diferentes interpretaciones de la propuesta original de Royce y lo que hoy se conoce como *modelo en cascada* se mantiene en la actualidad. Por ejemplo, en la definición del modelo en los libros de Ingeniería de Software de Pressman y Sommerville. En [Pressman05] el modelo en cascada cita a Royce y prevé ciclos con retroalimentación, pero indica que “la inmensa mayoría de las organizaciones que aplica este modelo de proceso lo trata como si fuera estrictamente lineal”. En [Sommerville11] se hace referencia al modelo en cascada como un modelo derivado de la propuesta de Royce. Toma como dato principal que las etapas del modelo original reflejan las actividades fundamentales del desarrollo y describe que cada fase itera hasta asegurar que está completa, y recién entonces se pasa a la fase siguiente, pero menciona que en la práctica las etapas se retroalimentan iterando entre ellas ya que una nutre a la otra.

En 1986, Parnas y Clements publican el artículo "Un proceso de diseño racional: cómo y por qué fingirlo", donde describen que, aunque creen en el ideal del modelo en cascada (especificaciones completas, correctas y claras antes del desarrollo), es poco práctico. Algunas de sus razones:

- Los usuarios de un sistema rara vez saben exactamente lo que quieren y no pueden expresar todo lo que saben.
- Incluso si pudiéramos establecer todos los requisitos, hay muchos detalles que sólo podemos descubrir una vez que están bien implementados.
- Incluso si supiéramos todos estos detalles, como humanos, no podemos dominar tanta complejidad.
- Incluso si pudiéramos dominar toda esta complejidad, las fuerzas externas conducen a cambios en los requisitos, algunos de los cuales pueden invalidar decisiones anteriores. y comentó que, por todas estas razones, "la imagen del diseñador de software que deriva su diseño de una manera racional y libre de errores de una declaración de requisitos es bastante poco realista".

La expresión “incluso si pudiéramos...” expone un conjunto de falencias en los procesos de construcción de software, particularmente en la primera fase del proceso, lo que luego fue absorbido por la Ingeniería de Requisitos. Pero esta necesidad de mejorar y asegurar los procesos de requisitos ya había sido declarada una década antes. En 1976 se llevó a cabo un estudio empírico [Bell76] de casos realizados en 1973 y vueltos a analizar en 1975. Dividieron los problemas en tempranos y tardíos, arrojando que en las revisiones tempranas aparece con más fuerza los problemas de requisitos "faltantes", "incompletos", "inadecuados", o errores de tipo "poco claros" y en las revisiones posteriores superó ampliamente la categoría requisitos "incorrectos" seguidos por requisitos “omitidos” e “incompletos”. En este estudio determinaron empíricamente que los problemas en los requisitos son una realidad y que requieren una revisión continua. Además, arribaron a que los problemas detectados son muy similares en todos los proyectos. Los requisitos cambian durante todo el desarrollo. Los errores en los requisitos sólo fueron detectados cuando este se acercaba al diseño, por lo tanto, es necesario contar con técnicas para identificar deficiencias en los requisitos lo más temprano posible para reducir costos y mejorar la calidad de los mismos.

Basili en [Basili75], recomienda la **técnica de mejora iterativa** como un medio práctico para realizar un enfoque de refinamiento gradual y top down para el desarrollo de software. De esta manera se pueden *analizar los requisitos para cada iteración*. El mecanismo consiste en crear un esqueleto inicial que sirva de guía para todo el proceso de construcción. Sobre este esqueleto se va iterando hasta tener la versión final. Crea una

lista de control del proyecto con todas las tareas necesarias para implementar el software deseado, y el proyecto finaliza cuando la lista está vacía. Las tareas pueden ser un rediseño, el diseño e implementación de futuras funcionalidades, la incorporación de funcionalidades que se han perdido o no han sido vistas en la implementación existente, y soluciones a problemas sin resolver. Cada iteración consiste en seleccionar y remover la próxima tarea de la lista, diseñando la implementación de la tarea seleccionada. Según Basili [Basili75], este esqueleto inicial puede traer algunos problemas, como ser una mirada parcial del problema en estudio. El esqueleto debe tener los puntos clave del problema y las restricciones. En cada iteración participan los usuarios y se incorporan las modificaciones necesarias. La utilización de la mejora iterativa al desarrollo de software, según los autores, es práctica y eficiente; fomenta la generación de un producto fácilmente modificable; ya que, en cada iteración, se realizan modificaciones de diseño junto con la incorporación de nuevas funcionalidades.

Según [Larman03], el *Desarrollo Iterativo*, ya estaba presente en 1968 en un informe de Randell y Zurcher en IBM. Un año más tarde, Lehman retomó el trabajo de Randell y Zurcher y en un informe interno para la gerencia insistió en recomendar el Desarrollo Iterativo diciendo que “el enfoque básico reconoce la inutilidad de separar los procesos de diseño, evaluación y documentación en el diseño de sistemas de software. El proceso de diseño está estructurado por un modelo en expansión sembrado por una definición formal del sistema, que proporciona un primer modelo funcional ejecutable. Se prueba y se expande aún más a través de una secuencia de modelos, que desarrollan una cantidad creciente de funciones y una cantidad creciente de detalles sobre cómo se debe ejecutar esa función. Finalmente, el modelo se convierte en el sistema”.

En realidad, el enfoque iterativo (ver Figura 24), surgió del trabajo de Walter Shewhart en 1930, un experto en calidad de Bell Labs que propuso una serie de ciclos cortos de "plan-do-study-act" (PDSA) [Shewhart39] para mejorar la calidad. A partir de la década del 40, otro gurú de la calidad, W. Edwards Deming, comenzó a promover vigorosamente el ciclo [Deming50] el cuál fue llamado en Japón “Ciclo Deming” en su honor [Deming82]. Otros autores, como Tom Gilb y Richard Zultner, exploraron el enfoque PDSA para el desarrollo de software [Larman03].

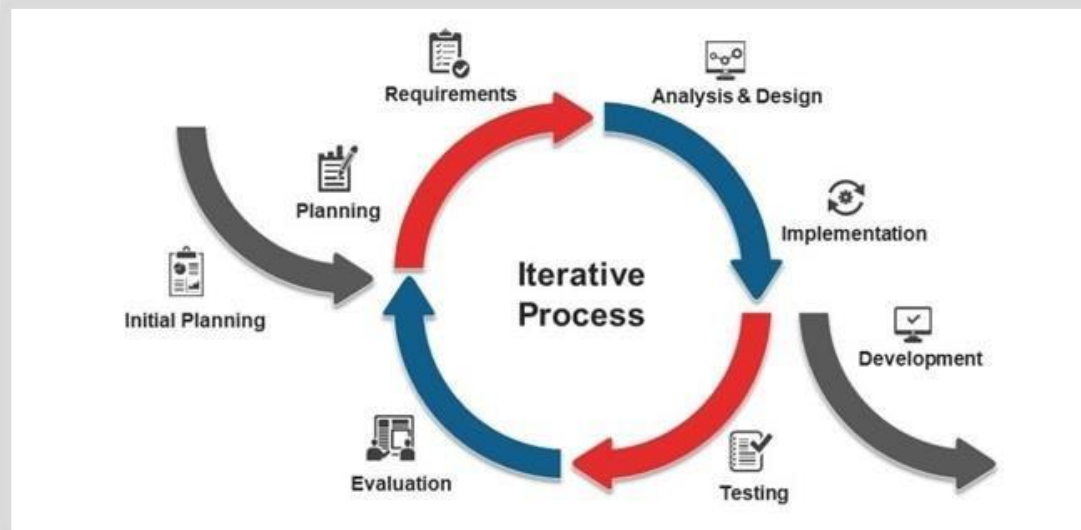


Figura 24 – Modelo Iterativo

Mills, quien trabajó en IBM desde 1964 hasta 1987, y donde alcanzó el estatus de investigador, se desempeñó en importantes cargos. En su vasta experiencia en proyectos de gran envergadura, aplicó el *Desarrollo Incremental* y la teoría estadística a las pruebas de software. En su conocida "Programación de arriba hacia abajo en grandes sistemas" publicado en 1970, promueve primero el desarrollo iterativo, pero no aclaró que se debía evitar una gran especificación inicial. Tampoco especificó el tamaño de la iteración. Luego, en 1976 pone el énfasis en el Desarrollo Incremental, con la participación continua del usuario y el rediseñando del software de acuerdo a los costos de cada incremento. En 1987 propone el método de desarrollo de software *Cleanroom* [Mills87] en uso en los laboratorios de IBM y de la NASA que proporciona una versión incremental de funciones de software o subsistemas (desarrollado a través de refinamiento gradual) para separar los equipos internos de garantía de calidad que aplican medidas estadísticas y análisis como base para certificar sistemas de software de alta calidad. Tanto Basili como Mills observaron que un problema importante de la construcción de software lineal es que no se valida con el cliente hasta ser puesto en producción. El modelo incremental actúa como un mecanismo de validación parcial del software a medida que se construye el software y permite detectar tempranamente errores en los requisitos. Por otro lado, [Glass69] menciona que el desarrollo incremental vale la pena, obliga a una exhaustiva revisión del sistema y evita el desaliento que produce la implementación y la administración. Desarrollar con un enfoque incremental permite al desarrollador aprovechar lo que se

aprendió durante el desarrollo de las versiones anteriores del sistema (ver Figura 25). El aprendizaje proviene tanto del desarrollo como del uso del sistema, siempre que sea posible. Los pasos clave en el proceso fueron: a) comenzar con una implementación simple de un subconjunto de los requisitos de software, b) cada subconjunto genera un entregable que va evolucionando hasta implementar el Sistema de Software completo. En cada incremento, se realizan modificaciones de diseño junto con la adición de nuevas capacidades funcionales.

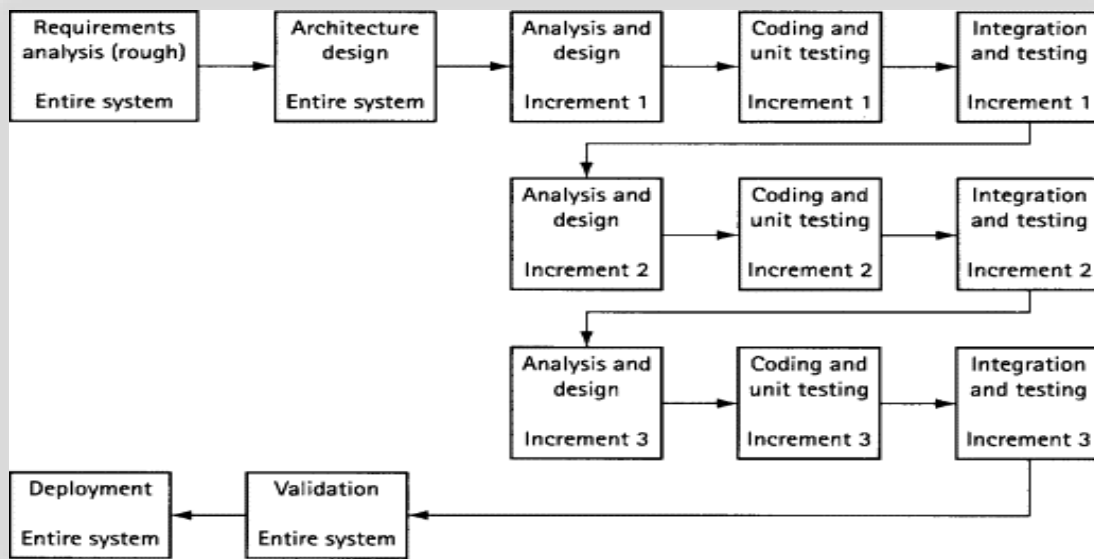


Figura 25 – Modelo Incremental

La primera publicación acerca del *Desarrollo Iterativo Incremental* (IID) fue en 1978, en la columna del UK's Computer Weekly donde escribía Tom Gilb. Pero, según Larman y Basili [Larman03], en la década del 50 la construcción del avión hipersónico X-15 utilizó un enfoque IID. Este proyecto, por su éxito, se convirtió en un hito de la década y, aunque no era de software, en 1960, algunos miembros del personal del proyecto Mercury de la NASA, retomaron la experiencia. Los resultados en cuanto a costos y satisfacción del cliente lo popularizaron en empresas como IBM, TRW donde trabaja Gilb, Mills, Royce, Boehm. El IID se convirtió en la mejor manera de construir software.

En 1982, Swartout y Balzer argumentó que la especificación y el diseño tienen una interacción necesaria, y promovieron un enfoque iterativo y evolutivo a la construcción de los requisitos. Esta metodología de desarrollo se la denominó *Prototipado* [Gomaa81] [Floyd84] y permite construir el Sistema de Software de una manera rápida y a un bajo

costo. El mismo software actúa como un elemento de comunicación entre el desarrollador y el cliente, permitiendo validar, de manera constante, los requisitos incorporados. Comienza con pocos requisitos, los más conocidos o esenciales, y los va completando de una forma evolutiva. De esta manera se asegura la comprensión de la relación entre las funciones del software y las tareas existentes.

El *Operational Specifications for Rapid Prototyping* [Balzer83] [Zave84] (ver Figura 26) supone la existencia de un lenguaje de especificación formal para soportar el desarrollo evolutivo de especificaciones en un prototipo de implementación. Las especificaciones en el lenguaje de especificación están codificadas, y cuando se evalúa computacionalmente, constituye un prototipo funcional del Sistema de Software. Cuando tales especificaciones se pueden desarrollar y procesar de forma incremental, el Sistema de Software resultante puede ser refinado y evolucionado en sistemas de software funcionalmente más completos. Sin embargo, los sistemas de software emergentes siempre están operativos de alguna forma durante su desarrollo.

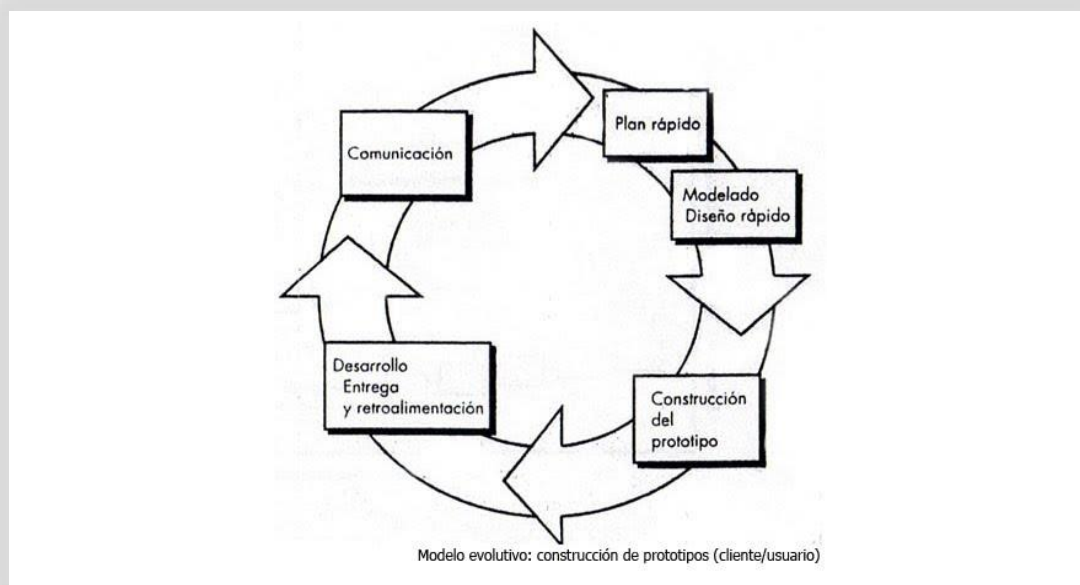


Figura 26 – Modelo Prototipado Evolutivo

Las variaciones dentro de este enfoque representan esfuerzos donde el prototipo es el fin buscado, o donde los prototipos especificados se mantienen operativos pero refinados en un sistema de software completo.

En 1988 se publica el *Modelo en Espiral* [Boehm88] (ver Figura 27) basado en la experiencia de 14 años en el uso del modelo en cascada. Boehm se basa en el modelo de

Royce, pero incorpora una fase de análisis de riesgo para evaluar los objetivos y restricciones definidas durante el planeamiento. Esto permite identificar áreas de incertidumbre que son potenciales riesgos para el proyecto. Luego, se debe evaluar el costo del riesgo construyendo un prototipo, realizando una simulación, o cualquier otra técnica de resolución de riesgo. Si el rendimiento o el riesgo de interfaz de usuario dominan fuertemente el desarrollo del programa o el riesgo de control interno de la interfaz, el siguiente paso puede ser un desarrollo evolutivo: un esfuerzo mínimo para especificar la naturaleza general de un producto, un plan para el siguiente nivel de prototipado y el desarrollo del prototipo más detallado para continuar resolviendo el principal problema de riesgo.

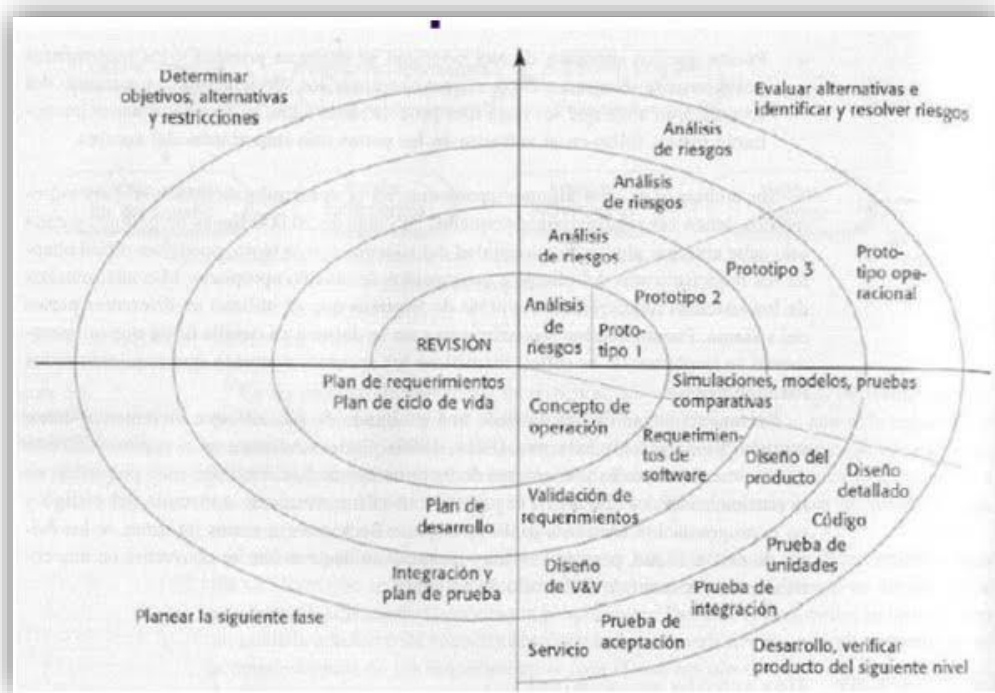


Figura 27 – Modelo en Espiral

El enfoque en general es por ciclos de desarrollo iterativos como una espiral en expansión, con ciclos internos que denotan análisis temprano del Sistema de Software y creación de prototipos, y ciclos externos que denotan el ciclo de vida clásico del software. La dimensión radial indica los costos de desarrollo acumulativos, y la dimensión angular indica progresos realizados en la realización de cada espiral de desarrollo. Boehm utiliza la propuesta de Royce en el cuadrante de desarrollo donde aplica un proceso secuencial,

pero además incorpora la estrategia incremental al permitir realizar el espiral tantas veces como sea necesario, de manera incremental hasta construir el software deseado.

La filosofía conocida como “compre, y no construya” divulgada por Brooks en 1987, fue aplicada en los 90 al *Desarrollo basado en Componentes* (CBD por Component Based Development) [Brown97] [Atkinson00] se define como un conjunto de tecnologías, herramientas y técnicas que permiten a las organizaciones de desarrollo pasar por todo el proceso de AD (análisis, diseño, construcción y montaje) o a través de cualquier etapa en particular mediante el uso de tecnologías predefinidas que permiten componentes (como patrones de AD, marcos, plantillas de diseño), herramientas y bloques de creación de aplicaciones. Propone una integración de componentes existentes para satisfacer los requisitos del software de una manera más segura y rápida, reduciendo el trabajo de construcción aproximadamente en un 70%.

En la década del 80 e inicios de los 90 existía una visión que la forma más adecuada de construir software era bajo un plan de proyecto riguroso, aseguramiento de la calidad, procesos de desarrollos rígidos y muy controlados. Esta idea se genera debido a que las grandes difusiones de la evolución en la construcción del software estuvieron centradas en la construcción de grandes sistemas de software de larga duración [Sommerville11]. La aplicación de esta forma de trabajar en sistemas de software pequeños y medianos se tornó difícil y costosa, donde hacer los cambios requeridos por un entorno dinámico y la presión del cliente en ir teniendo resultados, se volvió muy dificultoso. Se conocen las metodologías ágiles basadas en el desarrollo iterativo e incremental, la cual se centra en las personas que participan en el desarrollo más que en los procesos y herramientas. La mejora en los resultados obtenidos acerca de los defectos en el software, ha sido de aproximadamente un 25%. La propuesta de esta metodología se focaliza en:

- Individuos e interacciones sobre procesos y herramientas
- Entregas constantes sobre documentación
- Colaboración del cliente sobre negociación
- Respuesta al cambio sobre seguir un plan

En 2001 miembros de la comunidad adoptaron el nombre de métodos ágiles y poco después formaron la alianza ágil constituida por Extreme Programming (XP), Scrum, DSDM, Adaptive Software Development (ASD), Crystal, Feature-Driven Development (FDD), Pragmatic Programming, entre otros, para desarrollar el Manifiesto Ágil. Otros

métodos ágiles conocidos Agile Unified Process, Crystal Clear, Lean Software Development (LSD), Kanban, Open Unified Process (OpenUP), Método de desarrollo de sistemas dinámicos (D SDM), G300, 6D-BUM, PMI Agile, etc.

Las metodologías ágiles satisfacen las necesidades de los clientes mediante entregas continuas y la adaptación al cambio, pero este proceso está inmerso en la producción del software manteniendo la mirada en el producto a construir.

En la Figura 28 se pueden observar dos formas diferentes de construir software, una desde la perspectiva del Modelo Cascada y la otra desde las metodologías ágiles, específicamente desde Scrum.

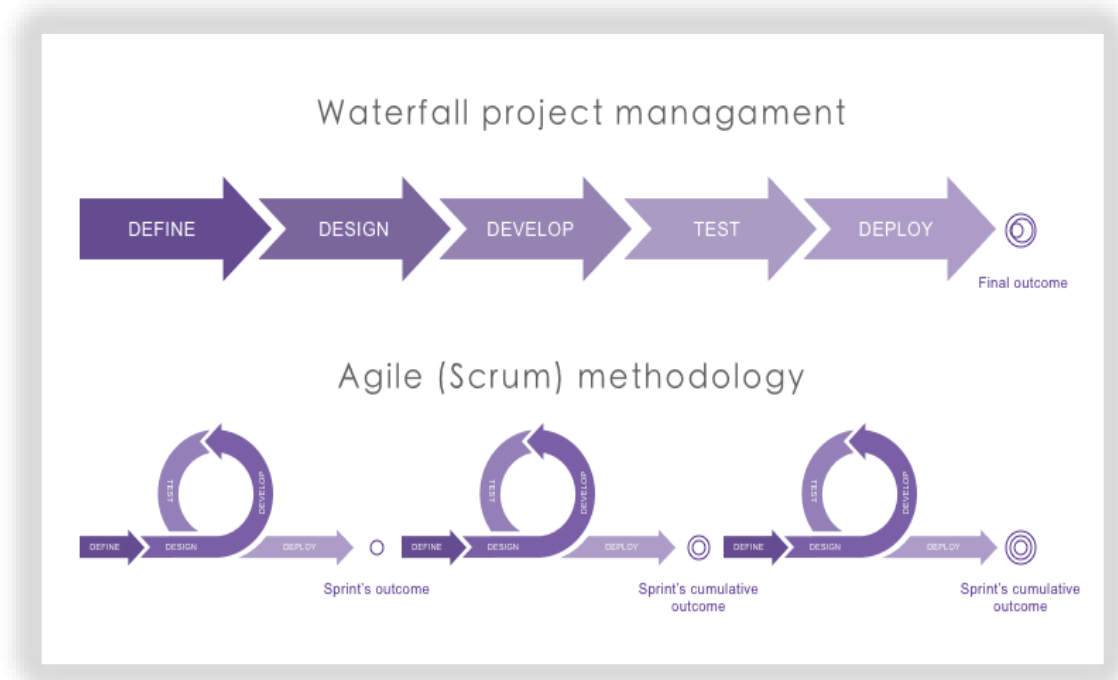


Figura 28 – Diferencia entre Cascada y Scrum

Para Tom Gilb [Gilb81] “Hay algo extraño en una cultura de software que está tan obsesionada con la novedad y el determinismo tecnológico que perpetuamente pierde el conocimiento del pasado en una búsqueda interminable de la reinención de la rueda. La carrera a partir del año 2000 para ‘agilizarnos’ y distanciarnos lo más posible de cualquier vieja perspectiva de la ‘Ingeniería de Software’, siempre con el fantasma del modelo Cascada ha resultado en la negligencia de una gran cantidad de conocimiento y sabiduría de los grandes ingenieros/as de sistemas de las décadas del 60 al 80. Es nuestra pérdida. Estamos olvidando cosas que son mejores de lo que creemos saber ahora.”. Esta opinión se sustenta en que Gilb es uno de los primeros y más activos promotores del IID. Su

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 5 - Reseña histórica de la Ingeniería de Requisitos

material fue probablemente el primero con un claro sabor de iteración ágil, ligera y adaptativa con resultados rápidos, similar al de los métodos IID más nuevos [RUP98] [Jacobson99]. Aunque algunos prefieren reservar la frase "desarrollo iterativo" simplemente para reelaborar, en los métodos ágiles modernos el término implica no sólo revisar el trabajo, sino también el avance evolutivo, un uso que data de 1968. IID representa mejoras y ampliaciones de funcionalidad en cada ciclo.

Parece ser que el modelo IID fue el que revolucionó la historia de la construcción del software. Se puede observar como impactó en la mayoría de las propuestas, inclusive hasta nuestros días. La estrategia Iterativa asegura el proceso, hacer las cosas hasta lograr la calidad deseada. El incremental asegura los requisitos del software al ir avanzando de a poco y revisando y adaptando los requisitos comprometidos. Esta estrategia permite incorporar cambios. En la última década (2010 a 2019), los cambios producidos han cambiado de eje. Ya no están tan centrados en los procesos, sino más bien en las personas, en la conformación de equipos de trabajo, la autogestión, la colaboración. Este es el caso de Spotify, Amazon, Google, Microsoft que ganaron en Wall Street el primer lugar al llevar a toda la organización las prácticas de desarrollo y transformarse en “organizaciones ágiles”.

Década	Algunos hitos que acompañaron	Publicaciones destacadas	De las organizaciones al mundo
1930		1930 IID con PDSA	
1950	Tarjetas perforadas Altos costos		1957 se utilizó el incremental IBM
1960	Codificar y corregir Ingeniería de Software		1968 se utilizó desarrollo iterativo en IBM
1970	Sistemas multiusuarios Sistemas en tiempo real Sist. De Base de datos Internet Programación estructurada	1970 Royce	1970 se utilizó desarrollo iterativo IBM
		1975 SDLC	
		1975 técnica de mejora iterativa	1976 se utilizó IID en IBM
		1978 IID	
1980	Sistemas Distribuidos Bajan los costos del hard Calidad del software	1980 Incremental	
		1983 Modelo de transformación formal	
		1984 Prototipado	
		1984 Operational Specifications for Rapid Prototyping	
		1984 Modelo operacional	
		1987 Cleanroom	
1988 Espiral			

		1990 Desarrollo basado en componentes	
1990	Sistemas expertos Redes neuronales Entorno cliente servidor WWW Ingeniería de Requisitos	1992 Modelo V	
		1998 RUP	
		2001 Agiles 2009 Agiles a gran escala	
2000	Sistemas potentes OO Reutilización		Organizaciones Agile (Spotify, Google, Microsoft, Amazon y otros)
2010	Omnipresencia de la web Automatización Se posicionan primeras en Wall Street Microsfot, Apple, Facebook y Amazon		

Tabla 7– Contexto y evolución de la construcción del software

Como se puede observar en la Tabla 7, esta evolución, acotada a algunos de los principales enfoques, encierra muchos aspectos relevantes de la construcción de software de los primeros 50 años, donde, los líderes del pensamiento de la Ingeniería de Software han sembrado los fundamentos para construir software con la mayor calidad y al menor costo posible. Ahora, comprendiendo la importancia de esta evolución, se retoma la pregunta inicial *¿La “crisis del software” se ha convertido en una enfermedad crónica?* Analizando el texto que precede, la Figura 21 y la Figura 29, la respuesta parece ser afirmativa, lo que conlleva a otro interrogante ¿por qué no se ha mejorado? En este sentido aparecen otros interrogantes:

1. ¿Será necesario profundizar aún más en el aprendizaje de los procesos, métodos, estrategias, paradigmas, etc. para construir software de calidad?
2. ¿Será que los nuevos métodos van por detrás de las necesidades reales de las organizaciones, y cuando un método es correctamente comprendido y aplicado esas necesidades cambiaron?
3. ¿Será que se deben personalizar aún más las metodologías a las particularidades de cada organización y a sus contextos y no aplicarlas como recetas?
4. ¿Será que las variables mencionadas en el **Capítulo 3-Figura 10** deben ser incorporadas a los procesos de construcción de software?
5. ¿Será que la naturaleza de la construcción de software sólo admite un promedio del 30% de éxito?

6. Finalmente, ¿será que la IR no se ha acoplado lo suficiente a los nuevos desafíos?

Naturalmente, no existe una única respuesta, que, además, excede la especificidad de la presente tesis, por lo tanto, sólo se deja sentada la reflexión para trabajos futuros, considerando que la propuesta de la presente tesis es un importante aporte en esta dirección.

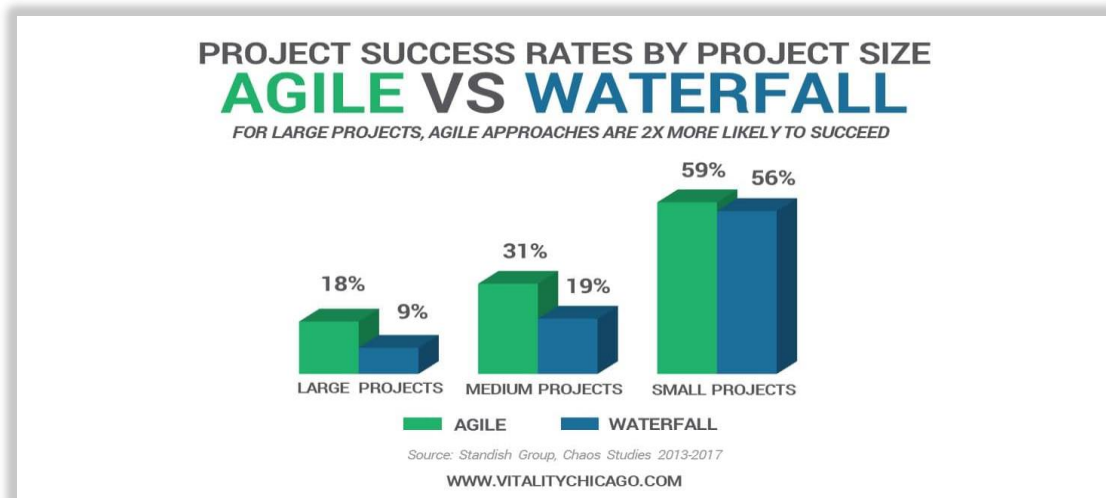


Figura 29 – Tasa de éxito por tamaño de proyectos (Cascada y Ágiles)

Este resumen acerca de la historia de la construcción del software, que por razones de espacio y centralidad de la presente tesis, no ha incluido todos los hitos importantes, encierra algunos aspectos relevantes de los primeros 50 años sin poder aclarar el cuestionamiento inicial de este Capítulo acerca de cómo se encuentra hoy la construcción del software ¿realmente se ha mejorado? Como se mencionó al comienzo de este Capítulo, han existido avances y retrocesos. Para responder correctamente, se deben tomar en consideración muchos aspectos que influyen en los resultados, como ser:

- contextos más complejos,
- usuarios más informatizados y, por lo tanto, más exigentes;
- sistemas de software con fuertes requerimientos de comunicación con otros sistemas informáticos,
- mercados cambiantes que obligan a las organizaciones a ser ágiles,
- entre otras.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 5 - Reseña histórica de la Ingeniería de Requisitos

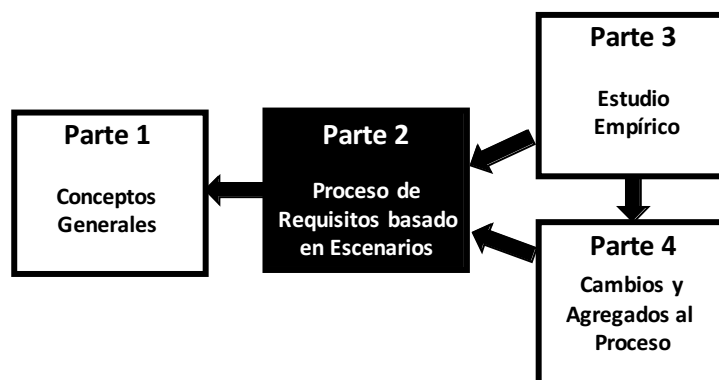
Es obvio que para responder acerca de la persistencia de la crisis del software, no alcanza con estadísticas aisladas, ni datos globales, se requiere evaluar múltiples variables que afectan a la construcción del software y en simultáneo. A pesar de esto, los datos que reflejan la Considerando que no disponemos de datos suficientes, pero considerando los resultados de la Figura 21, obligan a reflexionar acerca de las causas de tantos fracasos en la construcción de software. Entre muchas causas, se mencionan las que siguen:

1. Falta de **aprendizaje** acerca de cómo utilizar los procesos, métodos, estrategias, paradigmas, etc.,
2. Poca conciencia acerca de la **ética** profesional para la Ingeniería de Software,
3. Los **nuevos métodos** van por detrás de las necesidades reales de las organizaciones, y cuando un método es correctamente comprendido y aplicado esas necesidades cambiaron,
4. Las **variables** descritas en el **Capítulo 3 - Figura 10** deben ser tratadas en los procesos de construcción de software.
5. ¿Será que la naturaleza de la construcción de software sólo admite un promedio del 30% de éxito?

Por la complejidad de este análisis y porque excede la especificidad del presente trabajo, sólo se deja planteada la reflexión, deseando que la propuesta de la presente tesis sea una contribución para mejorar esta realidad.

Parte 2

Proceso de Requisitos basado en Escenarios



Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Introducción

En la presente Parte se describe el *Proceso de Requisitos basado en Escenarios* completo, tal como ha sido definido y publicado hasta el año 2010. El corte en este año se debe a que en esta fecha comienza el estudio empírico presentado en la Parte 3.

A continuación, en la Tabla 8, se describen las publicaciones realizadas desde 1997 a 2010 acerca de los modelos, las heurísticas de construcción, las diferentes verificaciones con inspecciones, etc. donde la tesista ha participado.

Año	Título	Autores	Lugar	Tipo	Cita
1997	Enhancing a Requirements Baseline with Scenarios	Leite J., Rossi G., Balaguer F., Maiorana V., Kaplan G., Hadad G., Oliveros A.	Requirements Engineering Journal	Revista Internacional	[Leite97]
1997	Construcción de Escenarios a partir del Léxico Extendido del Lenguaje	Hadad G., Kaplan G., Oliveros A., Leite J.	JAIIO 97		[Hadad97]
1998	Inspección de Escenarios	Doorn J., Kaplan G., Hadad G., Leite J.	WER 98	Congreso Internacional	[Doorn98]
1999	Integración de Escenarios con el Léxico Extendido del Lenguaje en la elicitación de requerimientos: aplicación a un caso real	Hadad G., Kaplan G., Oliveros A., Leite J.	Revista de Informática Teórica y Aplicada (RITA)	Revista Internacional	[Hadad99]
1999	Enfoque Middle-Out en la Construcción e Integración de Escenarios	Hadad G., Doorn J., Kaplan G., Leite J.	WER 99	Congreso Internacional	[Hadad99]
2000	A Scenario Construction Process	Leite J., Hadad G., Doorn J., Kaplan G.	Requirements Engineering Journal	Revista Internacional	[Leite00]
2000	Inspección del Léxico Extendido del Lenguaje	Kaplan G., Hadad G., Doorn J., Leite J.	WER 00	Congreso Internacional	[Kaplan00]
2002	Comprendiendo el Universo de Discurso Futuro	Doorn J., Hadad G., Kaplan G.	WER 02	Congreso Internacional	[Doorn02]
2003	Using Scenario Inspections on Different Scenarios Representations	Leite J., Doorn J., Hadad G., Kaplan G.	Monografias em Ciência da Computação, Departamento de Informática, PUC-Rio	Revista Internacional	[Leite03]
2003	Validación de Escenarios Futuros con prototipos	Kaplan G., Doorn H., Hadad G.	WICC 03	Congreso nacional	[Kaplan03]

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Introducción

2004	Defining System Context Using Scenarios	Leite J., Doorn H., Kaplan G. Hadad G., Ridao M.	Perspectives on Software Requirements	Capítulo de libro	[Leite04]
2005	Scenariu Inspection	Leite J., Doorn J., Hadad G., Kaplan G.	Requirements Engineering Journal	Revista internacional	[Leite05]
2006	Ingeniería de Requisitos	Kaplan G., Doorn J., Hadad G.	UTN-FRBA	Apunte de Clases	[Kaplan06]
2008	Una Visión Cognitiva de los Modelos de Requisitos	Kaplan G., Doorn J., Hadad G.	WICC 08	Congreso nacional	[Kaplan08]
2009	Explicitar Requisitos de Software usando Escenarios	Hadad G., Doorn J., Kaplan G.	WER 09	Congreso Internacional	[Hadad09a]
2009	Facilitando la Asignación de Prioridades a los Requisitos	Hadad G., Doorn J., Kaplan G.	WER 09	Congreso Internacional	[Hadad09b]
2009	Storyboard Basados en Escenarios Futuros	Kaplan G., Doorn J., Guatelli R., Gigante N., Hadad G.	WICC 09	Congreso nacional	[Kaplan09]
2010	Documentando Requisitos en el Contexto de su Elicitación	Hadad G., Doorn J., Kaplan G.	WICC 10	Congreso nacional	[Hadad10]

Tabla 8 – Publicaciones relacionadas con la Parte 2

Estas publicaciones son producto de diferentes proyectos de investigación. Nuevamente, en la Tabla 9 se describen estos proyectos hasta el año 2010 en los cuales la tesista ha participado.

Período	Institución	Nombre del Proyecto	Investigadores/as
01/01/1995 31/05/1997	UB	Uso de Escenarios en el Desarrollo de Software	Director: Julio Leite (Pontificia Universidade Católica, Río de Janeiro) Codirector: Jorge Doorn Codirector: Gustavo Rossi, Alejandro Oliveros. Investigadores/as: Gladys Kaplan y Graciela Hadad, Balaguer, Federico, Maiorana Vanesa
01/06/1998 31/12/2000	UB	Gerencia de Inconsistencias en Escenarios	Director: César Leite (Pontificia Universidade Católica, Río de Janeiro) Codirector: Jorge Doorn Investigadores/as: Gladys Kaplan y Graciela Hadad, Balaguer, Federico, Maiorana Vanesa
01/01/2001 31/12/2007	UTN - FRBA	Obtención de Requisitos mediante Escenarios.	Director: Jorge Doorn Investigadores/as: Gladys Kaplan y Graciela Hadad
01/01/2008 31/12/2009	UNLaM	Validación de Requisitos de Software.	Director: Jorge Doorn Investigadores/as: Gladys Kaplan y Graciela Hadad

Tabla 9 – Proyectos de Investigación relacionados con la Parte 2

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Introducción

Nota: en esta Parte se han incorporado cuadros en color azul, como el que contiene este texto, para describir algunos aprendizajes obtenidos durante el estudio empírico de la Parte 3, y en otros casos, se utilizan para relacionar los temas con los cambios y agregados de la Parte 4.

Capítulo 6

Proceso de Requisitos basado en Escenarios

Resumen

El *Proceso de Requisitos basado en Escenarios* que se describe a continuación es central en la presente tesis. Este proceso provee un escalonamiento cognitivo entre los modelos construidos, lo cual es una metáfora para describir que el pasaje entre modelos se produce a partir de la información existente en los modelos previamente construidos. De esta manera se asegura una buena calidad de la información para el nuevo modelo y se evita, lo que se conoce dentro del ámbito de los escritores, como el *síndrome de la hoja en blanco*⁸.

⁸ “El Síndrome de la hoja en blanco es el pánico a empezar algo, una parálisis psicológica inherente al principio del proceso creativo. Se llama así porque casi todos los procesos creativos comienzan con una hoja en blanco que se quiere llenar con las mejores ideas” (Wordpress.com).

Estrategia del Proceso de Requisitos

El *Proceso de Requisitos basado en Escenarios* tiene una estrategia de construcción secuencial, la cual se divide en tres etapas: *Comprender el UdeD actual*, *Planificar el UdeD futuro* y *Explicitar los Requisitos del Software* (ver Figura 30). Los verbos que refieren a cada etapa son autorreferenciales. Antes de comenzar con la primera etapa, se debe asegurar que el Objetivo General del Sistema se encuentre claramente definido, de no ser así, se debe trabajar junto al cliente hasta obtenerlo. Parviainen et. al [Parviainen05] afirman que no se puede modificar aquello que se desconoce. Haciendo un paralelo, el Proceso de Requisitos propone conocer el macrosistema en estudio antes de plantear los cambios necesarios en el proceso del negocio al incorporar el nuevo Sistema de Software. De esta manera, en la primera etapa ***Comprender el UdeD actual*** se elicitaba y modela el proceso del negocio tal como existe al momento de iniciar la IR. El conocimiento obtenido se utilizará para tomar las próximas decisiones acerca de los servicios del software. Por lo tanto, la próxima etapa es ***Planificar el UdeD futuro***, la cual consiste en definir junto a clientes y usuarios el proceso del negocio con el Sistema de Software incluido. La complejidad de esta etapa se debe a la necesidad de proyectar el futuro, y pensar en procesos del negocio que aún no existen. Finalmente, se deben ***Explicitar los requisitos***, los cuales se encuentran empotrados en un modelo de la etapa anterior (Escenarios Futuros). En este momento, sólo queda extraerlos de los Escenarios y generar el documento de especificación de requisitos. El formato de este documento dependerá de las políticas organizacionales, de los estándares nacionales o internacionales que se utilicen, etc., siendo uno de los más utilizados el estándar internacional IEEE 830-1998 [IEEE830].

Los procesos de requisitos en general y el que se describe en este Capítulo en particular, se encuadran en estrategias cuya ***elicitación es guiada por los modelos*** [Kaplan14b]. En este tipo de elicitación, el modelo a construir es lo que determina qué información se debe ir a buscar al UdeD. En el caso de un glosario, la atención está puesta en los aspectos léxicos; en un modelo de datos se concentra en las entidades o en los objetos del macrosistema; en los Escenarios en las situaciones del contexto, etc. Cabe destacar que la mayoría de los autores adhieren a esta forma de construir software [Yourdon75] [Booch91] [Jacobson92] [Loucopoulos95] [Leite04].

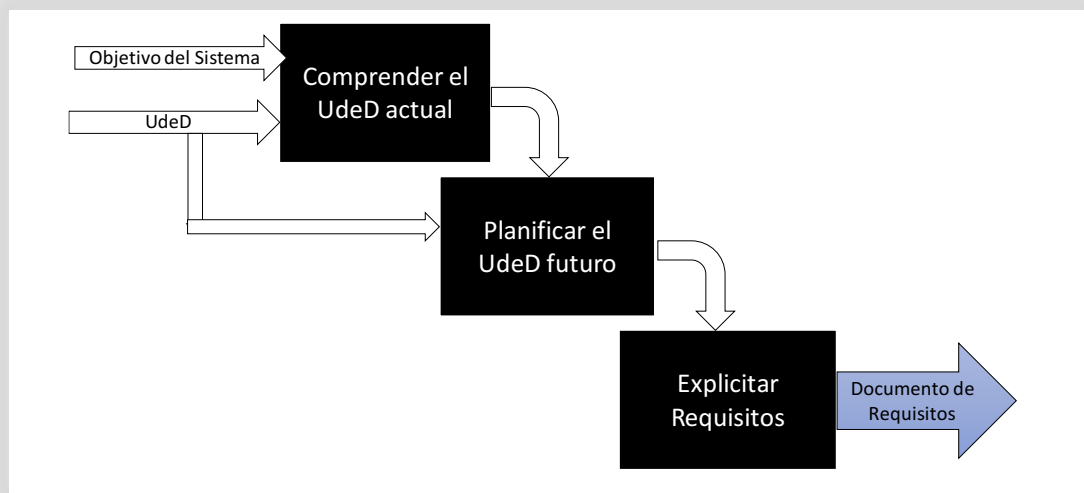


Figura 30- Estrategia del Proceso de Requisitos basado en Escenarios

La táctica de esta estrategia es básicamente construir dos modelos: el *Léxico Extendido del Lenguaje* (LEL) y los Escenarios. El LEL es un glosario cuyo objetivo es describir el léxico del macrosistema para mejorar la comunicación con el cliente y asegurar la comprensión de todos los artefactos producidos. La construcción de Escenarios propone concentrarse en situaciones particulares, las que se van agrupando en situaciones de mayor nivel constituyendo una *jerarquía de Escenarios*. En el tope de la jerarquía se encuentran los Escenarios Integradores que muestran el contexto global. Todos los Escenarios construidos deben maximizar el uso del léxico del macrosistema representado en el LEL, marcando una relación muy estrecha entre ambos modelos.

Los Escenarios pueden tener el punto de vista actual o del futuro. El primero, los Escenarios Actuales, representan el proceso del negocio existente, mientras que el otro, los Escenarios Futuros, representan el proceso del negocio futuro, con el nuevo Sistema de Software en ejecución. La principal diferencia entre los EA y los EF, es que los EF tienen empotrados los requisitos del software, aquellos que darán origen a la ERS.

En la Figura 31 se pueden analizar las principales actividades del Proceso de Requisitos en el marco de la estrategia presentada en la Figura 30. Puede observarse que antes de comenzar con la primera actividad, ya está definido el Objetivo General del Sistema y que, el mismo es transversal a todo el proceso.

La primera actividad que se realiza en el proceso es la **Construcción del LEL** con el objetivo de asegurar la comunicación oral y escrita durante todo el proceso. A partir del

LEL se deriva el primer conjunto de Escenarios, los cuales se completan retornando al UdeD durante la actividad **Construir los EA**.

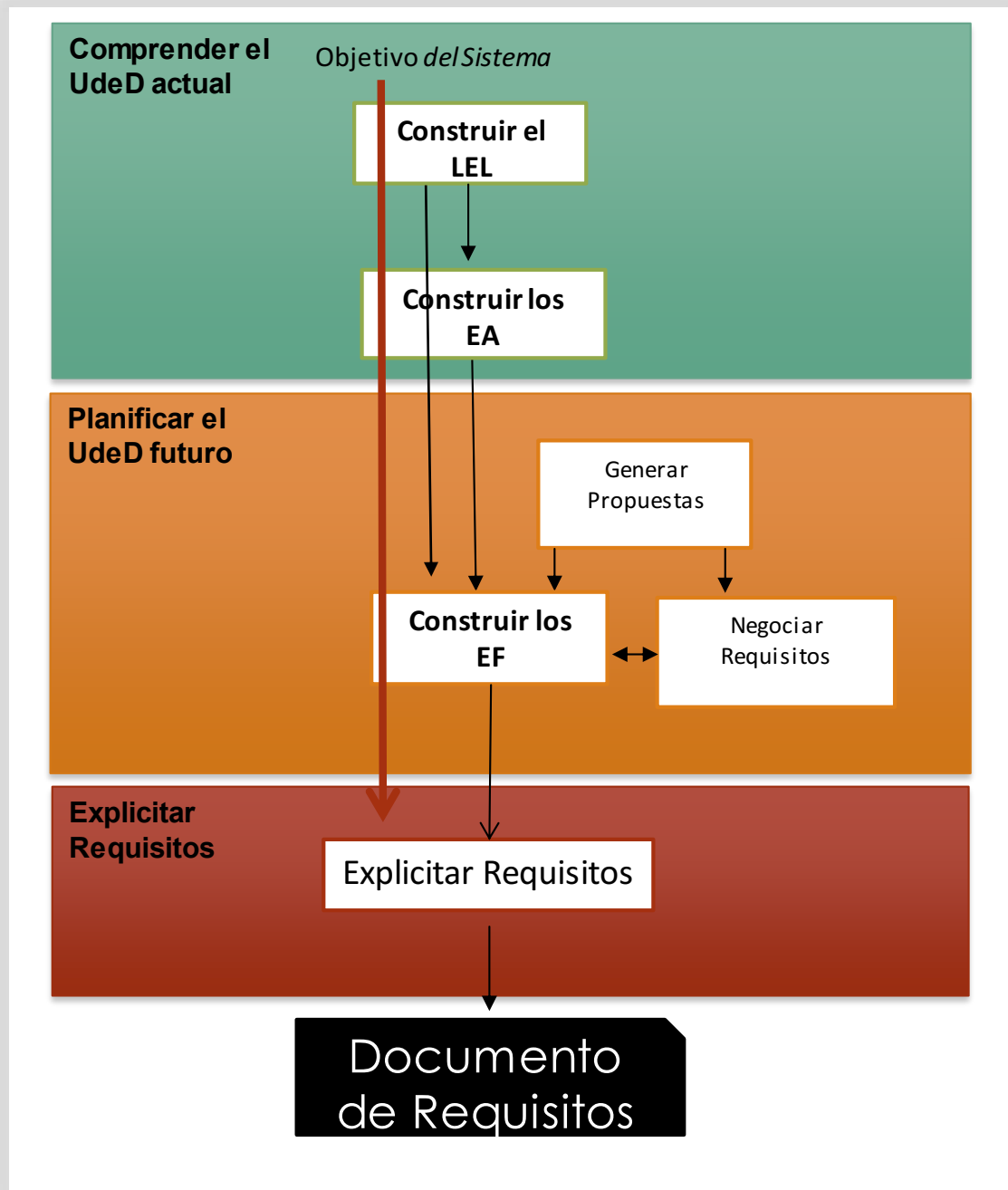


Figura 31 – Proceso de Requisitos basado en Escenarios

Con el conocimiento obtenido en esta etapa más las **propuestas de mejoras** se **Construyen los EF**, los que serán analizados y completados con el cliente-usuario, y donde, además, se **negocian** aquellos aspectos que presentan contradicciones, discrepancias, diferentes puntos de vista, etc. Puede observarse que los EF referencian al

LEL para reducir la ambigüedad de sus descripciones. Teniendo en cuenta que los EF tienen empotrados los requisitos del software, se pasa a la siguiente actividad **Explicitar Requisitos**, con el fin de describirlos en un documento. Estos requisitos están libres de conflictos ya que el mismo proceso los ha ido depurando. Los requisitos obtenidos desde los EF pertenecen tanto a la clasificación funcionales (RF) como no funcionales (RNF), aunque podrían aparecer algunos RNF por fuera del proceso, los que, de ser posible, deben ser incluidos en los Escenarios correspondientes.

El Proceso de Requisitos de la Figura 31 se ha modificado incorporando dos nuevas actividades, como puede observarse en la Figura 32. Estas actividades corresponden al **Tratamiento de la IE** (ver **Capítulo 18**) y a la **Construcción del LEL de Requisitos (LEL_R)** (ver **Capítulo 14**).

Para una mejor comprensión de los modelos utilizados en el Proceso de Requisitos y por cuestiones de organización del presente documento, se ha decidido describirlos en Capítulos separados. El LEL se describe en el **Capítulo 7**; las generalidades de los Escenarios en el **Capítulo 8**; los Escenarios Actuales en el **Capítulo 8.1** y los Escenarios Futuros en el **Capítulo 8.2**.

Todo el Proceso de Requisitos realiza la verificación de sus modelos utilizando la técnica de inspección con formularios o tablas. Se describe a continuación las generalidades de la técnica de inspección y las particularidades para cada modelo se describen en el Capítulo correspondiente.

Puede observarse en la Figura 32 que el *Tratamiento IE* comienza al construir el primer modelo (LEL) y se extiende hasta la actividad *Explicitar Requisitos*. En general, al llegar a los EF ya se ha incorporado toda la IE en el modelo correspondiente. Solo en casos especiales puede aparecer muy temprano en el proceso un requisito que deba ser tratado al explicitar los requisitos.

Verificación con formularios de Inspección

La actividad de Verificación, como ya mencionó, facilita la detección y corrección temprana de errores; reduce los riesgos de omitir aspectos relevantes del contexto; asegura la consistencia de cada modelo en particular y, por consiguiente, de todo el

proceso. Comprende un conjunto de tareas que aseguran que un modelo es correcto [Unhelkar05].

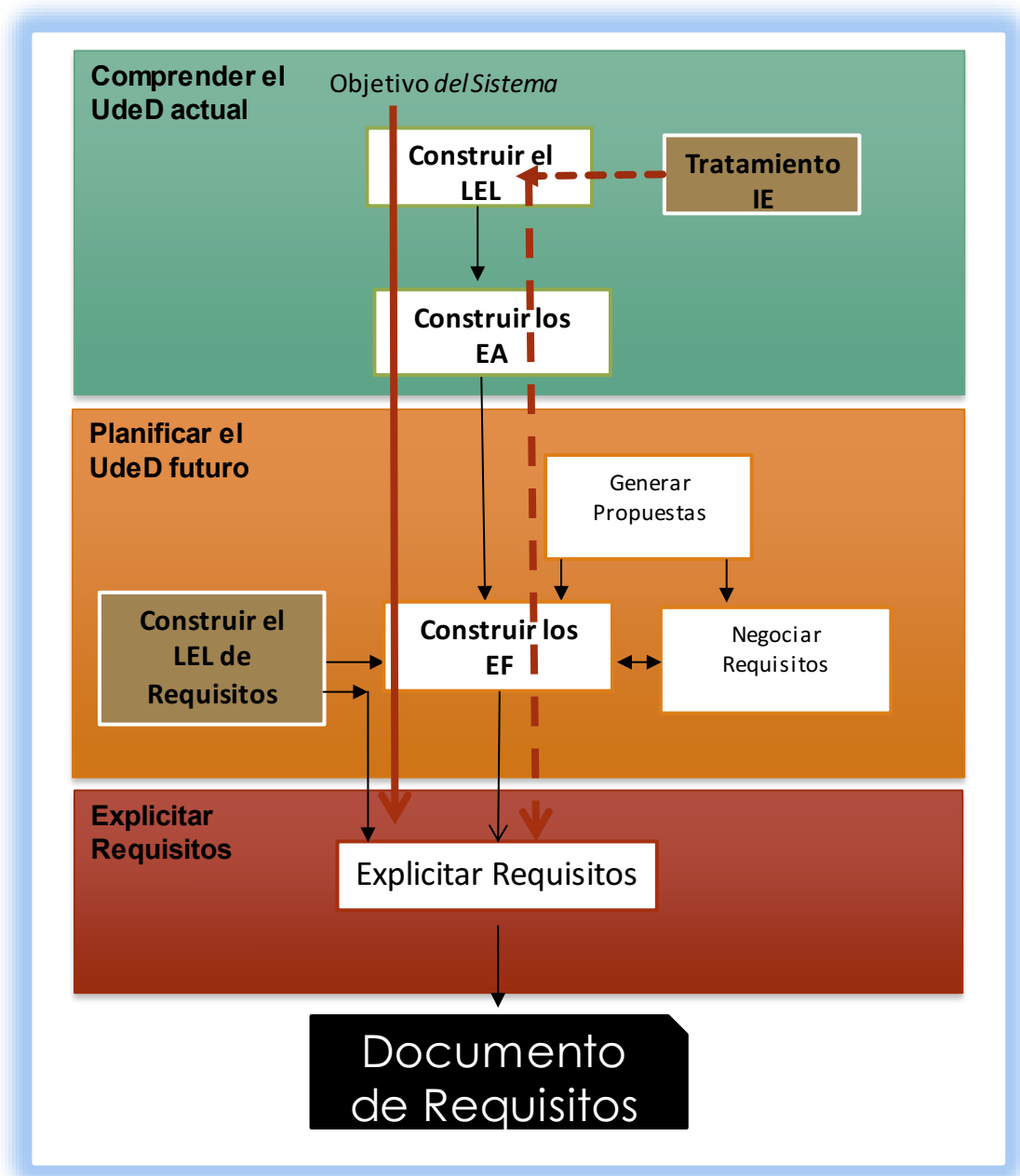


Figura 32 – Proceso de Requisitos basado en Escenarios actualizado

La entrada a la actividad es un artefacto y la salida un indicador de los defectos encontrados. En el caso del Proceso de Requisitos basado en Escenarios, como puede observarse en la Figura 33, la entrada es un modelo (LEL o Escenarios) y la salida es una lista de DEO (**D**iscrepancias, **E**rrores y **O**misiones), la cual es entregada a los autores del modelo una vez terminada la verificación, para que realicen las correcciones pertinentes. La lista DEO contiene todos los defectos encontrados y

algunas indicaciones para realizar las correcciones, ya que el inspector suele saber cómo corregir los defectos mencionados. En la presente tesis se utiliza como sinónimos los términos DEO y defectos.



Figura 33 – Actividad Verificar

A lo largo del tiempo, se han utilizado diferentes técnicas de verificación en la Ingeniería de Software [Boehm84] [Ackerman89] [Goedicke99] [Kazhamiakin04] [Drusinsky08] [Femmer14] [Takoshima15], siendo la técnica más utilizada la de inspecciones [Fagan74] [Fagan76]. A partir de la propuesta de Fagan, varios autores han propuesto utilizarla en la IR para verificar las ERS [Martin90] [Schneider92] [Porter95] [Gough95] o la han utilizado de manera diferente, como es el caso de la técnica N-fold [Martin90] [Schneider92]; el modelo probabilístico [Kantorowitz97]; etc. De la misma manera, en el Proceso de Requisitos basado en Escenarios se utiliza la técnica generando formularios compuestos de tablas, las cuales permiten detectar varios defectos simultáneamente [Kaplan00] [Leite05] (ver “Anexo A - Verificación de consistencias en el LEL y de los Escenarios”).

A modo resumen, Fagan propone realizar la inspección con las siguientes actividades: la *Planificación* es una apreciación global de los documentos, la *Preparación* es la verificación en sí misma, la *Reunión* se realiza entre los inspectores y los autores para analizar los defectos detectados, la *Corrección* es cuando los autores modifican el documento y finalmente el *Seguimiento* donde se controla que se hayan realizado todas las correcciones acordadas en la reunión y, de ser necesario, proponer nuevas inspecciones. Para la inspección del LEL y de los Escenarios, la fase de *Planeamiento* consiste en la selección del material a inspeccionar, la elección de los participantes, la identificación de los roles (inspector, moderador y escriba) y la preparación del material a inspeccionar: el LEL o Escenarios, los formularios de inspección y la guía de instrucciones. La *Preparación* es realizada por el inspector una vez que recibe el material y una copia del plan de inspección. La preparación consiste primero en la lectura cuidadosa de las instrucciones y luego en completar los formularios de

inspección, registrando toda DEO detectada. La *Reunión* apunta principalmente a confirmar o rechazar las DEO detectadas y secundariamente a descubrir nuevas DEO. En la reunión participan un moderador, un escriba, el inspector y los autores. Los autores convocados a la reunión realizan posteriormente las correcciones necesarias. Finalmente, en el *Seguimiento* se controlan dichas correcciones.

La actividad de Verificación puede repetirse dependiendo de dos factores: la cantidad y tipo de defectos encontrados. El peso de cada DEO se mide según su importancia en la consistencia del modelo, siendo una clasificación clásica utilizar la magnitud del defecto organizándolos en menor, mediana o mucha importancia. Por ejemplo, en el LEL, un defecto de menor importancia significa que su corrección no afecta a otros símbolos, como ser un símbolo que no está correctamente identificado. Uno de mediana importancia podría ser un símbolo sin impacto. Y uno importante es la existencia de homónimos no identificados donde se afecta la interpretación del mismo. Por lo tanto, la relación cuantitativa y cualitativa de las DEO es lo que determina la necesidad de volver a realizar una inspección, ya que muchos defectos de baja importancia quizás no requieran de una nueva inspección, mientras que pocos de mucha importancia, seguramente involucren a otros símbolos, hace necesario volver a verificar. Otro aspecto a considerar, son los cambios o actualizaciones realizados a un modelo, ya que éstos suponen la posibilidad de haber incorporado defectos y, por lo tanto, se requiere de una nueva verificación.

Capítulo 7

Léxico Extendido del Lenguaje

Resumen

Conocer el léxico utilizado en el contexto mejora la comunicación de todos los participantes en la construcción del software. Por tal motivo, el Proceso de Requisitos basado en Escenarios propone, como primera actividad, construir el Léxico Extendido del Lenguaje (LEL) que es un glosario que define la denotación y la connotación de cada término utilizado en el macrosistema. A continuación, se describe el proceso de construcción del LEL.

Glosarios en la construcción de software

Uno de los factores que afectan la calidad del software es la dificultad de establecer una comunicación segura entre los desarrolladores y los involucrados. Una comunicación fallida puede provocar distorsiones, malos entendidos, contradicciones y omisiones no deseadas. El uso de un glosario en la construcción de software tiene como propósito principal reducir la ambigüedad y mejorar la precisión, tanto en la comunicación escrita como oral. Diferentes autores han estudiado el impacto que tiene el vocabulario utilizado durante la construcción del software. Zave et al. [Zave97] afirma que “toda la terminología usada en Ingeniería de Requisitos deberá estar fundada en la realidad del ambiente para el cual se construirá una máquina”, además, sostiene que cada término debe estar descrito para asegurar su uso correcto. Weidenhaupt et al. [Weidenhaupt98] muestra un estudio donde se analizaron quince proyectos industriales, de los cuales cuatro usaron glosarios. El objetivo de estos glosarios fue procurar una mejor comunicación entre los involucrados y el equipo de desarrollo. Ben Achour et al. [Ben Achour99] determinó que la mitad de los casos de uso de un proyecto tenían errores basados en la falta de comprensión de la terminología utilizada. El estándar de la IEEE [IEEE830] para la Especificación de Requisitos de Software sugiere incluir un glosario con los términos utilizados en el documento, con el objetivo de evitar la ambigüedad y asegurar la consistencia interna del documento [Jacobson99].

En el marco del presente Capítulo, Leite et al. [Leite89] [Leite90], como ya se mencionó en el Capítulo anterior, propone el uso del Léxico Extendido del Lenguaje (LEL), cuyo objetivo principal es mejorar la comunicación entre los involucrados y los desarrolladores, homogeneizar las descripciones e identificar conflictos en el léxico utilizado por los usuarios, utilizando una representación de hipertexto. La construcción del LEL es un proceso cognitivo que permite registrar el léxico utilizado en el macrosistema, algo similar ocurre en algunos usos de las ontologías [Vasilecas06] [Dietz06] [Calero06]. En un glosario es frecuente encontrar términos utilizados en contextos reducidos con significados o usos notoriamente diferentes y a veces contradictorios con el uso corriente de los mismos. Es así que una persona altamente calificada puede no conocer el lenguaje de un contexto cerrado o interpretarlo diferente. Para describir este problema, se utiliza un pequeño pasaje del libro *Martin Fierro* de José Hernández:

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 7 – Léxico Extendido del Lenguaje

"Yo llevé un moro de número
sobresaliente el matucho
con el gané en Ayacucho
más plata que agua bendita
siempre un gaucho necesita
un pingo pa fiarle un pucho".

Puede observarse la necesidad de explicar el significado de ciertas palabras como moro, número, matucho, pingo, fiarle y pucho en este contexto. Un caballo *moro* es aquel cuya crin, la cola y la cabeza son más oscuras que el resto de la capa, y las cuatro patas son negras. En esta exquisita narrativa, en un sólo término se aglomera una gran cantidad de información, la cual se pierde si se desconoce el léxico particular utilizado. Continuando con el ejemplo del léxico gauchesco, y a efectos de mostrar la complejidad del mismo, en el siguiente texto⁹ se puede observar una descripción generalizada de cómo llamar a los caballos según su pelaje:

“Aun sin mencionar las razas, los nombres genéricos de los yeguarizos en la pampa gaucha son numerosísimos. Según el pelaje, los hay alazanes, bayos, blancos, cebrunos, colorados, doradillos, gateados, lobunos, moros, oscuros, overos, picazos, rosillos, tordillos y zainos, con variedades y combinaciones cuyos designadores son muchos y tan descriptivos como plateado, mosqueado, albino, blanco huevo de pato, sucio, azulejo, naranjado, encerado, atigrado, ruano, pangaré, barcino, requemado, malacara, pardo, tapado, zafrado, tobiano y sabino.

Estos últimos nombres obedecen a detalles existentes en cualquier parte del cuerpo, según explica Tito Saubidet en su irremplazable "Vocabulario y refranero criollo", y aún continúa la lista con otros del estilo de entrepelado, salpicado, aporotado, lunarejo, tiznado, chorreado, raya de mula, raya cruzada, yaguané, fajado, lagarto, bragado, marucha mora y muchos más. De acuerdo con otras características de sus miembros se los distingue como calzado si tiene pelo blanco en uno o varios remos -con variantes como argel, maneado, media res o bota con delantal-; mano mora -o pata mora- si posee este pelaje en sus miembros; cabos negros; cebrado... Y éstos son sólo ejemplos. También existe un profuso nomenclador del caballo según las peculiaridades de su cabeza: boca de mula, corazón, estrella, gargantilla, lucero, malacara, pampa, quitilipe, siestero, zarco, etcétera.

Distintos calificativos que suelen aplicarse a caballos y yeguas son: amadrinado, cuando sigue fielmente a la yegua madrina; bagual, cuando es arisco, aunque ya esté domado; estrellero, cuando marcha con la cabeza muy levantada; galán, cuando es armonioso por su forma y pelaje ... y hay más. A veces el criollo califica con los mismos adjetivos a sus congéneres y a sus caballos, como ocurre con curcuncho por jorobado; currutaco por chico,

⁹ “Rincón gaucho. El arte de nombrar al caballo”, Diario La Nación, 7 de febrero de 1998

malcriado o cursiento, muy usado en mi infancia despectivamente, como quien hoy dice mocososo.”

Se puede observar en ambos ejemplos que la definición de “moro” está contextualizada y utilizada de manera particular. Si bien el término tiene la misma denotación, la utilización en el primer caso propende a describir los detalles del caballo, mientras que, en el segundo, sólo se conoce que es parte de una clasificación general. Esta necesidad de conocer más acerca de cada léxico utilizado para comprender su significado, no le pertenece sólo al contexto gauchesco, sino que es de todos los léxicos. Esta realidad no es ajena a la IR, muy por el contrario, es muy importante y enciende una alerta sobre la naturaleza de muchos errores en los requisitos del software, los que son originados en el desconocimiento del vocabulario utilizado.

Léxico Extendido del Lenguaje (LEL)

El LEL es un glosario que permite comprender el léxico utilizado en el contexto. Contiene las palabras y frases que utilizan los usuarios y que son relevantes en el contexto. También se incluyen aquellas cuyo significado se desconoce. Cada palabra o frase se la denomina “símbolo” o “término” del LEL. Describir el vocabulario del dominio, permite detectar particularidades que en algunas oportunidades son desconocidas aún por los propios usuarios. Este puede ser el caso de algunos sinónimos y homónimos utilizados en distintas áreas de la organización.

Es importante resaltar que durante esta actividad el ingeniero/a de requisitos se debe centrar en conocer el vocabulario del contexto, dejando para una etapa posterior la comprensión del proceso del negocio y sus necesidades.

En la Figura 34 se presenta el meta modelo LEL, donde cada símbolo se identifica con un nombre (o más de uno en caso de sinónimos) y tiene dos componentes, uno llamado **Noción** que describe la denotación del símbolo, y otro, denominado **Impacto** que describe la connotación del mismo. Al describir los símbolos, dos principios deben cumplirse: el *principio de circularidad* que postula la maximización del uso de símbolos en la descripción de otros símbolos y el *principio de vocabulario mínimo* que postula la minimización del uso de términos que son externos al léxico. Estos términos externos deben pertenecer a un pequeño subconjunto de un diccionario en lenguaje natural. Estas reglas enfatizan la descripción del vocabulario como un hipertexto auto contenido y altamente conectado.

LEL: representación de símbolos del léxico del dominio.

Sintaxis:

{Símbolo}₁^N

Símbolo: entrada del léxico que tiene un significado especial en el dominio de la aplicación

Sintaxis:

{Nombre}₁^N + {Noción}₁^N + {Impacto}₁^N

Nombre: identificación del símbolo. Más de uno indica la presencia de sinónimos.

Sintaxis:

Palabra | Frase

Noción: denotación del símbolo. Debe ser expresada usando referencias a otros símbolos y usando el vocabulario mínimo.

Sintaxis:

Oración

Impacto: connotación del símbolo. Debe ser expresado usando referencias a otros símbolos y usando el vocabulario mínimo.

Sintaxis:

Oración

donde Oración está compuesta solamente por Símbolos y No Símbolos, éstos últimos pertenecientes al vocabulario mínimo.

+ significa composición, {x} significa cero o más ocurrencias de x, | representa **or**

Figura 34 – Meta modelo LEL

Para darle homogeneidad al glosario y con el objetivo de equilibrar el nivel de detalle de las descripciones, los símbolos se clasifican en una taxonomía propia: **sujeto, objeto, verbo y estado**.

Esta clasificación agrupa los símbolos y permite definir un patrón sintáctico que ayuda al ingeniero/a de requisitos a saber qué información debe contener la noción y el impacto y así, homogenizar las descripciones. Esta taxonomía puede ser modificada para representar particularidades del contexto, en cuyo caso, se deben modificar los patrones sintácticos correspondientes. La modificación a la clasificación general ha sido para incorporar alguna especialización, con el objetivo de resaltar alguna característica del contexto y homogeneizar las descripciones. Por ejemplo, incorporar la especialización “Formularios” dentro de la taxonomía objetos. De esa manera se pueden describir las particularidades de los formularios de forma uniforme para todos los símbolos. Aun así,

se considera que la clasificación general es suficiente y que la posibilidad de necesitar modificarla es casi nula.

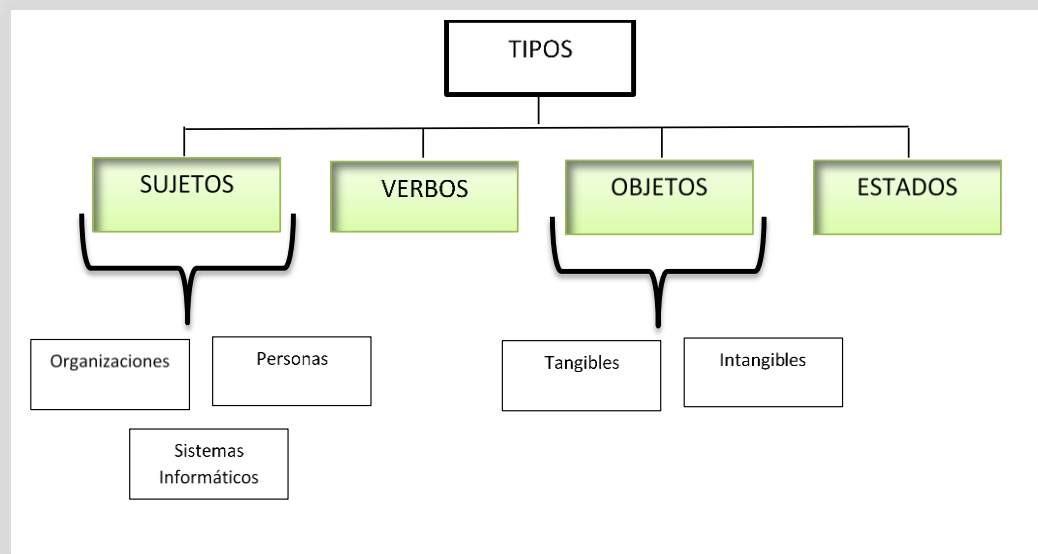


Figura 35– Jerarquía intrínseca del LEL

Como puede observarse en la Figura 35 , existe una jerarquía intrínseca entre los Tipos y las especializaciones sujetos, verbos, objetos y estados. Los sujetos agrupan a las *organizaciones*, las *personas* y los *sistemas informáticos*. En ocasiones los lugares donde se realizan las tareas son utilizados como sujetos, como puede ser el caso de una “caja” donde nunca se hace referencia al cajero sino se utiliza el lugar físico para corresponderlo, por ejemplo, “La caja registra todos los pagos”. Los objetos se componen de *tangibles*, que son las cosas que se manipulan para realizar las tareas, e *intangibles* que es todo aquello conceptual, pero que son necesarios para la realización o consistencia de los procesos (número de legajo, código de cargo, etc.). Los verbos se refieren a las actividades que realizan los sujetos y los estados representan a qué cambios pueden estar sometidos los otros símbolos.

Proceso de Construcción del LEL

La construcción del glosario tiene dos ejes centrales: ceñirse por el Objetivo General del Sistema y el UdeD. En función a esto y considerando que es una elicitación guiada por lo modelos, se debe comenzar identificando dónde se encuentra la información necesaria (personas, sistemas informáticos, manuales de procedimiento, estatutos, etc.). Esta tarea puede resultar muy simple o muy compleja dependiendo de cada contexto. Existe la

creencia acerca de que cualquier FI contiene toda la información que se busca, pero en realidad se complementan. Cuando se trata de documentos, se debe asegurar que se encuentran vigentes al momento de realizar el proceso. En este contexto, para comenzar el proceso (ver Figura 36), se deben **Identificar las fuentes de información**. Esta actividad es muy dependiente del contexto en estudio, pero en general se debe solicitar el organigrama, mapas estratégicos, manuales del macrosistema, normas nacionales e internacionales utilizadas, manuales de usuario de los sistemas informáticos en uso, etc., o sea, todo aquello que pueda proveer información.

Con las FI seleccionadas se deciden las mejores técnicas de elicitación [Goguen93] que nuevamente, dependerán de las particularidades del macrosistema. Entre las técnicas más conocidas se encuentran las entrevistas, observaciones, lectura de documentos, análisis de protocolos, discusión, etc. A pesar de la diversidad de técnicas disponibles, la más utilizada en la IR es la entrevista [Antonelli02]. Esta técnica es la que mayor riqueza de información provee, pero también, es muy compleja porque debe contemplar aspectos psicológicos, sociales, etc. Su implementación también depende de la complejidad del contexto como ser la cantidad y disponibilidad de los involucrados, el grado de pericia del usuario entrevistado, la experiencia del entrevistador, entre otros factores.

El presente Proceso de Requisitos utiliza las entrevistas como principal técnica de elicitación. Propone iniciar el proceso de construcción con entrevistas no estructuradas con el cliente. El objetivo es que el cliente transmita todo el conocimiento general del contexto en estudio que considere pertinente y relevante. Las siguientes entrevistas ya son estructuradas, donde el ingeniero/a de requisitos genera guías, cuestionarios, listas de dudas, con el propósito de completar la información faltante de cada símbolo.

En las primeras entrevistas es donde se **identifican los símbolos**, a medida que el cliente se expresa se genera la *lista de símbolos candidatos* que contiene las palabras o frases repetidas o enfatizadas por el cliente. Con esta lista comienzan a describirse los símbolos y a medida que esto sucede, se va refinando la lista candidata hasta convertirse en la lista de símbolos.

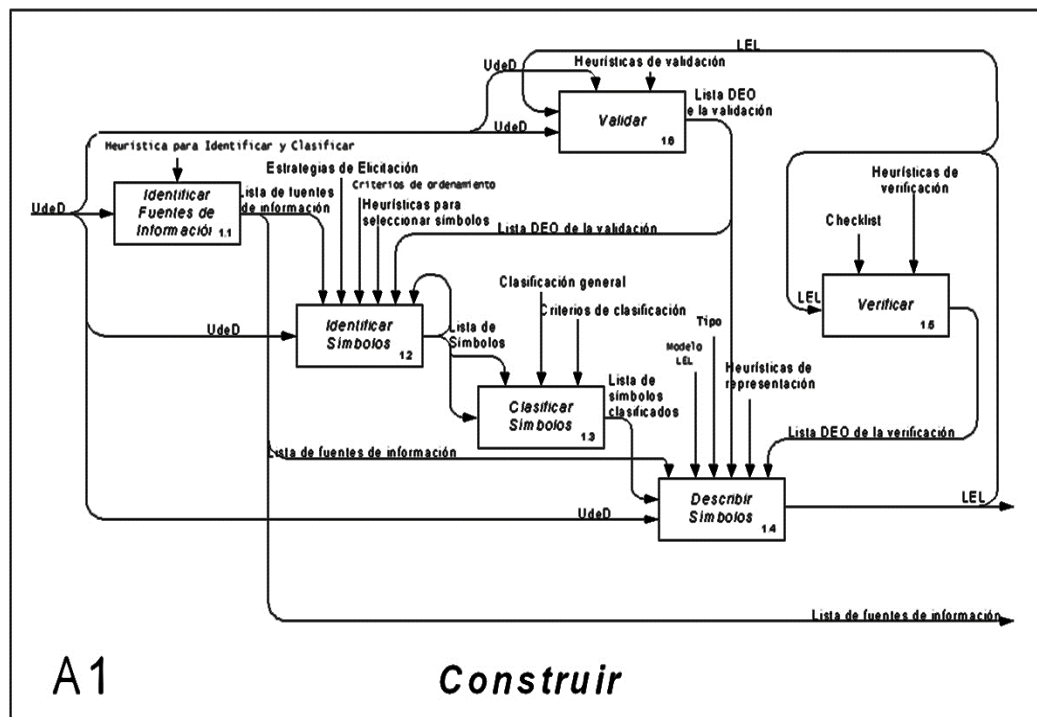


Figura 36 – SADT del Proceso de Construcción del LEL

Se ha detectado un problema con la lista inicial, el cual es desarrollado en el **Capítulo 10**. La solución a este problema es una nueva heurística de construcción del LEL por Proximidad que se describe en el **Capítulo 14**.

Durante la identificación de símbolos aparecen varias dudas acerca de qué símbolos incluir y cuales descartar. A continuación, se describen algunos aspectos importantes a tener en cuenta:

- El LEL debe contener exclusivamente palabras o frases pertenecientes al UdeD.
- Se deben excluir las palabras o frases que son de conocimiento público y que no tienen un particular uso en el UdeD.
- Se deben excluir las palabras o frases ambiguas, que no pueden ser resueltas en el UdeD.
- Se deben excluir palabras o frases que no tienen un comportamiento particular en el UdeD.

Al crear la lista candidata de símbolos se tiene la información necesaria acerca de cómo es utilizado en el contexto. Ese es el momento preciso para *Clasificar los símbolos*. La

clasificación de cada símbolo se debe asignar exclusivamente en función a cómo es utilizado en el contexto. Como ya se mencionó, la clasificación general se divide en sujeto, objeto, verbo y estado (ver Figura 35) . Se debe asignar lo más temprano posible, ya que es cuando se comprende cómo es tratado en el contexto y de esta manera se pueden identificar los homónimos. Una vez clasificados, se identifican los patrones sintácticos correspondientes. Estos patrones sintácticos se describen en la Tabla 10, y son utilizados por todos los ingenieros/as de requisitos que participan en la construcción del LEL dejando un glosario homogéneo de fácil lectura.

	NOCIÓN	IMPACTO
SUJETO	dejar claro quién es el sujeto.	registrar las responsabilidades y actividades que realiza o recibe
OBJETO	definir el objeto e identificar otros objetos con el cual tiene relación.	describir las actividades / acciones que pueden ser aplicadas al objeto.
VERBO	indicar quien la ejecuta, cuando sucede y la información necesaria para la ejecución de la acción	registrar las operaciones / procedimientos envueltos en la acción e identificar las situaciones que impiden que la misma se realice, qué otras acciones desencadenan en el ambiente y qué situaciones son causadas por la acción. Indicar que estados se desencadenan a partir de esta acción.
ESTADO	dejar claro qué significa y qué actividades y acciones llevaron a este estado	identificar otros estados y actividades/acciones que pueden ocurrir a partir de este estado

Tabla 10 – Patrones sintácticos del LEL [Kaplan00]

Se han propuesto modificaciones a los **patrones sintácticos** para mejorar la completitud del LEL, los cuales se describen en la **Capítulo 12**. También, se ha propuesto una mejora particular para el tratamiento de los Estados.

Con la lista de símbolos clasificados, se comienza a **Describir los símbolos**. Esto implica completar el componente Noción y el Impacto. Cabe destacar que, a esta altura del proceso, sólo se cuenta con la información de las primeras entrevistas. Se deben completar aquellos símbolos para los que hay información y retornar al UdeD para elicitar la información faltante. En la Figura 37 se muestra un ejemplo¹⁰ de cada tipo. El entorno de hipertexto se presenta entre los símbolos identificados en azul y subrayado (la marcación

¹⁰ Los casos utilizados en los ejemplos se describen en la Introducción de la Parte 3.

de un término depende de la herramienta utilizada), lo que hace posible navegar por esos símbolos. Finalmente, se puede observar en el nombre de algunos símbolos la presencia de sinónimos, los cuales se encuentran separados por una barra (“/”).

JEFE DE CONTROL DE CALIDAD (sujeto)

Noción

- Es la persona responsable de garantizar la calidad de los [insumos\(1\)](#)
- Trabaja en la [oficina técnica](#)
- Responde a la autoridad del [Director técnico](#).

Impacto

- [Autoriza el estado de los insumos\(1\)](#)
- [Actualizada las técnicas](#)
- [Actualiza el cronograma de trabajo](#)
- Determina cuando una [devolución](#) requiere [control de calidad](#)
- [Completa el cuaderno de asignación de números de análisis](#)
- [Verifica los análisis](#)
- Aprueba cómo se va a tomar una [muestra](#)
- Puede supervisar que el tercerista valide los equipos
- [Determina los apto de proveedores](#)
- [Determina los aptos de fabricantes conjuntamente con el jefe de producción](#)
- Verifica que se realicen los controles ambientales en los depósitos
- [Solicita y controla las validaciones de las técnica](#)
- [Envía la solicitud de pedido de material a administración](#)
- Controla la documentación que se le envía y recibe del tercerista
- [Asigna las fechas de análisis](#)
- [Determina la prioridad de análisis.](#)

CRONOGRAMA DE TRABAJO / ORGANIGRAMA DE TRABAJO (objeto)

Noción

- Es un calendario de outlook
- Contiene información sobre los futuros [análisis](#).

Impacto

- Se le cargan los [insumos\(1\)](#) sin [fechas de análisis](#)
- Se determinan las [fechas de vencimiento de los análisis](#)
- Se lo utiliza para [planificar el trabajo del área técnica.](#)

DEVOLVER (verbo)

Noción

- Lo realiza una [persona de producción](#) cuando existe un ingreso de [productos terminados](#) proveniente de ventas, como ser una mercadería en consignación, o una mercadería que no es pagada por el cliente y la devuelve, un [producto vencido](#) o un [recall](#).

Impacto

- Selecciona una [norma](#)
- Registra el motivo de la devolución en la [planilla de devolución](#)
- Puede generar la necesidad de hacer un nuevo [análisis](#)
- Puede solicitar una nueva [muestra](#)
- Puede solicitar la [destrucción de los productos](#)
- Los [insumos\(1\)](#) se almacenen en el [depósito de devoluciones.](#)

ASIGNADO (estado)

Noción

- Indica que un [insumo\(1\) pendiente](#) tiene asignada una [fecha de análisis](#) y un [técnico](#).

Impacto

- El [insumo\(1\)](#) se [analiza](#) a la brevedad
- Puede pasar al estado [aprobado](#) o [rechazado](#).

Figura 37- Ejemplos de símbolos del LEL (Caso SCC)¹¹

En el **Capítulo 11** se ha propuesto la generación de una *Vista Clasificación* que proporciona información resumida de cada símbolo del LEL para facilitar la adhesión al Principio de Circularidad.

Es necesario prestar una atención especial a los estados del LEL ya que estos deben aparecer como tal en el propio contexto de la aplicación. No siempre aparecen de manera espontánea. Es importante identificarlos, pero además se debe conocer con qué otros estados se relacionan y qué los cambios producen en el UdeD. Para ser parte del LEL, los estados deben aplicar a un símbolo del LEL, de cualquier tipo. Ahora, el estado pudo haberse generado adentro del contexto o afuera. Cuando el símbolo al que aplica un estado desaparece del LEL, también lo hacen sus estados. Por ejemplo, un estado que aplica a un objeto y dicho objeto fue excluido del LEL porque no pertenecer al contexto, también lo hacen sus estados. Por lo tanto, se puede decir que los estados son dependientes de su símbolo origen.

A continuación, se describen algunas reglas para describir símbolos:

- Cada noción y cada impacto debe ser descripto como una sentencia simple y directa.
- Los símbolos que comparten su significado, llamados sinónimos, son una única entrada del LEL.
- Las descripciones de las nociones y los impactos de cada símbolo se deben ajustar a su patrón sintáctico.
- Si dos símbolos comparten parte de las nociones o parte de los impactos, se deben repetir las partes comunes en ambos símbolos.

¹¹ En el caso de *Insumo(1)*, la utilización de los paréntesis se refiere a la representación de homónimos, los cuales son tratados detalladamente en el **Capítulo 12**.

- El nombre del símbolo debe ser registrado en el LEL tal como aparece en el UdeD, sin importar su longitud ni ninguna apreciación particular de quien está construyendo el glosario.
- Los impactos de los sujetos se deben corresponder con los verbos del LEL y viceversa.
- Los estados deben ser buscados en el contexto, porque en muchos casos no están visibles.

Además de los cambios antes mencionados, en el **Capítulo 12** se propone mejorar el uso de sinónimos, homónimos, jerarquías conceptuales del contexto y puntos de vista del contexto. También se propone utilizar el empotramiento de símbolos para detectar omisiones y mejorar la comprensión léxica.

Una vez que el LEL está completo comienza la actividad *Verificar*, que consiste en controlar los defectos que puedan volverlo inconsistente [Nentwich03] [Nguyen08]. Como se adelantó en el **Capítulo 6**, la verificación del LEL utiliza la técnica de Inspecciones con formularios [Kaplan00]. Una propuesta reciente para la verificación del LEL es la utilización de mapas conceptuales [Sebastián15], la cual se ha centrado en detectar omisiones y ambigüedades y en formular correcciones a las mismas. Sus resultados obtenidos son similares a los del uso de formularios.

Los principales defectos que se detectan en las primeras versiones del LEL pueden agruparse según se indica en la Tabla 11. A continuación, se describe cada grupo:

- **Defectos de Descripción**

En este grupo se incluyen los defectos “internos” de los símbolos, es decir, fundamentalmente aquellos vinculados con omisiones o falta de concordancia con el modelo utilizado.

Símbolos mal descriptos:

Descartando lo que habitualmente se conoce como errores de redacción, se entiende que un símbolo está mal descrito cuando su noción no se refiere exclusivamente a la denotación y su impacto a la connotación, o contiene

oraciones que no pertenecen al símbolo. Este último caso generalmente ocurre cuando se incorpora a un símbolo información perteneciente a otro.

Indudablemente, la diferencia entre la noción y el impacto es semántica, pero al extraer conocimiento del UdeD no siempre resulta evidente a los/las ingenieros/as de requisitos cómo separar las afirmaciones relacionadas con la “esencia” del símbolo, de aquéllas relacionadas con los efectos sobre el contexto, o sea qué es noción y qué es impacto.

También puede ocurrir que una oración contenga errores lógicos, éste es un caso más difícil de detectar y plantea dificultades similares a las relacionadas con el punto siguiente.

Grupo	Defectos
<i>De Descripción</i>	<ul style="list-style-type: none"> • Símbolos mal descriptos • Símbolos incompletos • Descripción incompatible con el tipo
<i>De Clasificación</i>	<ul style="list-style-type: none"> • Clasificación incorrecta
<i>De Identificación</i>	<ul style="list-style-type: none"> • Símbolos omitidos • Sinónimos incorrectos • Símbolos incorrectamente incluidos
<i>De Referencia</i>	<ul style="list-style-type: none"> • Falta de referencias a otros símbolos • Mal uso de símbolos

Tabla 11– Taxonomía de Defectos del LEL

Símbolos incompletos:

Se entiende que un símbolo se encuentra incompleto cuando falta total o parcialmente la noción o el impacto o se han omitido otros nombres del símbolo que representan sinónimos.

En general resulta difícil de detectar los símbolos incompletos más allá del caso obvio de una noción o un impacto completamente vacía, ya que el LEL mismo no contiene pistas que permitan fácilmente su detección.

El único signo, difícil de percibir, que permite sospechar la existencia de un símbolo incompleto ocurre cuando se está analizando un símbolo relevante para el UdeD, ya sea por ser frecuentemente mencionado por otros símbolos o

por una percepción “intuitiva”, pero sin embargo su noción y su impacto resultan demasiado lacónicas.

Descripción incompatible con el tipo:

Es un tipo de defecto muy frecuente, pero a su vez más simple de detectar y resolver. Este caso se presenta cuando el contenido de las sentencias no es consistente con lo esperado para ese tipo de símbolo. Esto ocurre, por ejemplo, cuando a un sujeto se le describen sus impactos sin enfatizar las acciones que realiza.

- **Defectos de Clasificación**

La asignación del tipo correcto a cada símbolo del LEL es intrínsecamente importante, pero resulta especialmente relevante para el proceso de derivación de Escenarios, ya que el conocimiento preciso de los símbolos de tipo sujeto y verbo es relevante en esta actividad. La persistencia de alguna DEO en la asignación de tipos puede traer como consecuencia la no detección de situaciones del UdeD.

Clasificación incorrecta:

Este defecto se presenta cuando a un símbolo se le ha asignado un tipo al que no pertenece. Este fenómeno ocurre por ejemplo cuando a los símbolos de tipo verbo no se los presenta en infinitivo y por ende tienden a confundirse con los sustantivos o estados asociados, es decir con las acciones o efectos de la ejecución del verbo. Otro caso frecuente, está asociado con los estados cuando se tiene un sujeto u objeto que puede tener más de un estado suele asociarse el estado al sujeto dificultando la asignación del tipo.

- **Defectos de Referencia**

La adhesión a los principios de circularidad y de vocabulario mínimo requiere una permanente atención por parte de los/las ingenieros/as de requisitos. Es frecuente que se produzcan estos defectos, especialmente con aquellos símbolos que son incorporados tardíamente al LEL.

Falta de referencia a otros símbolos:

Este caso se presenta cuando no se ha cumplido total o parcialmente con el principio de circularidad. Es decir, no se ha reforzado el uso de símbolos del LEL y en su lugar se han insertado partes de sus definiciones. Esto ocurre generalmente cuando se reemplaza el nombre del símbolo por una de sus nociones. Este tipo de defectos es de una dificultad intermedia porque no es directamente visible, pero

un símbolo con escasas o nulas referencias a otros es un signo que hace sospechar la presencia de este tipo de defecto. Mientras que la presencia de símbolos que no son mencionados o lo son en muy pocas oportunidades permite sospechar la presencia de este efecto en otros símbolos.

Mal uso del símbolo:

Esto ocurre cuando al describir símbolos se utiliza otro símbolo, también perteneciente al LEL, con un significado que no es el específico en el UdeD. Generalmente, el nombre del símbolo no aparece destacado como tal, excepcionalmente puede encontrarse el doble defecto de que el símbolo haya sido mal destacado. Ambos casos ocurren frecuentemente cuando un símbolo tiene un significado posible en un contexto más general que el específico del UdeD.

- **Defectos de Identificación**

Durante las actividades centrales de la construcción del LEL, es decir en la identificación, clasificación y descripción de los símbolos, se produce un gran reordenamiento de los mismos. Algunos son eliminados, otros son asociados como sinónimos y finalmente se pueden incorporar nuevos símbolos. Estos cambios pueden fácilmente introducir deformaciones en el LEL obtenido, fundamentalmente por meros defectos de transcripción.

Símbolos omitidos:

Puede tratarse de símbolos detectados, y por lo tanto, mencionados en otros símbolos cuya descripción se omitió o de símbolos no detectados. En el primero de los casos la manifestación es evidente y su corrección inmediata. En el segundo, se está en presencia de un defecto de difícil detección, aunque existen dos posibles indicadores utilizables. Por un lado, la presencia de símbolos del LEL con impactos y eventualmente nociones sin referencias a otros símbolos permite sospechar que en esos impactos o nociones existe una palabra o frase que amerita ser incorporada al LEL. Por otro lado, la existencia de palabras o frases repetidas en las descripciones de varios otros símbolos permite presumir la omisión de un símbolo.

Sinónimos incorrectos:

Este defecto se manifiesta generalmente a través de la presencia de sinónimos parciales. Un par de símbolos constituyen sinónimos parciales si en alguno de sus

usos uno de ellos puede ser reemplazado por el otro sin inconvenientes, pero en otros usos esto no es posible.

En este punto corresponde analizar si los sinónimos parciales se comportan en alguna de las siguientes formas:

- uno de ellos aparece como un subconjunto propio del otro,
- ambos símbolos tienen una intersección no vacía, pero ninguno es subconjunto propio del otro.

En el primero de los casos hay varias posibles explicaciones y varios aspectos a evaluar. Por ejemplo, se podría estar en presencia de la omisión de un tercer símbolo que no ha sido percibido y que también es sinónimo parcial del más amplio de los dos. No necesariamente tiene que existir ni debe ser uno sólo, puede haber varios. También puede ocurrir que el defecto ponga de manifiesto una incoherencia propia del UdeD. En el segundo de los casos, generalmente se está en presencia de símbolos que no son sinónimos, pero podría tratarse de la omisión de un símbolo que se usa expresamente en el UdeD para referirse a la intersección de ambos símbolos. Si así ocurriera estaríamos en presencia del caso anterior.

Una de las formas más frecuentes de sinónimos parciales ocurre cuando se trata de símbolos cuyo nombre es una frase compuesta por un sustantivo y modificador o verbo y complemento.

Símbolos incorrectamente incluidos:

Los símbolos incorrectamente incluidos pueden provenir de una imprecisión en el alcance del UdeD o de la identificación de términos pertenecientes al *vocabulario mínimo* sin ninguna noción o impacto destacable en el UdeD. La significativa diferencia en los orígenes posibles del defecto naturalmente provoca estrategias diferentes en su detección. Cuando la causa fue un error en el alcance del UdeD, no es sencillo un tratamiento individual asociado a cada símbolo, sino que es necesario un estudio más global del mismo. Si por el contrario la causa del defecto está relacionada con un error individual asociado a un símbolo en particular, su detección puede realizarse mediante una evaluación de las diferencias semánticas entre la noción y el impacto incluidas en el LEL con el significado habitual del término. La presencia de símbolos con muy pocas referencias a otros símbolos o que son poco mencionados en otros símbolos permite sospechar la presencia de este defecto.

Algunos de los defectos aquí presentados pueden detectarse durante la verificación, pero otros sólo podrán descubrirse en la validación con los clientes y usuarios. Los defectos de referencia, de clasificación, de descripción incompatible con el tipo y de sinónimos incorrectos pueden ser detectados mediante mecanismos de verificación. En los casos de símbolos mal descritos, símbolos incompletos y símbolos omitidos, dependiendo de sus manifestaciones, podrán ser detectados en la actividad de verificación o de validación. Aun así, es factible que un porcentaje mínimo no sea detectado a través de ninguno de estas actividades y surjan recién durante la construcción de los Escenarios.

En el estudio realizado en la presente tesis, se ha detectado la necesidad de modificar algunos formularios de inspección de la **Verificación** del LEL. Por razones de espacio, en el **Capítulo 13**, solo se mencionan algunos de estos cambios. Cabe aclarar que la inspección no cubre todos los defectos posibles, sino que se encarga fundamentalmente, de aquellos que pueden dejar inconsistente al modelo.

La última actividad es **Validar**. En general esta actividad se desarrolla una vez que el LEL tiene cierto grado de calidad. La validación como último paso del proceso no es el único momento en el cual se realiza, se debe contemplar que también se valida en simultáneo a la elicitación de información. La validación se realiza para asegurarse que el LEL refleja con la mayor exactitud posible, el léxico del macrosistema:

- Chequear la información elicitada (ratificar y rectificar las definiciones),
- Clarificar las dudas,
- Identificar sinónimos,
- Detectar nuevos símbolos,
- Eliminar símbolos incorrectos.

Si bien el proceso de validación no es formal, existen diferentes maneras de llevarlo a cabo:

- Particionar el LEL según algún criterio. Por ejemplo, entregar a cada usuario los símbolos relacionados con su actividad,
- Realizar reuniones con todos los usuarios para contrastar diferentes puntos de vista del vocabulario generado,

- Realizar reuniones e ir corroborando junto a diferentes usuarios las descripciones.

Todas estas opciones pueden ser utilizadas de manera individual o combinadas. La elección del mecanismo de validación es absolutamente dependiente del contexto.

Capítulo 8

Escenarios

Resumen

El modelo central del Proceso de Requisitos basados en Escenarios es justamente, los Escenarios. En este modelo se representan las situaciones observadas del UdeD como aquellas referidas a la proyección de los procesos del negocio con el Sistema de Software en ejecución. De esta manera, se pueden definir dos conjuntos de Escenarios: los Actuales y los Futuros. Si bien cada conjunto de Escenarios tiene sus particularidades, ambos comparten modelo, actividades, técnicas, etc. Por tal motivo, en el presente Capítulo se describen aquellos aspectos comunes y luego, en los Capítulos que siguen, las particularidades de cada uno.

Estrategias existentes para la construcción de Escenarios

Los Escenarios han sido utilizados con anterioridad, tal es el caso de [Potts95], [Schneider98], [Alspaugh99], [Gough95], [Jarke98]. El abordaje de la construcción de los Escenarios puede variar dependiendo de las circunstancias, sin embargo, se suelen destacar dos enfoques contrapuestos, top-down y bottom-up. Usualmente. En el caso de los Escenarios Actuales, el enfoque top-down requiere que el ingeniero/a de requisitos decida cómo se subdividirá el problema cuando aún no lo conoce. Por lo tanto, es una decisión con un alto riesgo de error, no sólo por no distinguirse correctamente los componentes principales involucrados en las divisiones sino también por perder información valiosa de las interrelaciones entre los mismos. Por otro lado, su estructura jerárquica obliga a observar el mundo que debe modelar de una manera que tal vez no sea la real, ocultando de esta manera, algunas estructuras paralelas, empotradas o superpuestas. Weidenhaupt et al. [Weidenhaupt98] han encontrado cuatro tipos de evolución para la definición de Escenarios en la práctica real: “i) *Top-down decomposition*”, “ii) *From black-box to white-box scenarios*”, “iii) *From informal to formal scenario definition*” and “iv) *Incremental scenario development*.” El primero, el segundo y el cuarto son enfoques top-down. A diferencia de los Escenarios Actuales donde el abordaje de construcción top-down puede perjudicar seriamente la completitud del modelo, en los Escenarios Futuros se convierte en una gran ventaja. Esto se debe a que cuando se comienzan a construir los EF ya se cuenta con un importante conocimiento del UdeD actual, representado en el LEL y en los EA. Los EA integradores, agrupan toda esta información existente y la disponen ordenada para su análisis. El grado de cambio que se espera en el contexto futuro determinará el grado de transformación de los EA, o sea, a menor grado de cambio en el futuro existirá un importante apareo entre los EA y los EF ya que sus objetivos se mantienen en el contexto futuro. En un contexto futuro con mucho cambio, los objetivos de los EA se transformarán.

Modelo Escenario

Son narrativas estructuradas de situaciones del contexto [Carroll95] [Zorman95], centrando la atención en su comportamiento. Si bien cada Escenario describe una situación particular, ninguno de ellos es completamente independiente del resto de los

Escenarios, sino que por el contrario cada uno guarda una relación con los otros [Booch94].

Las características esenciales de un Escenario son [Breitman01]:

- Cumplir un fin específico,
- Identificar actores y recursos para satisfacer el objetivo de la situación,
- Ocurrir en un tiempo dado continuo (sin interrupciones),
- Realizarse en un contexto geográfico establecido,
- Poder ser restringida por ciertas condiciones,
- Ser independiente (comprensible aisladamente de otras situaciones, aunque relacionada con ellas),
- Ser concreta y
- Puede contener caminos alternativos de acción.

En la presente tesis se utiliza la propuesta de Escenarios de Leite et al. [Leite97], la cual ha sido actualizada primero en [Leite00] y luego, en [Leite04]. La estructura de cada Escenario es de un conjunto de componentes que permiten describir con claridad una situación particular del macrosistema. Estos componentes son:

Título, Objetivo, Contexto, Recursos, Actores, Episodios y Excepciones.

El *título*, el *objetivo*, el *contexto*, los *recursos*, los *actores* y las *excepciones* son declarativos, mientras que los *episodios* son un conjunto de sentencias que muestran descripciones operacionales de comportamiento de los actores en el Escenario. El *contexto*, los *recursos* y los *episodios* pueden contener *restricciones*, las cuales pueden estar asociadas a requisitos no funcionales.

Cabe destacar que, durante la construcción de Escenarios Actuales, las **restricciones** se refieren a RNF del negocio, los cuales rara vez se transforman en RNF del software. Mientras que durante la construcción de EF, efectivamente corresponden a RNF del software.

Se ha incorporado como **mejora** en el **Capítulo 16**, las restricciones a los actores.

Los episodios pueden ser en sí mismos Escenarios, posibilitando relaciones de jerarquía entre Escenarios y sub-Escenarios. Esta relación de jerarquía surge naturalmente cuando al descubrir una situación se prefiere no detallar la situación de menor cuantía que es parte

de la primera. Asimismo, esta relación es utilizada para construir los Escenarios Integradores, los cuales describen grupos de situaciones relacionadas. Estos Escenarios Integradores no tienen una unidad temporal o geográfica como los Escenarios y se utilizan para adquirir una visión global del UdeD.

Escenario: descripción de una situación que ocurre en el contexto del problema.
 Sintaxis: Título + Objetivo + Contexto + {Recursos}₁^N + {Actores}₁^N + {Episodios}₂^N + {Excepciones}

Título: identificación del Escenario. En el caso de un sub-Escenario, el título es el mismo que la sentencia episodio, sin las restricciones.
 Sintaxis: Frase | ([Actor | Recurso] + Verbo + Predicado)

Objetivo: finalidad a ser alcanzada. El Escenario describe la forma de lograr el objetivo.
 Sintaxis: [Actor | Recurso] + Verbo + Predicado

Contexto: compuesto por al menos uno de los siguientes ítems:
 Ubicación Geográfica: lugar físico donde se produce el Escenario.
 Ubicación Temporal: especificación de tiempo para el desarrollo del Escenario.
 Precondición: estado inicial del Escenario.
 Sintaxis: {Ubicación Geográfica} + {Ubicación Temporal} + {Precondición}

 donde Ubicación Geográfica es:
 Frase + {Restricción}

 donde Ubicación Temporal es:
 Frase + {Restricción}

 donde Precondición es:
 [Sujeto | Actor | Recurso] + Verbo + Predicado + {Restricción}

Recursos: elementos físicos o información relevantes que deben estar disponibles en el Escenario.
 Sintaxis: Nombre + {Restricción}

Actores: personas o estructuras organizacionales que tienen un rol en el Escenario.
 Sintaxis: Nombre

Episodios: conjunto de acciones que detallan al Escenario y proveen su comportamiento. Un episodio también puede ser descrito como un Escenario.
 Sintaxis (usando BNF parcial):

```
<episodios> ::= <serie grupo> | <serie episodios>
<serie grupo> ::= <grupo> <grupo> | <grupo no secuencial> | <serie grupo> <grupo>
<grupo> ::= <grupo secuencial> | <grupo no secuencial>
<grupo secuencial> ::= <sentencia básica> | <grupo secuencial> <sentencia básica>
<grupo no secuencial> ::= # <serie episodios> #
<serie episodios> ::= <sentencia básica> <sentencia básica> | <serie episodios> <sentencia básica>
<sentencia básica> ::= <sentencia simple> | <sentencia condicional> | <sentencia optativa>
<sentencia simple> ::= <sentencia episodio> CR
<sentencia condicional> ::= Si <condición entonces <sentencia episodio> CR
<sentencia optativa> ::= [ <sentencia episodio> ] CR
donde <sentencia episodio> se describe:
                (([Actor | Recurso] + Verbo + Predicado) | ([Actor | Recurso] + [Verbo] + Título)) +
{Restricción}
```

Excepciones: generalmente reflejan la falta o mal funcionamiento de un recurso. Una excepción impide el cumplimiento del objetivo del Escenario. Eventualmente se puede indicar el tratamiento de la excepción a través de un Escenario.
 Sintaxis: Causa [(Solución)]
 donde Causa es:
 Frase | ([Sujeto | Actor | Recurso] + Verbo + Predicado)
 donde Solución es:
 Título

Restricción: es un requerimiento de calidad o alcance referido a una entidad dada. Es un atributo del Recurso, Episodio o de los subcomponentes del Contexto.
 Sintaxis: ([Sujeto | Actor | Recurso] + [No] Debe + Verbo + Predicado) | Frase

+ significa composición, {x} significa cero o más ocurrencias de x, () para agrupar, | para or, [x] significa que x es opcional.

Figura 38 - Modelo Escenarios

Las relaciones entre los Escenarios y los sub-Escenarios responden más a aspectos de granularidad de la representación que a cuestiones de factorización de aspectos comunes a situaciones. Es así que se debe evitar asociar conceptualmente los sub-Escenarios a construcciones como las funciones, los procedimientos o los métodos de los lenguajes de programación.

Cada Escenario tiene un *objetivo* que es satisfecho en el curso de acción de los episodios. El *contexto* es descripto detallando ubicación geográfica, ubicación temporal y precondiciones. Cada componente puede ser expresado por unas o más sentencias simples vinculadas por los conectores “y” y “o”. Los *episodios* se representan a través de sentencias simples, condicionales u opcionales. Las sentencias simples expresan acciones atómicas. Una secuencia de episodios implica en sí misma un orden de precedencia. Para indicar un orden no-secuencial se debe utilizar una sintaxis que permita agrupar dos o más episodios que indiquen paralelismo u orden no secuencial (#). Las sentencias condicionales son las que dependen de la ocurrencia de una condición específica. La condición puede ser interna o externa al Escenario. Las sentencias opcionales son aquellas que dependen de condiciones que no pueden ser explicitadas, y se indican encerrándolas entre corchetes (“[]”). Independientemente del tipo, un episodio puede ser expresado como una acción simple o puede ser concebido en sí mismo como un Escenario, posibilitando la descomposición en sub-Escenarios. Un sub-Escenario es identificado cuando:

- Aparece una situación de menor cuantía con un objetivo concreto y preciso dentro de un Escenario,
- Aparecen varios episodios condicionales con la misma condición,
- Se detecta una misma situación dentro de un Escenario que se repite en diferentes Escenarios, la cual no involucra ninguna particularidad relaciona a alguno de esos Escenarios. En este caso se debe asegurar que tanto el contexto temporal como la ubicación geográfica sean coherentes en todos los Escenarios involucrados.

Las *restricciones* aplican al componente Contexto, Recursos y Episodios. Algunas restricciones aparecen espontáneamente en el UdeD, mientras que otras deben ser buscadas examinando los episodios. Las *Excepciones* suceden cuando no es posible lograr el objetivo del Escenario. Algunas causas de excepción son elicitadas desde las fuentes de información mientras que otras pueden deducirse analizando los episodios y la no

disponibilidad o mal funcionamiento de los recursos. Cuando se descubren las causas de una excepción, el ingeniero/a de requisitos debe investigar cómo es tratada la excepción en el UdeD, surgiendo así una nueva situación que podría necesitar ser descrita en otro Escenario. Cuando se descubren las causas de una excepción, el ingeniero/a de requisitos debe informarse cómo es tratada la excepción en el UdeD, detectando si existen nuevas situaciones que podrían necesitar ser descritas en otros Escenarios.

Mientras se desarrolla un Escenario puede ocurrir una *excepción*, indicando un obstáculo en la realización del objetivo del Escenario. El tratamiento de la excepción puede satisfacer el objetivo o no. Cada excepción es descrita en una sentencia simple que especifica la causa de la interrupción. Cuando la excepción tiene un tratamiento elaborado, este se describe en otro Escenario y se incluye sólo el título del mismo (sub Escenario). Es de esperar que durante la construcción de los EF aparezcan algunas excepciones relacionadas con inconvenientes en la ejecución del nuevo Sistema de Software. Por ejemplo, por cuestiones de consistencia se determina que deben existir algunos datos obligatorios del cliente. Por otro lado, existe una política organizacional que determina que toda persona u organización que se contacte de alguna manera debe ser tomado desde ese momento como un cliente potencial. A la organización le interesa tener algún contacto con el cliente, para enviar publicidad, promociones, etc. O sea que los datos necesarios para la organización pueden diferir de la necesidad de consistencia del sistema. En este caso, los datos obligatorios que salen del EF actúan como una restricción durante la ejecución, generando un conflicto con la política de la organización acerca de no perder información de ningún posible cliente. En este caso, la excepción debe describir que si faltan datos obligatorios igual se debe resguardar la información del cliente para completarla lo antes posible. Luego, en el diseño se determinará la mejor manera de hacerlo, generando posiblemente casos especiales o alternativos durante la ejecución. Estas excepciones deben aparecer durante la validación de los EF, en las cuales deben participar tanto personas autorizadas a tomar decisiones como aquellas que conocen los aspectos estratégicos de la organización.

Cómo pensar un Escenario

Antes de comenzar, es necesario aclarar que en esta sección se describe cómo deben ser pensados en función al modelo, pero en cada proceso de construcción (Escenarios Actuales, Escenarios Integradores, Escenarios Futuros) puede determinar un orden

diferente. Por ejemplo, al derivar un Escenario Actual desde el LEL, sólo se completan los componentes para los cuales se tiene información.

Si bien un Escenario puede ser completado sin un orden establecido, el orden de aparición de cada componente en el modelo de la Figura 38 no es arbitrario, sino que induce a un proceso cognitivo donde cada componente lleva a pensar en el siguiente. De esta manera se debe pensar como está compuesta la situación en general, para luego pasar a los detalles descriptos en los episodios. En este marco, se debe comenzar por el **título** del Escenario que debe ser representativo de su contenido. El **objetivo** indica lo que se desea alcanzar. Es un componente transversal a todo el Escenario, delimitando al resto de los componentes. Por su importancia, se debe prestar mucha atención a su definición, la cual debe ser clara y precisa. El Objetivo se alcanza en un **contexto** determinado. Para una mayor claridad de este componente, se separó en subcomponentes: ubicación geográfica (UG), ubicación temporal (UT) y precondiciones. La ubicación geográfica, determina el espacio físico donde se desarrolla el Escenario. La ubicación temporal, determina aspectos temporales que pueden condicionar la realización del Escenario, como, por ejemplo, que se realice una vez por mes. Finalmente, las precondiciones indican las condiciones previas que restringen la realización del Escenario. A diferencia de la ubicación temporal, éstas deben ser resueltas en el UdeD. Sólo las precondiciones pueden tener restricciones. Cabe aclarar que una precondición nunca es un Escenario, sino que describe qué acciones previas inmediatas deben haberse realizado. Con los componentes previos es posible determinar los **actores** que participan en la situación. Se debe tener en cuenta, que pueden ser sujetos del LEL o no serlos. Si no son sujetos del LEL se debe analizar la posibilidad de haberlos omitido, esto dependerá de la importancia del actor en cuestión en el UdeD. Por ejemplo, el chofer de un transporte participa en el Escenario de descarga de mercadería, pero no es relevante para el control stock. Los actores realizan alguna acción directa o indirecta. Por ejemplo, un actor registra la salida de mercadería del stock (acción directa), mientras que otro sólo recibe el informe generado (acción indirecta). En el componente Actores deben ir todos. En este momento, se debe precisar cuáles son los **recursos** que utilizan los actores para alcanzar el objetivo del Escenario. Igual que sucede con las precondiciones, pueden aparecer restricciones a los recursos, como ser que una impresora necesite una tinta especial para imprimir los cheques o que un formulario debe estar completo. Las restricciones a los recursos tienen el mismo carácter restrictivo que las precondiciones del contexto. La decisión de incorporarla en

un componente u otro (recursos o contexto), depende de la importancia o visibilidad que se le desee otorgar a la restricción. Por ejemplo, en el caso de la tinta tiene más relevancia en el componente Recursos, mientras que un formulario este lleno, es más importante como Precondición. Hasta este momento se definió el alcance del Escenario y ahora, es necesario representar las acciones involucradas en el componente **episodios**. En cada episodio se debe realizar una acción en la cual participa uno o varios actores, los cuales utilizan los recursos mencionados. Estas acciones deben realizarse dentro de la ubicación geográfica y temporal descritas en el componente contexto y deben comenzar inmediatamente después de la precondición y terminar cuando se alcanzó el objetivo.

Es necesario que los componentes actores y recursos contengan todos los actores y los recursos involucrados en los episodios y todo actor o recurso indicado en el correspondiente componente participe en al menos un episodio. Se debe prestar atención a que pueden aparecer actores y recursos implícitos en el texto de los episodios que deben ser explicitados en los componentes correspondientes.

Cada episodio debe representar, siempre que sea posible, una única acción, o sea, tener un solo verbo principal. Por otro lado, la granularidad de los episodios es de suma importancia. Si un episodio tiene un verbo que se conforma de varias acciones (por ejemplo, controlar), se debe descomponer en varios episodios o analizar la incorporación de un sub-Escenario. Los episodios, como las precondiciones y los recursos, pueden tener restricciones. Como se mencionó, al describir un grupo de episodios, se puede reconocer la presencia de situaciones de menor nivel, en este caso también se debe analizar la presencia de un sub-Escenarios. Esta condición es analizada en la sección que sigue. Una vez que se completó el Escenario es necesario volver a revisar algunos aspectos. Se debe asegurar que el alcance del objetivo se satisface con los episodios y que estos últimos, no exceden dicho Objetivo. Los episodios también deben ser revisados para confirmar la secuencia de orden o no-secuencia de orden y detectar episodios condicionales y opcionales que no fueron identificados. Estas omisiones pueden determinar la presencia de sub-Escenarios. Con el Escenario completo, se debe analizar la posibilidad de que aparezcan **excepciones** o casos alternativos, en ocasiones aparecen espontáneamente y en otras hay buscarlas intencionalmente. Algunas causas de excepción son elicitadas desde las fuentes de información, por ejemplo, preguntando al usuario si existe alguna ocasión en la cual el Escenario deba ser interrumpido. Mientras que otras pueden deducirse analizando los episodios y la no disponibilidad o mal funcionamiento de los recursos.

Cuando se descubren las causas de una excepción, se debe analizar en el UdeD cómo es resuelta ya que pueden ser con una acción simple o requerir la ejecución de un Escenario. Es importante remarcar que durante la construcción de cada Escenario se debe pensar exclusivamente en la situación en cuestión, sin tener en cuenta el resto de los Escenarios, con la única excepción de los sub-Escenarios. Esto permite pensar de manera independiente cada situación.

Para distinguir las *excepciones* de los *episodios condicionales* es necesario prestar especial atención al contexto. Si las acciones que atienden la potencial causa de una excepción se llevan a cabo en la misma ubicación geográfica y temporal, sin ningún tipo de discontinuidad, no se trata de una excepción. Por ejemplo, si hay un fallo de una impresora, pero se recurre inmediatamente a un talonario manual, la representación correcta es la de episodios condicionales, mientras que si el documento se emite posteriormente y se entrega por una vía alternativa se trata de una excepción.

A continuación, se resumen las sugerencias para describir un Escenario:

- Son preferibles las sentencias cortas,
- Las sentencias deben ser escritas maximizando el uso de otros símbolos del LEL,
- Los actores y recursos deben ser preferentemente símbolos del LEL del tipo sujeto y objeto respectivamente,
- El objetivo debe ser preciso y concreto,
- Por lo menos uno de los sub componentes del contexto debe ser llenado,
- Todos los recursos enunciados en el componente Recursos deben estar utilizados en los episodios, los recursos triviales deben ser excluidos,
- No se deben incluir los recursos de los sub-Escenarios,
- Los actores del componente Actores deben estar en los episodios y cumplir algún rol directo o indirecto,
- No se deben incluir los actores de los sub-Escenarios,
- Se debe tratar de evitar el uso de más de un símbolo verbo por sentencia,
- Se debe describir los episodios en tiempo presente y

- Evitar usar: podría, puede, debe en los episodios.

Se debe tener presente que, durante la generación de estas sugerencias, no existía el LEL de Requisitos (LEL_R) descrito en el **Capítulo 15**. A partir de la presente tesis, se debe utilizar el LEL o LEL_R, según se esté trabajando en el UdeD actual o futuro respectivamente.

Finalmente, es importante resaltar una característica muy relevante de los Escenarios, es su poder de contextualización. O sea, las situaciones se describen incorporando las acciones involucradas. La única restricción para los episodios es estar en el alcance del objetivo del Escenario, pero no necesariamente en el alcance del Objetivo General del Sistema. Esto permite darle una amplitud al Escenario que facilita comprenderlo en el contexto real. Lo mismo sucede con los Escenarios, pueden existir con el único fin de ayudar a comprender el UdeD, aun cuando no están en el alcance del Objetivo General del Sistema. A tal punto son importantes, que pueden permanecer y evolucionar con el Proceso de Requisitos hasta el momento de la especificación, donde son excluidos. A continuación, se presenta un ejemplo de un Escenario que excede el Objetivo General del Sistema, pero fue incluido porque aportaba mayor conocimiento del contexto:

Objetivo General del Sistema: “optimizar la planificación de la producción”.
Escenario: “generar pedidos de producción”. En este caso, el Escenario ayuda a comprender como llegan los pedidos a la planificación, pero exceden el Objetivo General del Sistema.

En el **Capítulo 16** se han incorporado algunas mejoras para generar un conjunto de Escenarios de mayor calidad. Entre ellas:

- ✓ Utilizar la *Vista Clasificación* del LEL, la cual permite tener información de los tipos, las jerarquías y los estados asociados a los símbolos utilizados.
- ✓ Incorporar puntos de vista en los EA, para luego ser analizados en la construcción de los EF.
- ✓ Mejorar la expresividad incorporando episodios ubicuos.
- ✓ Incorporar restricciones a los actores.
- ✓ Modificar el modelo para reflejar estos cambios.

Jerarquía de Escenarios

Los Escenarios describen situaciones del macrosistema dentro de una estructura jerárquica. Por lo tanto, es posible identificar situaciones de menor nivel adentro de otras situaciones más complejas. De esta manera, como puede observarse en la Figura 39 se identifican tres niveles de Escenarios. En principio cabe destacar que todos los niveles se conforman de Escenarios y por lo tanto, utilizan el mismo modelo (ver Figura 38) aunque atienden algunas particularidades en cuanto al contenido de algunos componentes. La diferencia entre ellos radica en la amplitud de la situación que representan. Los Escenarios Integradores representan situaciones globales del macrosistema, los Escenarios situaciones individuales que, a su vez, pueden estar conformados por situaciones más específicas, representadas en los sub-Escenarios.

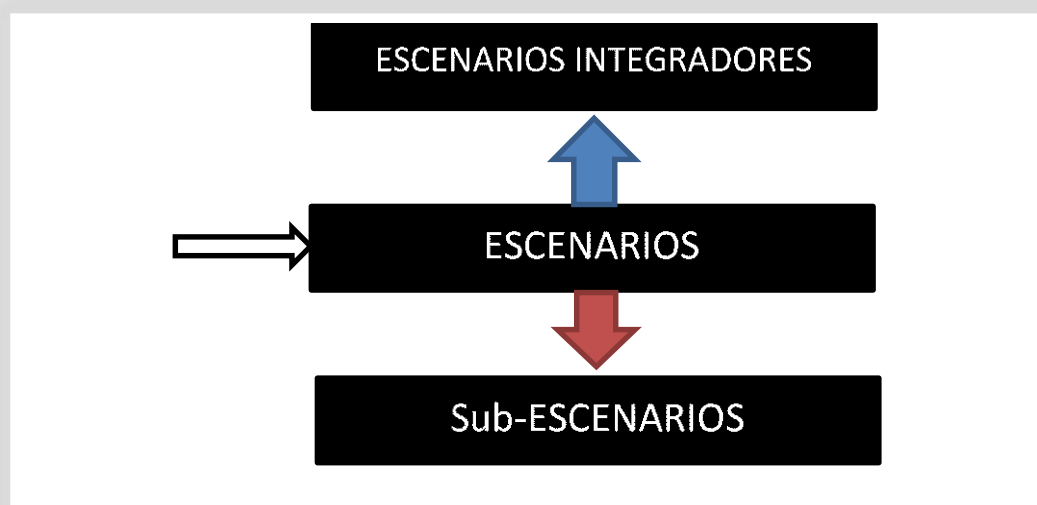


Figura 39 – Jerarquía de Escenarios

Para indicar que un Escenario es parte de otro, se utiliza el Título del Escenario en mayúscula, como es el caso de *VERIFICA LOS CÁLCULOS* de la Figura 40. Estos episodios son en sí mismo hipervínculos dentro del conjunto de Escenarios.

Se debe tener en cuenta que los actores y recursos de un Escenario que es parte de otro, no deben ser incluidos en el Escenario que los contiene, tal es así, que los Escenarios Integradores deben tener sus componentes actores y recursos vacíos. Los Escenarios que contienen sub-Escenarios no deben incluir los actores y recursos de estos últimos. Por ejemplo, en el sub-Escenario *Rechazar Presupuesto* participa personal administrativo, pero estos actores no se han incluidos en el componente Actores de *Revisión de presupuestos*.

EXAMINAR LOS RESULTADOS

Objetivo: registrar en el [cuaderno de asignación de número de análisis](#) los [resultados](#) obtenidos

Contexto:

Ubicación Geográfica: [oficina técnica](#)


Ubicación Temporal:

Precondición: Debe tener en su poder el [cuaderno del analista](#) o [cuaderno del analista de microbiología](#)

Recursos: [cuaderno del analista de microbiología](#), [cuaderno del analista](#), [cuaderno de asignación de número de análisis](#), [resultados](#), hojas de Excel, sello, [cálculos](#).

Actores: [Jefe de control de calidad](#)

Episodios:

- # El [Jefe de control de calidad](#) transcribe del [cuaderno del analista](#) o del [cuaderno del analista de microbiología](#) a la parte inferior de la hoja del [cuaderno de asignación de número de análisis](#) los [parámetros](#), las [especificaciones](#), los [resultados](#) y el folio del cuaderno donde están los [cálculos](#).
- **VERIFICA LOS CÁLCULOS** 
- Controla que todos los datos que transcribió estén correctos #
- Firma y sella la hoja del [cuaderno de asignación de número de análisis](#)
- Si los [resultados](#) no están dentro de los [parámetros](#) esperados pero hay posibilidad de corregirlos entonces aclara en la misma hoja qué se debe hacer.
- Si los [resultados](#) no están dentro de los [parámetros](#) esperados pero no existe posibilidad de corregirlos entonces **IDENTIFICA INSUMOS(1) RECHAZADOS**
- Anota en una hoja borrador los [protocolos](#) por generar

Excepciones:

Figura 40 – Ejemplo de Escenario con sub-Escenario (Caso SCC)

Operaciones estructurales de Escenarios

Durante la descripción de los Escenarios, principalmente cuando participan varios ingenieros/as, pueden quedar algunos aspectos ocultos que deben ser mejorados para asegurar la comprensión de las situaciones descriptas y facilitar su lectura. Por lo tanto,

una vez que los Escenarios están completos, se los debe *Verificar* y *Validar* antes de comenzar la actividad *Organizar*. Esta última actividad se realiza cuando se tiene confianza en la calidad de los Escenarios construidos, idealmente cuando las listas DEO están vacías. La actividad *Organizar* se divide en dos sub-actividades: *Reorganizar* e *Integrar*.

La sub-actividad *Reorganizar* Escenarios consiste en aplicar un conjunto de operación que mejoran la relación entre los Escenarios. Estas operaciones son más relevantes para los Escenarios Actuales, ya que el mismo proceso de construcción de los Escenarios Futuros evita varios de los problemas mencionados. Aun así, es posible aplicar estas operaciones a cualquier conjunto de Escenarios, algo similar se utilizó en [Breitman01]. La duración de algunas situaciones puede llevar a dudar si es mejor representarla mediante un solo Escenario o con dos o más. En algunos casos, dos Escenarios están tan relacionados temporalmente que puede ser mejor unificarlos en un solo Escenario. Por otro lado, un Escenario con un lapso de vida largo conteniendo episodios no muy compactos temporalmente pueden aconsejar la creación de dos o más Escenarios diferentes.

Básicamente la *Reorganización* de Escenarios consiste en ensamblar dos o más Escenarios en uno o fragmentar un Escenario en dos o más. El ensamblado se realiza cuando un único Escenario fue artificialmente dividido, mientras que se fragmenta un Escenario cuando contiene más de una situación. Se describen estas operaciones como pares en distintas dimensiones. Las tres dimensiones principales en las cuales la *Reorganización* de Escenarios puede aplicarse son:

- Sub-Escenario;
- Situaciones con variaciones;
- Contexto temporal.

A continuación, se define para cada dimensión las operaciones que deberían usarse.

Hay dos operaciones en la *dimensión sub-Escenario*:

- * Empotrar
- * Factorizar

Hay dos operaciones en la *dimensión situaciones con variaciones*:

- * Consolidar
- * Dividir

Hay dos operaciones en la *dimensión contexto temporal*:

- * Fusionar
- * Separar

Es necesario aclarar que las operaciones fueron pensadas originalmente muy cercanas a la programación y, a partir de esta tesis, se entiende que es necesario repensarlas desde las situaciones que conforman el contexto.

Para establecer la necesidad de las operaciones de composición/descomposición, se deben identificar previamente algunas propiedades y relaciones de los Escenarios. Cuando un Escenario presenta propiedades especiales principalmente en sus episodios, puede aplicarse descomposición. Por otro lado, la composición puede usarse cuando se descubren relaciones específicas entre los Escenarios: superposición fuerte, orden de precedencia o relación jerárquica.

Debe notarse que tanto la composición como la descomposición involucran el riesgo de degradación semántica. La descomposición crea dos o más Escenarios donde había previamente uno sólo. Este procedimiento no debe crear situaciones “artificiales”; es decir, la descomposición debe aplicarse sólo cuando los Escenarios resultantes describan mejor al UdeD. Por el contrario, la composición reemplaza dos o más Escenarios por uno. Las operaciones se sugieren como guías para mejorar la comprensión y administración de los Escenarios. Por lo tanto, la reorganización no debe asumirse como una actividad obligatoria sino como una buena práctica.

A continuación, se describen las operaciones con sus correspondientes propiedades o relaciones entre Escenarios:

Operaciones en la dimensión sub-Escenario

- * **Empotrar** (ver Figura 41): se puede aplicar cuando se detectan sub-Escenarios no relevantes con poca ocurrencia en otros Escenarios. Aplanar incorpora los episodios del sub-Escenario dentro de cada Escenario que lo menciona. El sub-Escenario debería presentar la misma granularidad que el resto del Escenario en el que se incrusta. El sub-Escenario original se elimina cuando todas sus ocurrencias se han aplanado. Esta operación permite reducir la cantidad de Escenarios no relevantes y, por lo tanto, facilita su

administración. El conjunto resultante de Escenarios decrece la profundidad de la jerarquía en el punto de aplanamiento.

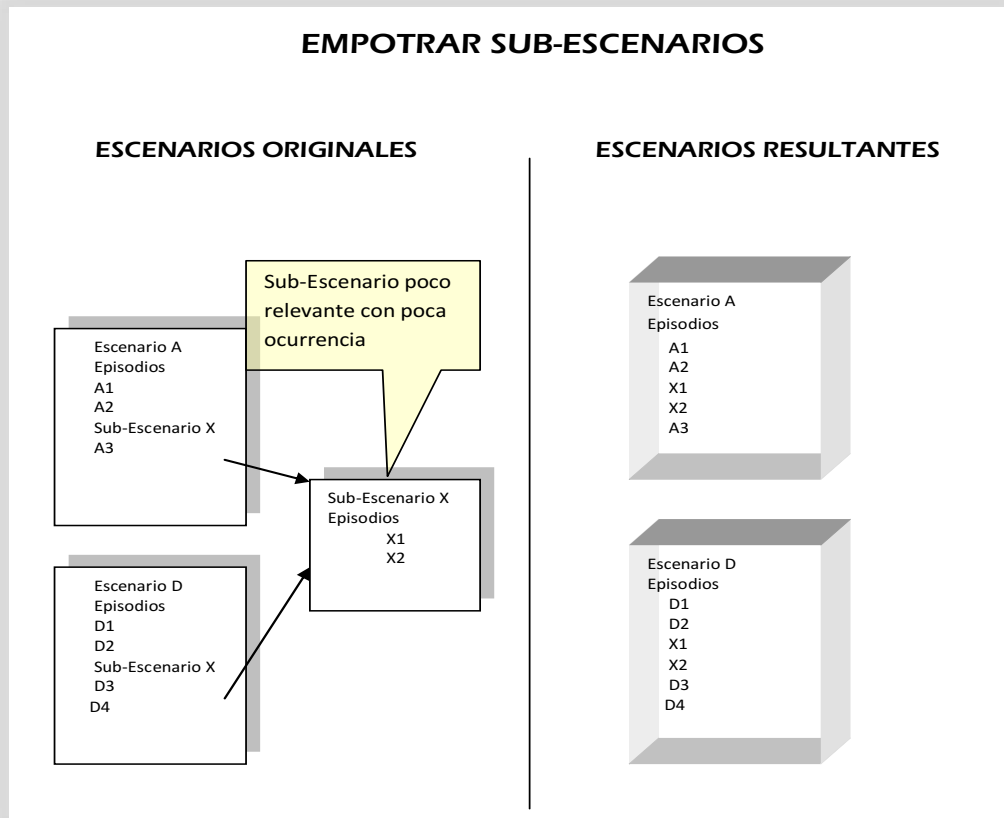


Figura 41 – Operación Empotrar Escenarios

Se debe estar seguro que el Sub-Escenario no representa una situación del contexto. De existir como tal, se recomienda no empotrar y mantener esas situaciones visibles. Solo se debe aplicar cuando el sub-Escenario no describe una situación en sí misma.

* **Factorizar** (ver Figura 42): puede aplicarse cuando dentro de un Escenario se detecta un conjunto de episodios muy relevantes o que presentan diferente nivel de detalle en relación al resto; en cualquiera de estos casos, el conjunto merece un tratamiento especial. Esta operación también se puede aplicar cuando se descubre la ocurrencia de un mismo conjunto de episodios en dos o más Escenarios con comportamientos comunes. Factorizar crea un sub-Escenario factorizando episodios de uno o más Escenarios. El grupo de episodios se reemplaza por el título del nuevo

sub-Escenario que los contiene. Factorizar hace que los Escenarios sean más fáciles de comprender y reusar.

La Figura 43 corresponde a un caso especial que se da cuando es factible factorizar entre varios Escenarios, cuando todos tienen el mismo contexto y no aparecen particularidades que los distingan. Por tal motivo se incorporó la Figura 42 que muestra el caso más encontrado para factorizar, cuando aparece una situación de menor nivel que se desea visualizar mejor.

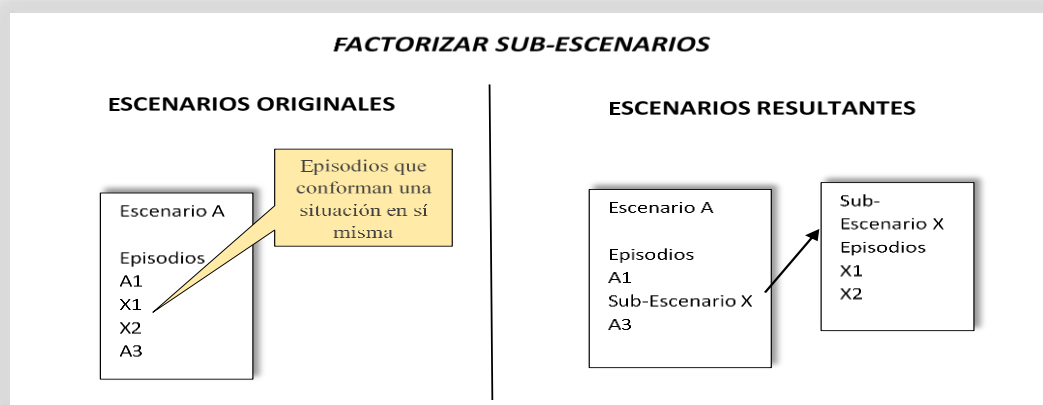


Figura 42 –Operación Factorizar Escenarios

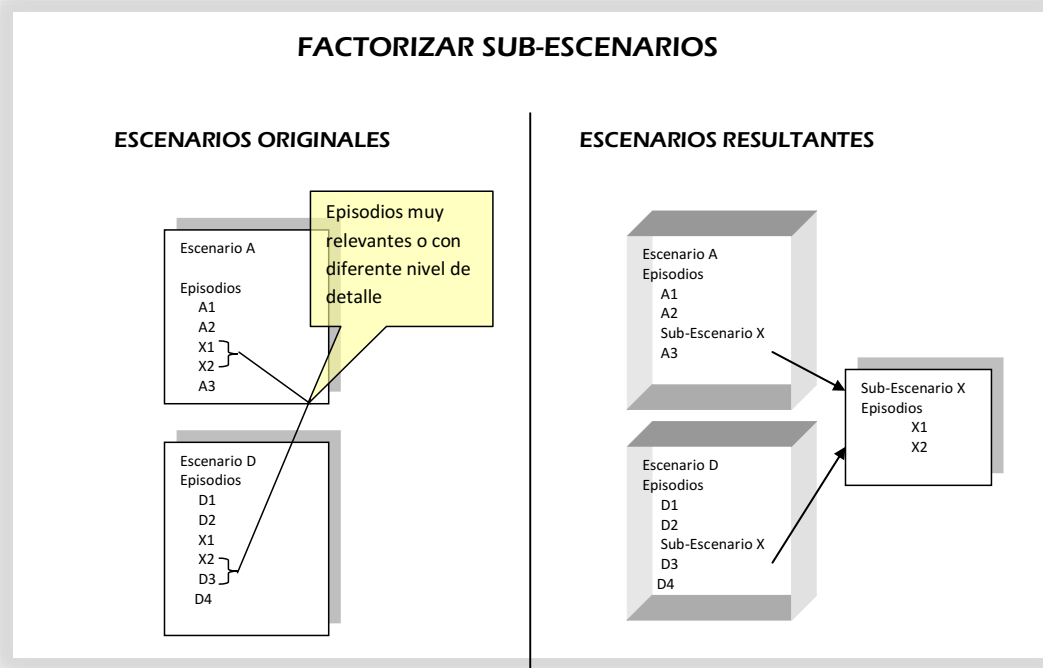


Figura 43 – Caso especial de la Factorización Escenarios

Operaciones en la dimensión situaciones con variaciones

* **Consolidar** (ver Figura 44): puede aplicarse a Escenarios superpuestos. Dos o más Escenarios tienen superposición fuerte si sus objetivos y contextos son similares y tienen varias coincidencias en sus episodios. Los Escenarios originales deben presentar el mismo “núcleo de acción”. Consolidar copia los episodios comunes de los Escenarios originales a un Escenario nuevo y crea nuevos episodios condicionales usando los episodios no compartidos, siendo la condición parte de las precondiciones de los Escenarios originales. Los Escenarios originales se eliminan. En consecuencia, esta operación permite reducir redundancia en el conjunto de Escenarios.

Se debe tener cuidado acerca de la necesidad de incluir un conector del tipo “Y” en el Nombre del Escenario o en su Objetivo. Esta necesidad indicaría que la consolidación es inadecuada.

* **Dividir** (ver Figura 45): puede aplicarse cuando se detecta la presencia de varios episodios condicionales, dirigidos por la misma condición. Dividir produce dos nuevos Escenarios, trasladando la condición disparadora a la precondición de ambos Escenarios. Los episodios que no contienen esta condición disparadora se trasladan tal cual a ambos Escenarios. Por otro lado, los episodios con la condición disparadora se trasladan sólo al Escenario correspondiente eliminando la condición. El Escenario original se elimina. Esta operación ayuda a evitar Escenarios confusos, reduciendo así la complejidad.

Operaciones en la dimensión del contexto temporal

* **Fusionar** (ver Figura 46): puede aplicarse a Escenarios que presentan un orden temporal continuo, objetivos complementarios y contextos combinables. No debe existir una brecha temporal entre estos Escenarios. Una brecha temporal ocurre cuando existe un intervalo de tiempo largo entre dos Escenarios. Fusionar copia los episodios de cada Escenario original al nuevo Escenario en el orden correspondiente. Se eliminan los Escenarios originales, dando la oportunidad de reducir la cantidad de Escenarios y facilitando entonces su administración.

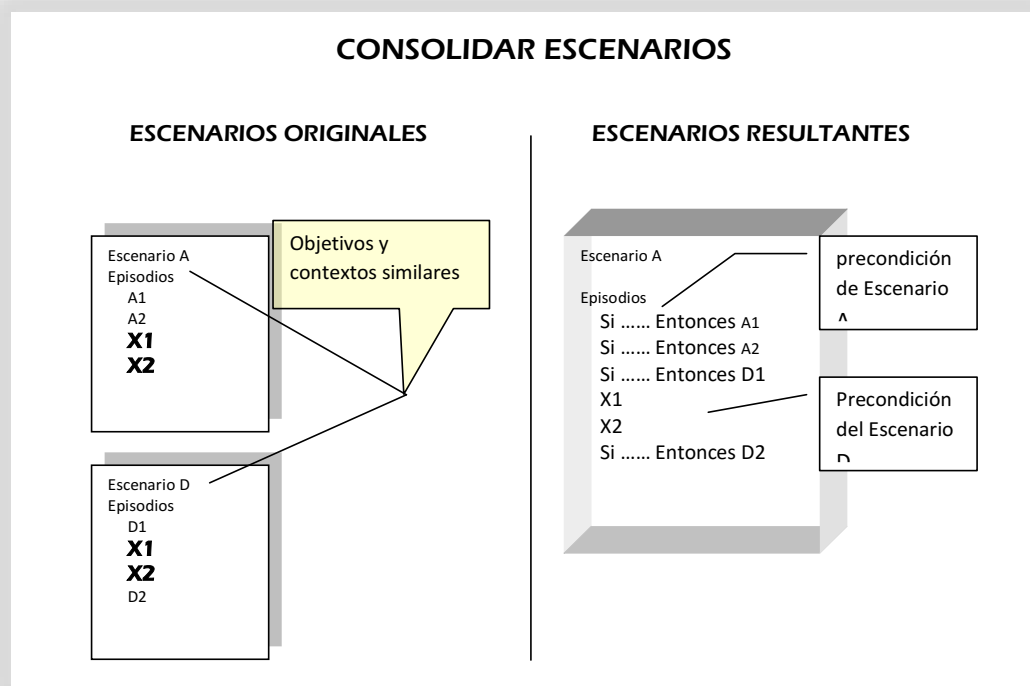


Figura 44 – Operación Consolidar Escenarios

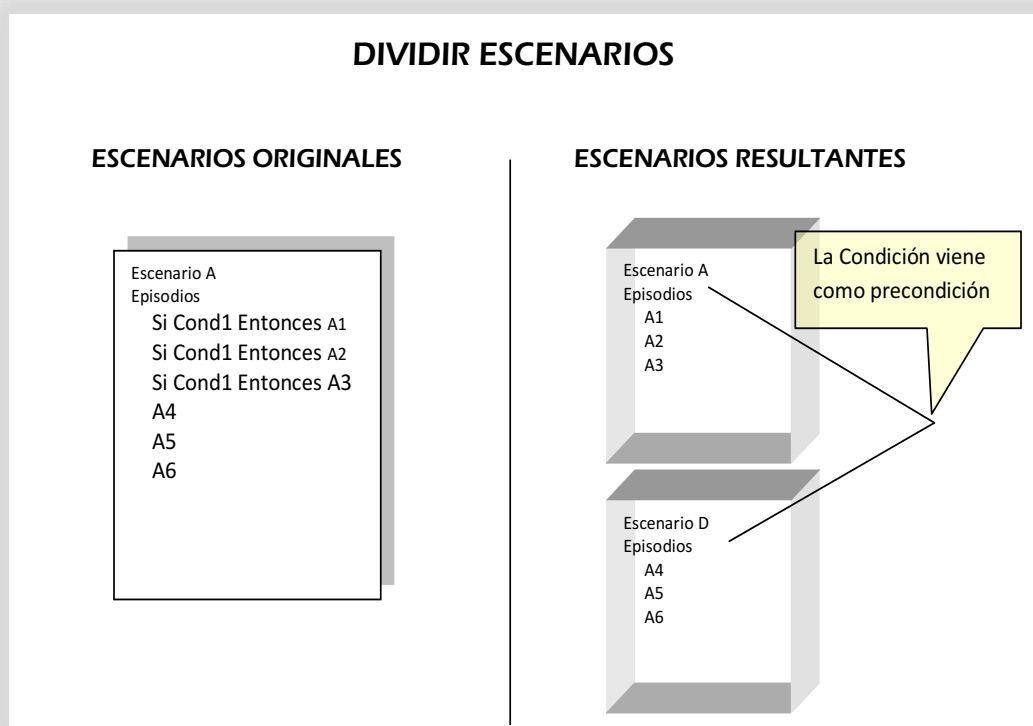


Figura 45 – Operación Dividir Escenarios

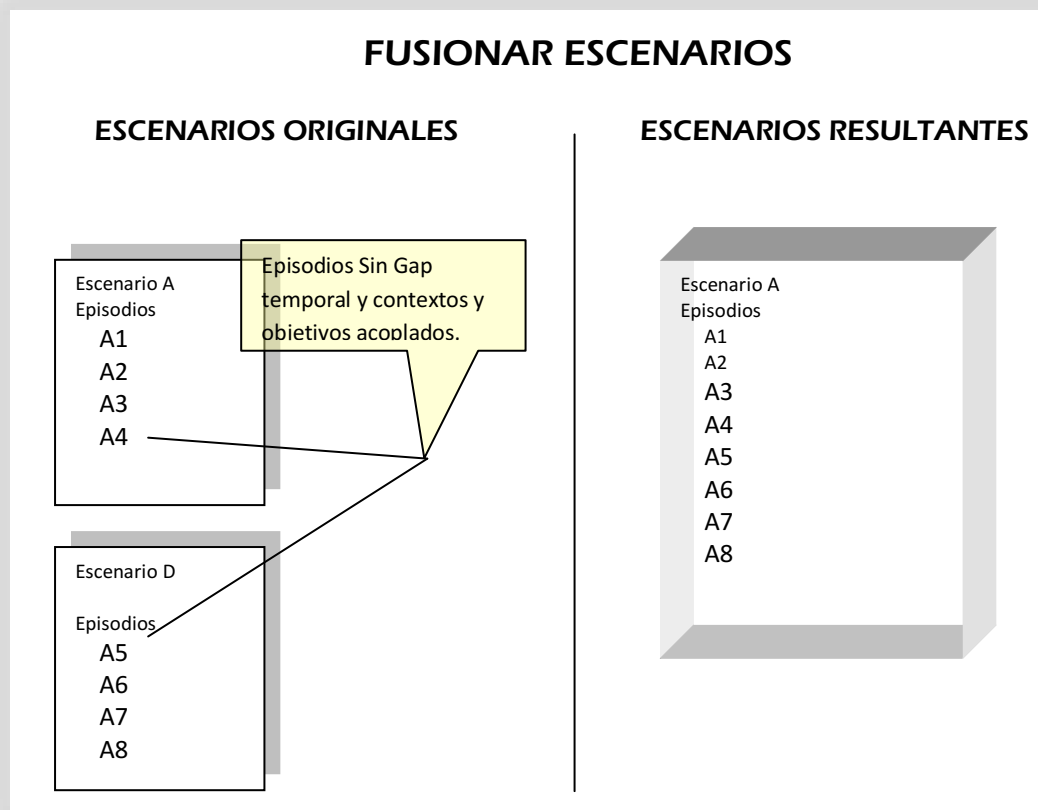


Figura 46 – Operación Fusionar Escenarios

Nuevamente, se debe tener cuidado acerca de la necesidad de incluir un conector del tipo “Y” en el Nombre del Escenario o en su Objetivo. Esta necesidad indicaría que la fusión es inadecuada.

* **Separar** (ver Figura 47): puede aplicarse a un Escenario si éste tiene una brecha temporal entre episodios o cuando se detecta un contexto temporal largo. Un contexto temporal largo puede detectarse por el contexto en sí mismo o por la secuencia de los episodios. Esta operación también puede considerarse cuando se descubre más de una ubicación geográfica en un Escenario. Es decir, el Escenario está representando más de una situación contradiciendo la definición misma de Escenario. La descomposición de un Escenario debería producir Escenarios con un contexto temporal bien acotado. Separar copia a un nuevo Escenario todos los episodios que preceden a una brecha temporal y aquellos episodios que siguen a la brecha temporal se copian a otro nuevo Escenario. El Escenario original se elimina. Las precondiciones del segundo y siguientes nuevos Escenarios reflejan el orden de

precedencia establecido. Luego, separar permite reducir la complejidad y comprender mejor las situaciones.

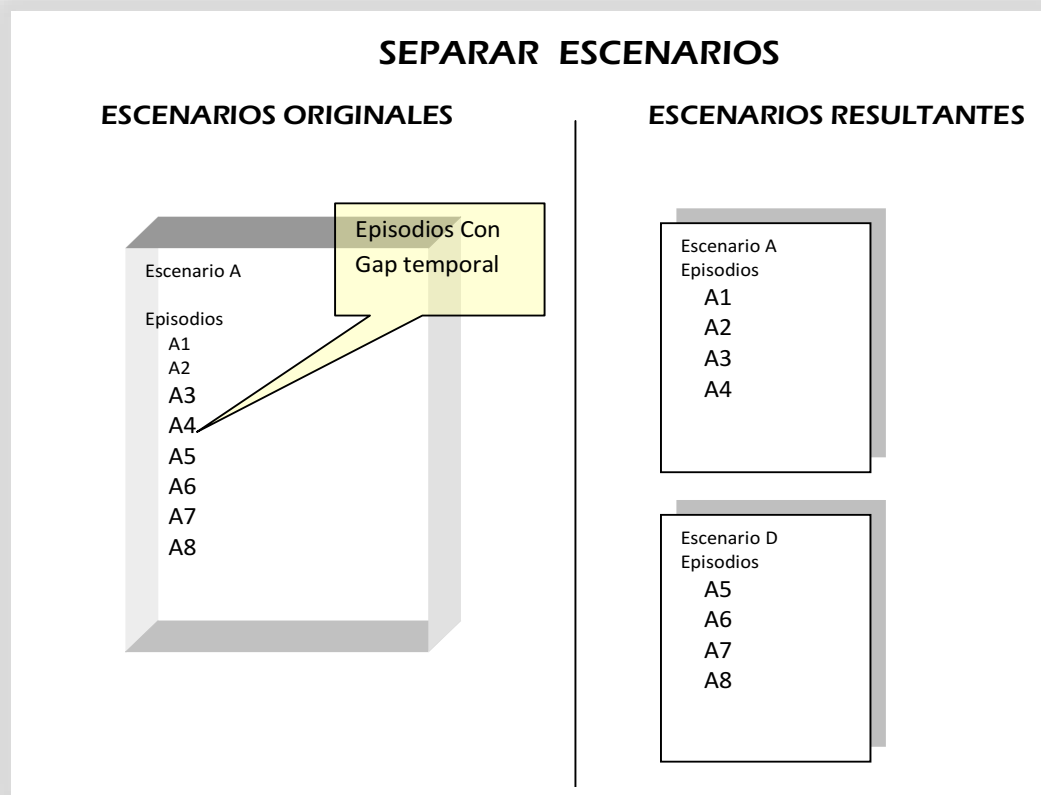


Figura 47 – Operación Separar Escenarios

La Figura 48 muestra la operación factorizar aplicada a dos Escenarios con varios episodios comunes que tiene un objetivo específico; se crea un nuevo Escenario conteniendo estos episodios; los Escenarios originales se re-escriben, reemplazando los episodios comunes por el título del nuevo Escenario. La Figura 49 ejemplifica la operación consolidar aplicada a dos Escenarios superpuestos, produciendo un único Escenario que contiene los episodios comunes mientras los episodios no compartidos se incorporan con una sentencia condicional.

Cuando se aplican las operaciones de composición, el objetivo del Escenario resultante deberá extenderse para abarcar la situación completa, especialmente al fusionar; en caso de consolidar, el objetivo se extiende por generalización. Por el contrario, cuando se aplica descomposición, el objetivo del nuevo Escenario puede ser menos global que el original.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 8 – Escenarios

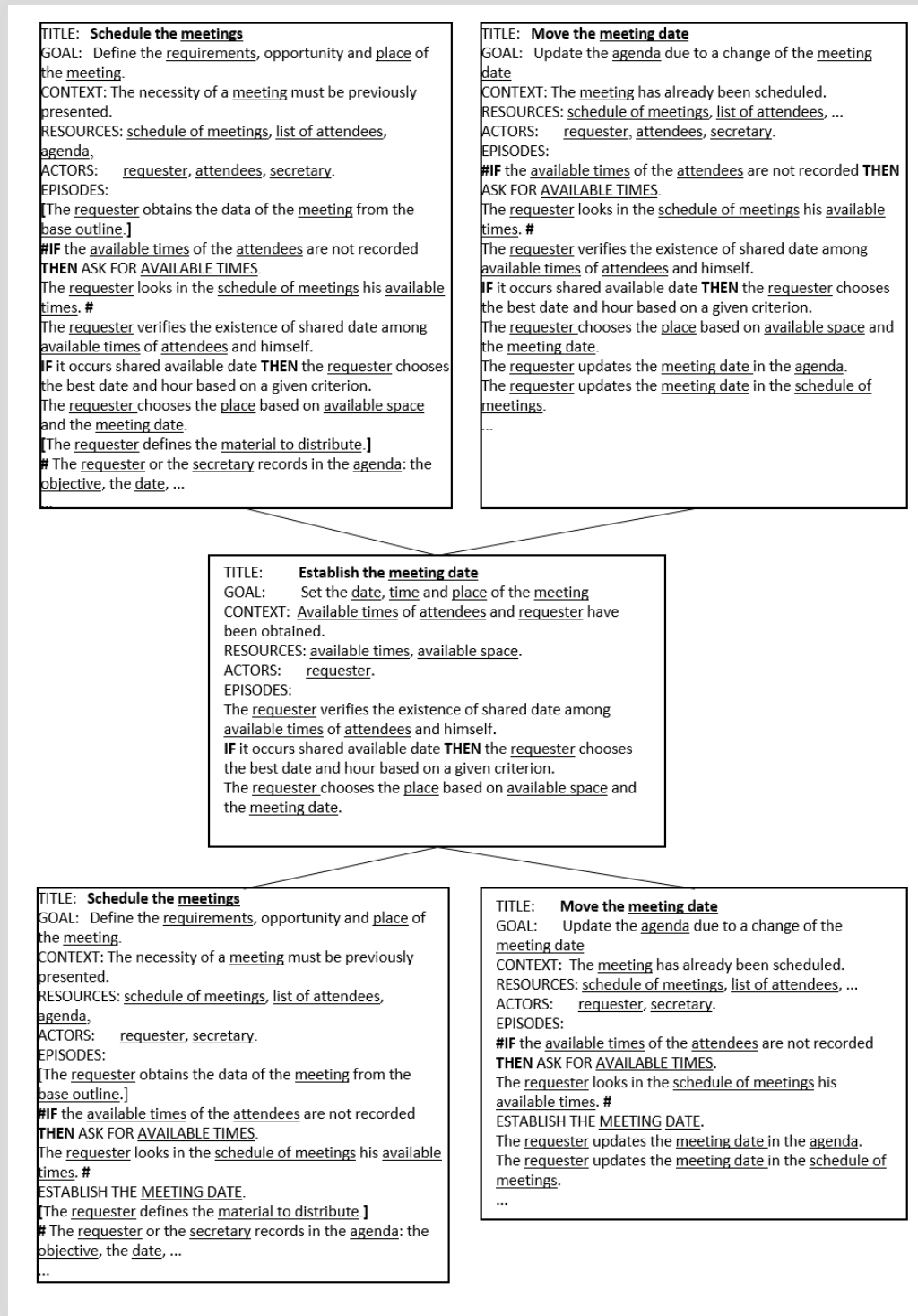


Figura 48 - Ejemplo de la Operación Factorizar Escenarios (Caso SMS)

Dado que consolidar y dividir son operaciones opuestas, después de consolidar, el Escenario resultante puede ser considerado para una división, retornando al estado original y viceversa. Se requieren entonces criterios para decidir la aplicación de esta operación. Consolidar deberá aplicarse si aparecen pocos episodios condicionales en el

Escenario resultante, de lo contrario debe evitarse. Por otro lado, dividir deberá elegirse sólo si varios episodios condicionales se ven afectados por la operación.

Fusionar y separar son operaciones opuestas también; por lo tanto, ambas requieren criterios de decisión sobre la mejor dirección a tomar. La clave es la brecha temporal, tanto respecto a Escenarios como episodios. Una demora entre dos episodios puede considerarse como una brecha temporal sólo si no puede ser abarcada por el contexto temporal del Escenario. En otras palabras, un episodio específico puede ser precedido por una brecha temporal en un Escenario, pero no en otro.

Las operaciones relativas a la dimensión sub-Escenario también requieren criterios de decisión para evitar ciclos infinitos pasando de un estado al opuesto. Luego, aplanar deberá aplicarse cuando los episodios una vez aplanados en el Escenario resultante tengan el mismo nivel de detalle que el resto de los episodios y cuando el sub-Escenario original no represente él mismo una situación, lo que puede confirmarse por la falta de significado de su objetivo. Por el contrario, factorizar deberá usarse sólo si se puede encontrar un objetivo significativo para el nuevo sub-Escenario como consecuencia de haber descubierto una situación específica y real dentro de un Escenario más global.

En el caso particular de los Escenarios Actuales, durante las actividades de *Derivar* y *Describir*, el foco no está en los sub-Escenarios, por lo tanto, en este momento los sub-Escenarios derivados requieren una atención especial para confirmar que ellos son necesarios y debe encontrarse aquellos sub-Escenarios escondidos para clarificar el conjunto de Escenarios. Si esto no es salvado durante la construcción de los EA, este problema puede propagarse a los EF.

Integración de Escenarios

La sub-actividad *Integración* de Escenarios se realiza para construir los *Escenarios Integradores*. En primer lugar, se deben definir las relaciones de jerarquías existentes. Para ello es necesario identificar diferentes relaciones entre los Escenarios para luego integrarlos.

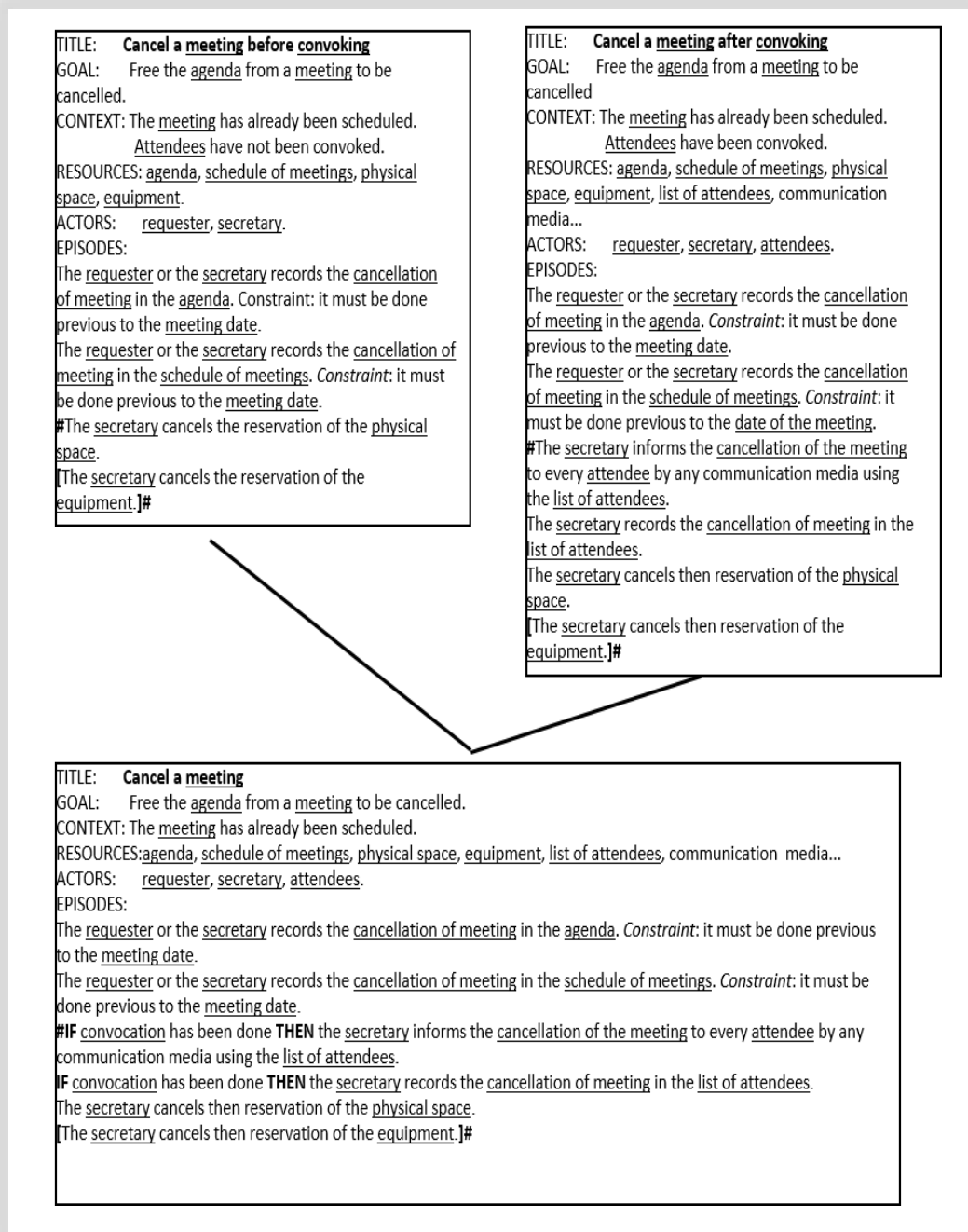


Figura 49 - Ejemplo de la Operación Consolidar Escenarios (Caso SMS)

A continuación, se describen los tipos de relaciones:

- **Relación de Jerarquía** es la que se establece entre un Escenario y un sub-Escenario. Esta relación ocurre naturalmente mientras se describen los Escenarios o cuando se organizan. El comportamiento de un Escenario está descrito en sus episodios, pudiendo ser una acción simple o un sub-Escenario; el último caso ocurre cuando la acción es más compleja o es común a distintos Escenarios. Un Escenario

puede contener un sub-Escenario o más. Un sub-Escenario podrá ser incluido en uno o más Escenarios y puede contener él mismo sub-Escenarios, sumando así niveles de jerarquía. Entonces, se puede definir una jerarquía como un conjunto de Escenarios y sus sub-Escenarios.

➤ **Relación de Superposición** (overlap) es aquella que se establece entre Escenarios cuando existe una parte en común. Esta relación se da principalmente cuando distintos episodios comunes están presentes en diferentes Escenarios, donde además suelen existir actores y recursos comunes.

➤ **Relación de Orden** es la que se establece entre dos jerarquías cuando una precede a la otra. Una jerarquía que se establece antes que otra determina una relación de orden temporal parcial entre ellas. Una jerarquía puede tener cero o más predecesores y cero o más sucesores. Una secuencia se establece cuando una jerarquía está inmediatamente precedida por otra. La segunda puede estar también precedida por una tercera, un gran número de jerarquías pueden estar involucradas en una secuencia comenzando por lo menos con una jerarquía raíz. Una secuencia puede tener más de una jerarquía raíz y también más de una jerarquía final.

➤ **Relación de Excepción** es la que se establece entre un Escenario y los Escenarios que tratan excepciones. Cuando un Escenario tiene una excepción, sus causas están descritas y tratadas en la excepción especificada en el Escenario. Un Escenario puede hacer referencia a una o más excepciones. Una excepción puede ocurrir en diferentes Escenarios.

Si dos Escenarios están conectados por una relación de excepción ellos pueden ponerse juntos en un Escenario Integrador como una secuencia, donde el Escenario se pone primero como un episodio normal y el siguiente (la excepción) es tratado como un episodio condicional.

La relación de superposición no es usada frecuentemente en la actividad *Integrar* Escenarios, esta actividad utiliza esencialmente relaciones de jerarquía y de orden.

Se puede ver en cinco pasos. Los primeros cuatro se refieren principalmente a un análisis sintáctico, pero en el último es semántico. En esta actividad, primero, se agrupan los Escenarios en jerarquías y luego estas se agrupan en secuencia. Finalmente, estas secuencias son usadas para construir los **Escenarios Integradores**. La Figura 50 muestra un ejemplo de un Escenario Integrador. Es de destacar que el Escenario Integrador fue

producido después de la actividad organizar y que en ellos no se llenan los componentes actores y recursos.

➤ **Construir jerarquías de Escenarios** es el primer paso en la actividad de integración. Una jerarquía es un conjunto de Escenarios conectados por una relación jerárquica. Siempre existe un Escenario raíz, el cual no debe estar mencionado como sub-Escenario por ningún otro Escenario. Las jerarquías están identificadas por los Escenarios raíz. Un Escenario aislado que no pertenece a ninguna jerarquía es en sí mismo una jerarquía.

➤ **Detectar Orden Parcial entre Jerarquías** está basado en la comparación entre las precondiciones, recursos o restricciones de una jerarquía y el título, objetivo o episodios de otra jerarquía. Si una precondición de una jerarquía identifica un estado inicial que es satisfecho por otra jerarquía, éstas tienen orden parcial. Lo mismo puede ocurrir con un recurso requerido en una jerarquía que es producido por otra. Además, las restricciones de los recursos y de los episodios pueden ser satisfechas por otra jerarquía. Estas restricciones pueden ser usadas en la comparación cuando el estado final de los recursos o ciertas condiciones deben ser previamente resueltas.

Precondiciones, recursos y restricciones de la jerarquía no se obtienen del Escenario raíz, sino que son parte de la jerarquía. Para establecer las precondiciones de una jerarquía, primero las precondiciones de todos los Escenarios de la jerarquía deben ser unificadas, luego, las precondiciones requeridas por un sub-Escenario y satisfechas por otro sub-Escenario de la jerarquía puede ser removidas de la unión. Los recursos y restricciones de la jerarquía se obtienen de la misma forma. El objetivo de la jerarquía es el mismo del Escenario raíz y la comparación puede parar cuando un orden parcial es encontrado o cuando no es posible detectarlo. Si el orden parcial entre dos jerarquías se establece sólo para cierta condición, ésta es puesta como condición de la relación de orden. Para reducir la búsqueda de la jerarquía proveniente de los recursos o de las restricciones y precondiciones satisfechas es mejor chequear antes el título y el objetivo y chequear luego los episodios. Si una comparación es satisfecha, un orden parcial es encontrado, y se pasa a la siguiente comparación. Si no se satisface, entonces no hay orden parcial. Las comparaciones usadas en este paso son:

- * Precondición del Escenario raíz de una jerarquía contra los títulos de las restantes jerarquías.

- * Restricciones del Escenario raíz de una jerarquía contra los títulos de las restantes jerarquías.
- * Recursos del Escenario raíz de una jerarquía contra los títulos de las restantes jerarquías.
- * Precondición del Escenario raíz de una jerarquía contra los objetivos de las restantes jerarquías.
- * Restricciones del Escenario raíz de una jerarquía contra los objetivos de las restantes jerarquías.
- * Recursos del Escenario raíz de una jerarquía contra los objetivos de las restantes jerarquías
- * Precondición del Escenario raíz de una jerarquía contra los episodios de las restantes jerarquías.
- * Restricciones del Escenario raíz de una jerarquía contra los episodios de las restantes jerarquías.
- * Recursos del Escenario raíz de una jerarquía contra los episodios de las restantes jerarquías.

➤ **Construir las Secuencias de las Jerarquías** comienza con la revisión de las relaciones de orden parcial detectadas entre las jerarquías. Estas se utilizan para clarificar cuál es el conjunto de jerarquías que inmediatamente preceden a otra. Para hacer esto, todos los órdenes parciales derivados por transitividad son eliminados. Es decir, entre las jerarquías A, B y C hay transitividad, si A y B preceden a C pero también A precede a B. Las transitividades no son fáciles de eliminar. Estas pueden ser eliminadas agrupando todos los órdenes parciales de la misma jerarquía a la izquierda de la relación. Los órdenes parciales se ponen en el lado derecho para identificar las transitividades. Nueva información proveniente del UdeD puede ser elicitada cuando se detectan omisiones entre las jerarquías. Esto sucede cuando dos jerarquías en secuencia no coinciden en su relación, o sea, quedan acciones fuera de una u otra jerarquía. Algunas jerarquías pueden ser secuencias en sí mismas.

➤ **Construir el esqueleto de la Integración** es el paso donde sólo construye el componente episodio de los Escenarios Integradores. Primero un Escenario principal es construido y luego, un nivel intermedio de integración puede ser obtenido. Todas las secuencias son puestas en el Escenario Integrador principal marcados con el indicador no-secuencia (#). Si una secuencia está compuesta de más de un Escenario,

se puede crear un fragmento de un nivel intermedio de Escenario y un título nominal es usado en ambos Escenarios, para el integrador principal y para el fragmento de Escenario. Esto es aplicado para cada fragmento existente, creando nuevos fragmentos cuando se detectan sub-secuencias sin secuencia.

➤ **Completar el título, objetivo y contexto de los Escenarios Integradores** es el paso donde es necesario agregar nueva información semántica. Se miran los episodios de Escenario Integrador para completar los restantes componente. Este paso actúa como un ordenamiento de la verificación final, donde el título y el objetivo pueden ser escritos preservando la cohesión, en otras palabras, se descartan las conjunciones. Las descripciones deben ser escritas utilizando los símbolos del LEL.

CONTROLAR LA CALIDAD

Objetivo: conocer los [insumos\(1\)](#) que pueden ser utilizados en [producción](#)

Contexto:

Ubicación Geográfica: [laboratorio](#)

Ubicación Temporal:

Precondición: deben existir [insumos\(1\) asignados](#)

Recursos:

Actores:

Episodios:

- [ANALIZAR UN INSUMO\(1\)](#)
- [EXAMINAR LOS RESULTADOS](#)
- [IDENTIFICAR INSUMOS\(1\) ANALIZADOS](#)
- [GENERAR EL PROTOCOLO](#)
- [#IMPRIMIR Y FIRMAR EL PROTOCOLO](#)
- [ALMACENAR LOS INSUMOS\(1\)#](#)

Excepciones:

Figura 50 - Ejemplo de Escenario Integrador (Caso SCC)

Se ha comprendido que la validación requiere **agrupar los Escenarios** por diferentes aspectos. Es posible que este sea el mayor problema por el cual no se ha podido encontrar un mecanismo mejor. Los Escenarios Integradores proponen comprender el contexto según el flujo de trabajo, pero existen algunos casos particularidades donde este agrupamiento no es suficientemente representativo, como por ejemplo al usar los Escenarios para el diseño, para capacitación, etc.

Capítulo 8.1

Construcción de Escenarios Actuales

Resumen

Los Escenarios Actuales describen el proceso del negocio observable, tal como existe en el momento de comenzar el Proceso de Requisitos. De esta manera, con el LEL y los EA, el ingeniero/a de requisitos puede Comprender el UdeD actual y definir una propuesta/solución adecuada a las necesidades reales del cliente. Por lo tanto, tomando los aspectos comunes descritos en el **Capítulo 8**, en el presente Capítulo se describen todas las particularidades de los *Escenarios Actuales*.

Proceso de construcción de Escenarios Actuales

Tomando como base el **Capítulo 8**, se describe en el presente Capítulo el mecanismo específico para generar el conjunto de Escenarios Actuales (EA). Estos Escenarios modelan el contexto actual, o sea el proceso del negocio que opera simultáneamente a la realización de la Ingeniería de Requisitos.

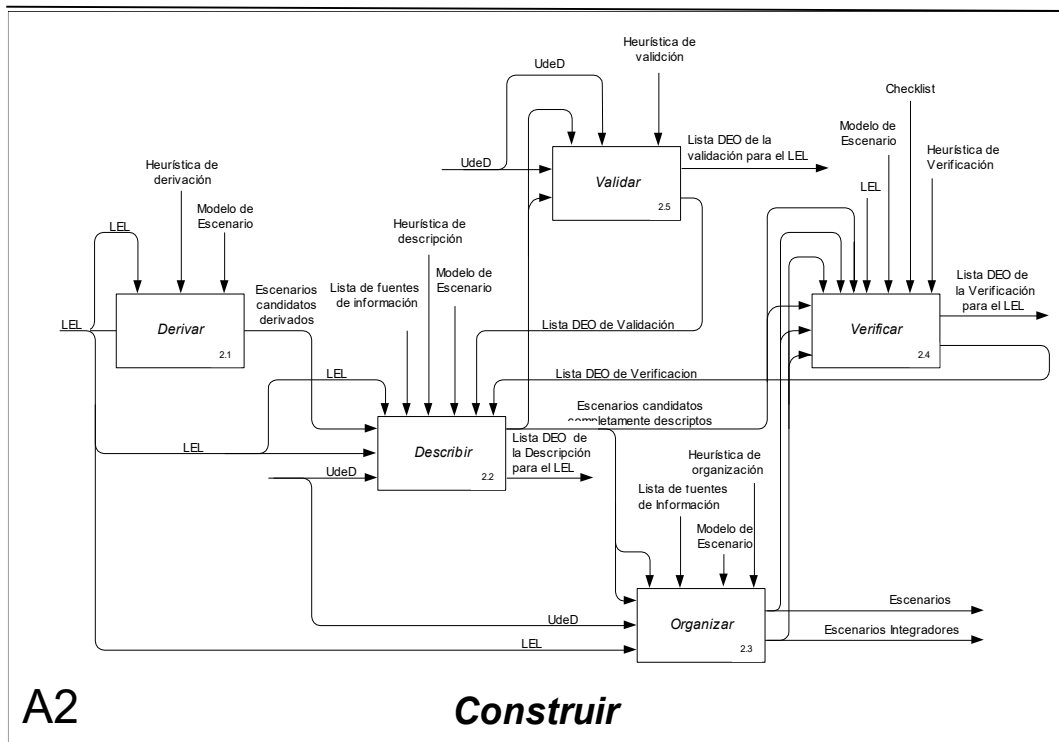


Figura 51– SADT del proceso de Construcción de Escenarios

El proceso de construcción de EA, como puede observarse en la Figura 51, comienza produciendo una primera versión de Escenarios que se producen en la actividad **Derivar**, cuyo origen data de la información existente en el LEL. Una vez generada esta primera versión, se deben consultar diferentes fuentes de información para completarlos, retornando al UdeD tantas veces como sea necesario. Esta actividad se denomina **Describir**. Esos Escenarios que pueden haber sido escritos por diferentes ingeniero/as de requisitos, deben ser **Organizados** para eliminar duplicaciones y mejorar la estructura de los mismos. Una vez organizados deben ser **Verificados** para asegurar su consistencia y finalmente deben ser **Validados** con los clientes-usuarios para asegurar su veracidad. El proceso de construcción se describe en el SADT de la Figura 51. Puede observarse que las actividades *Describir* y *Organizar* no son estrictamente secuenciales. Algunas

actividades se realizan concurrentemente existiendo un mecanismo de reciclaje entre ellas. Así es que la actividad *Verificar* y *Validar* retornan a la actividad *Describir*, donde las correcciones son realizadas en base a la lista de defectos que se genera en la verificación y la información rectificada o nueva que se adquiere en la validación.

Actividad Derivar

Esta actividad se realiza generando los Escenarios candidatos desde la información del LEL, usando el modelo de Escenarios y aplicando la heurística de derivación [Hadad97]. La actividad *Derivar* consta de tres sub-actividades: *Identificar los actores* del UdeD, *Identificar los Escenarios* y por último *Crearlos* usando el LEL.

a) IDENTIFICAR ACTORES

Los símbolos que representan actores en el UdeD son identificados en el LEL. Ellos corresponden a los símbolos clasificados con el tipo sujeto.

b) IDENTIFICAR ESCENARIOS

Cada impacto de los símbolos sujeto representa un posible Escenario candidato y son incorporados a la lista de Escenarios Candidatos. El título del Escenario está compuesto por la acción (verbo) incluido en el impacto, pero representado en infinitivo.

Cuando diferentes actores ejecutan la misma acción, puede producir que dos o más Escenarios de la lista puedan compartir el título; en este caso es recomendable no eliminar ninguno hasta asegurarse que realmente son iguales.

c) CREAR

Aquí se intenta construir los Escenarios con la mayor cantidad posible de información sustraída del LEL, aplicando la heurística de creación de Escenarios. El producto son los Escenarios candidatos derivados.

El contenido de cada impacto es analizado para encontrar en ellos símbolos del LEL del tipo verbo.

⇒ Si existe un símbolo verbo en el impacto:

El objetivo es definido en base al título del Escenario y la noción del verbo.

Los actores y recursos son identificados en la información del símbolo verbo, mirando los símbolos sujetos y objetos existentes en él.

Los episodios son derivados de cada impacto del símbolo verbo.

La precondition del Contexto es derivada de las situaciones que impiden que la acción se realice del símbolo verbo.

⇒ Si no existe un símbolo verbo en el impacto:

Se verifican los símbolos del impacto en cuestión y se toman como posible fuente de información.

El objetivo se define en base al título del Escenario.

Posibles actores y recursos son seleccionados de los símbolos en cuestión leyendo toda su descripción. Los actores son obtenidos de los símbolos sujetos que le dieron origen y los recursos de los símbolos objetos.

No se derivan episodios desde el LEL, ellos serán detectados en los pasos siguientes.

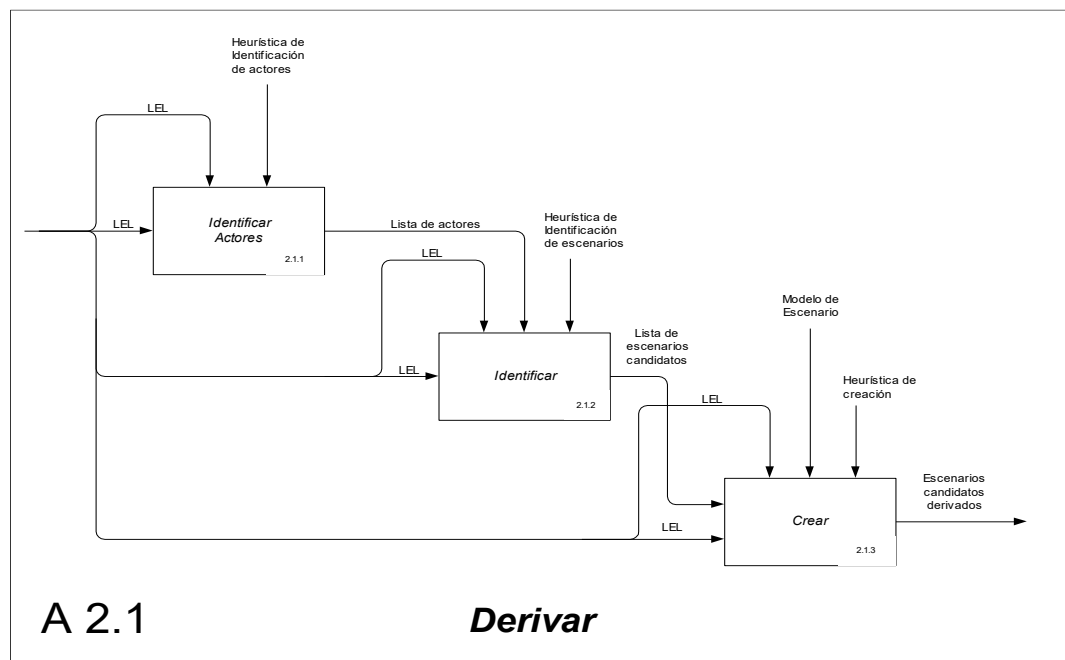


Figura 52 - SADT de la Actividad Derivar Escenarios

En ambos casos, la ubicación geográfica y la ubicación temporal del contexto puede ser extraída de los impactos del símbolo que originó el Escenario (símbolo sujeto). También se puede obtener información relevante de los símbolos relacionados con el símbolo sujeto, o sea se pueden revisar los objetos que se encuentran en otros impactos.

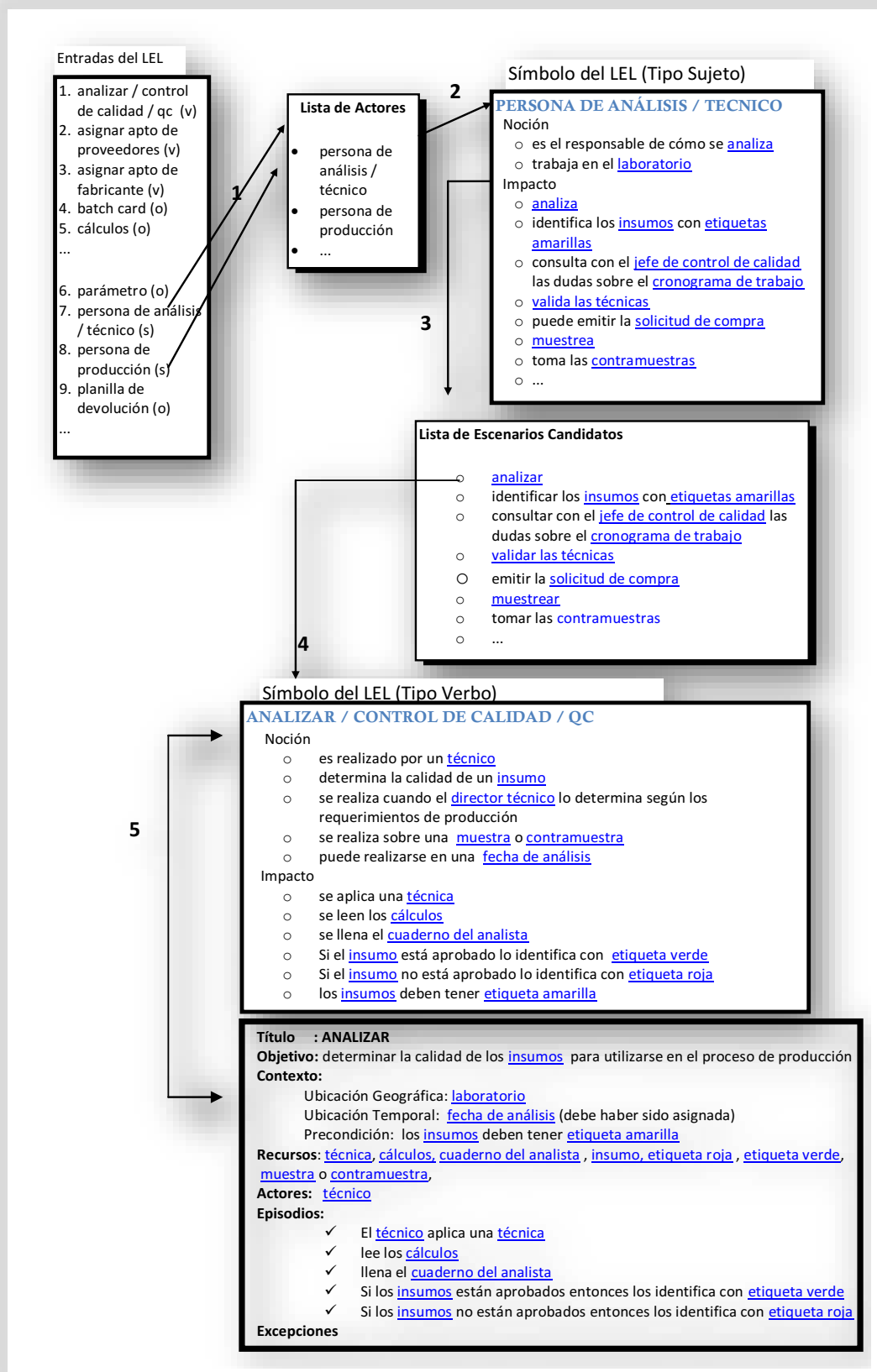


Figura 53 - Ejemplo de Derivar Escenarios (Caso SCC)

Puede observarse que la Figura 52 muestra en un SADT la actividad Derivar y la

Figura 53 la ejemplifica.

La actividad **Derivar** presenta varias debilidades, por lo que no asegura que el conjunto de Escenarios Actuales Candidatos refleje adecuadamente la realidad. Por tal motivo, fue reemplazada por una nueva actividad en el **Capítulo 18**.

Actividad Describir

Es importante que la actividad Describir este planificada, ya que generalmente se realizan entrevistas estructuradas que requieren de una preparación previa. También se puede realizar lectura de documentos. Lo primero que se debe confirmar es la información disponible de la derivación y luego, cuando sea necesario el ingeniero/a de requisitos retornará al UdeD para completarla. Durante esta actividad también se pueden detectar nuevos Escenarios. Es posible que la información recolectada durante la construcción del LEL y no incluida en el léxico, sea ahora utilizada. El resultado es un conjunto de Escenarios completamente descriptos. Puede observarse en la Figura 54 que esta actividad se compone de cuatro sub-actividades: *Completar componentes*, *Crear sub-Escenarios*, *Completar restricciones* y *Completar excepciones*. En el **Capítulo 8** se ha descrito gran parte de la información necesaria para completar cada sub-actividad.

Observando la Figura 54 se puede ver que existe un reciclo desde la actividad Verificar y Validar, representado por las listas DEO (discrepancias, errores y omisiones). Estas listas pueden motivar cambios en las descripciones de los Escenarios. También la actividad Describir puede generar una lista de DEO para el LEL, la cual deberá ser atendida oportunamente.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 8.1 – Construcción de Escenarios Actuales

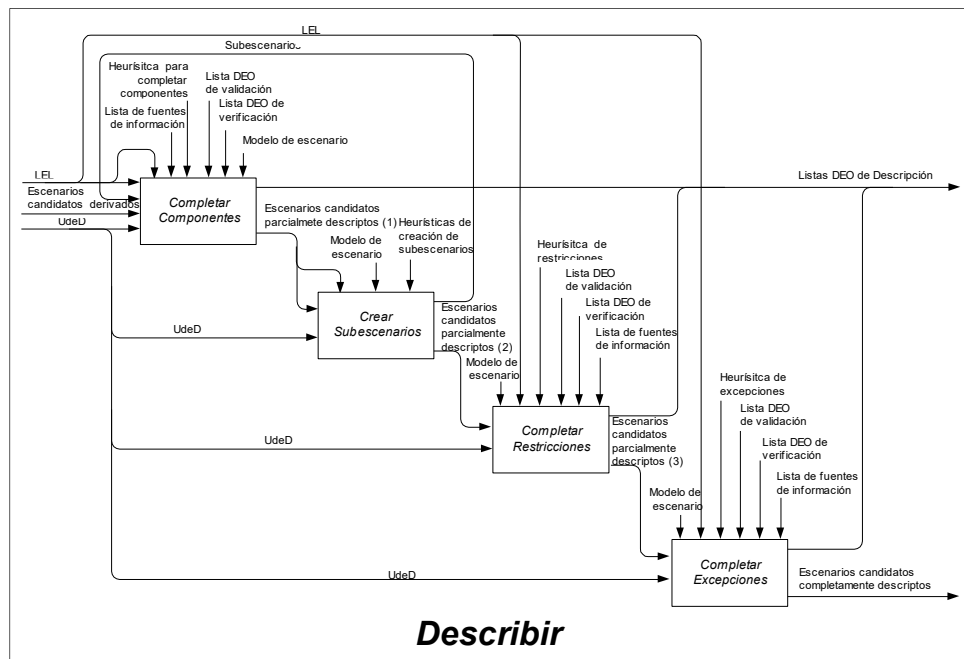


Figura 54 - SADT de la Actividad Describir Escenarios Actuales

La siguiente guía puede ser utilizada para realizar una revisión inicial de las descripciones:

- Son preferibles las sentencias cortas.
- Las sentencias deben ser escritas maximizando el uso de otros símbolos del LEL.
- Los actores y recursos deben ser preferentemente símbolos del LEL del tipo sujeto y objeto respectivamente.
- El objetivo debe ser preciso y concreto.
- Por lo menos uno de los sub componentes del Contexto debe ser llenado
- Todos los recursos enunciados en el componente Recurso deben estar utilizados en los episodios, los recursos triviales deben ser excluidos.
- No se deben incluir los recursos de los sub-Escenarios.
- Los actores del componente Actores deben estar en los episodios y cumplir algún rol directo o indirecto.
- No se deben incluir los actores de los sub-Escenarios.
- El verbo de cada episodio debe ser preciso y concreto, especificando la acción final y descartando posible ambigüedad.

- Se debe tratar de evitar el uso de más de un símbolo verbo por sentencia.
- Los episodios deben realizarse en la ubicación geográfica y con la ubicación temporal descriptos en el Contexto.
- Se debe describir los episodios en tiempo presente
- Evitar usar: podría, puede, debe en los episodios.

A la heurística correspondiente a la actividad *Describir* se incorporó un **Catálogo de Patrones de Escenarios** [Ridao01] (ver Figura 54). De este modo, se aportó la experiencia acerca de situaciones similares acumulada en los patrones. El proceso así modificado, analiza las situaciones desde un punto de vista estructural, asociándolas, completa o parcialmente, con uno o más patrones de un catálogo. Una vez determinado el patrón que más se aproxima a la situación real, se reusa su estructura con el fin de *Derivar* el Escenario más fácil y directamente.

Existen los siguientes patrones: *Producción*, con un sólo actor llevando a cabo una actividad, y *Colaboración*, *Servicio* y *Negociación* con dos o más actores ejecutando una actividad que puede ser cooperativa, en beneficio de uno de los actores, o una secuencia coordinada de acciones, respectivamente. En el caso de la *Negociación*, dependiendo de si la situación comienza, continúa, finaliza o se desarrolla completamente en el Escenario, será *Negociación Inconclusa*, *Etapas de Negociación*, *Fin de Negociación* o *Negociación Terminada*. Cuando la actividad queda inconclusa en un Escenario, es posible que su finalización dependa de varias situaciones futuras. En ese caso, los patrones serán *Negociación Inconclusa con Disparador de Escenarios* y *Etapas de Negociación con Disparador de Escenarios*. Existen situaciones donde se presentan combinaciones de diferentes tipos de episodios, dando lugar a patrones compuestos, como, por ejemplo, *Negociación inconclusa con producción o servicio o colaboración*. Los patrones definidos consisten en un texto guía, que para cada componente del Escenario incluye pautas acerca del contenido que deberá tener dicho componente. Por ejemplo, para el componente episodios, se da una descripción general del tipo de episodios, dando pautas acerca de la cantidad de episodios de cada tipo que deberán aparecer en el Escenario y el orden en que deberán escribirse.

La propuesta de la presente tesis es **Describir los Escenarios**, en primera instancia, tomando toda la información existente, o sea, la literatura del dominio y el LEL. Luego, realizar entrevistas en forma jerárquica, respetando los roles en la organización. Esto permite identificar más claramente los diferentes puntos de vista.

Actividad Organizar

Esta actividad es una de las más complejas y sistematizadas en el proceso de construcción de Escenarios. Su origen es la idea de “Escenarios Integradores”, descripciones “artificiales” con el sólo propósito de hacer que el conjunto de Escenarios sea más comprensible y manejable. Los Escenarios Integradores dan una visión global de la aplicación. Se incorpora una descripción de la actividad en la Figura 55.

Las sub-actividades *Reorganizar* y *Definir Relaciones entre Escenarios e Integrar* se han analizado en el **Capítulo 8**.

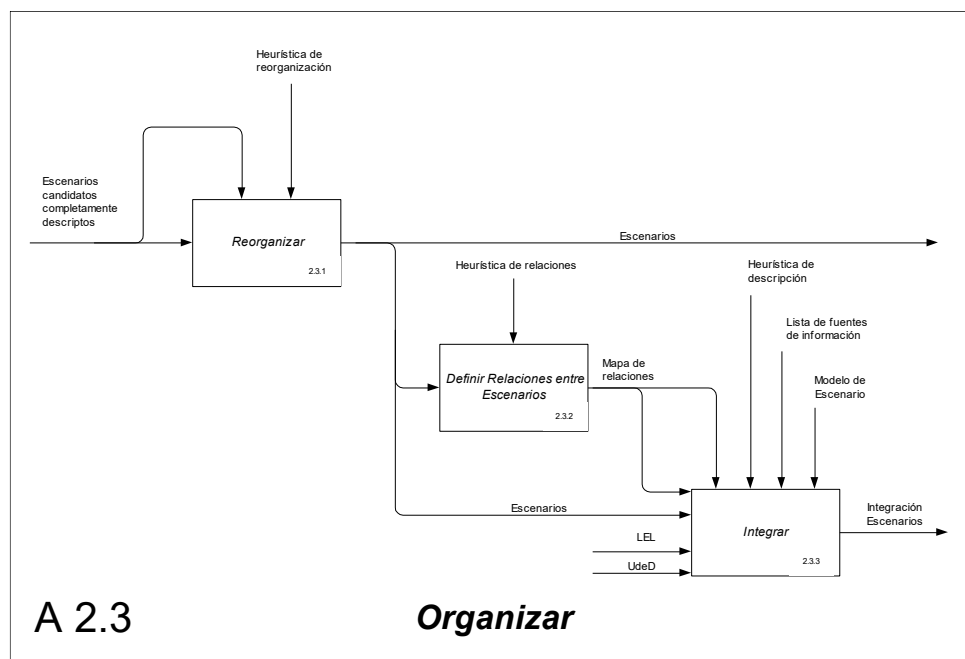


Figura 55 - SADT de la actividad Organizar

Actividad Verificar

La inspección de los Escenarios [Doorn98] [Leite05] (ver Anexo A), se realiza dos veces durante el proceso de construcción de Escenarios: cuando los Escenarios están completos

después de la actividad *Describir* y después de la actividad *Organizar*. Una consecuencia de esta actividad es la producción de dos listas DEO. Una usada en la actividad *Describir* y otra que contiene defectos del LEL.

Pese a la gran flexibilidad con que se pueden escribir los Escenarios, éstos son enormemente ricos en cuanto a su posibilidad de contrastar unos contra otros y contra el LEL, es así que se pueden definir dos tipos de consistencias posibles aplicables a un conjunto de Escenarios:

I. Consistencia interna de Escenarios

I.1 Verificación sintáctica

I.2 Interrelación con el LEL

I.3 Interrelación entre componentes

II. Consistencia entre Escenarios

II.1 Relación entre Escenarios

II.2 Superposición entre los Escenarios

II.3 Cubrimiento del LEL

Cabe aclarar que la Consistencia entre Escenarios fue publicada oportunamente como “Consistencia entre Escenarios de un punto de vista”. En este caso, los “puntos de vista” se refieren a la mirada particular de cada autor. Para evitar cualquier confusión, se ha decidido en la presente tesis, eliminarlo del título ya que el termino es utilizado para marcar la mirada particular de cada usuario.

En la **Figura 56** se esquematizan los tipos de inspección que se realizan sobre los Escenarios. La *Consistencia interna de Escenarios* y la *Consistencia entre Escenarios*.

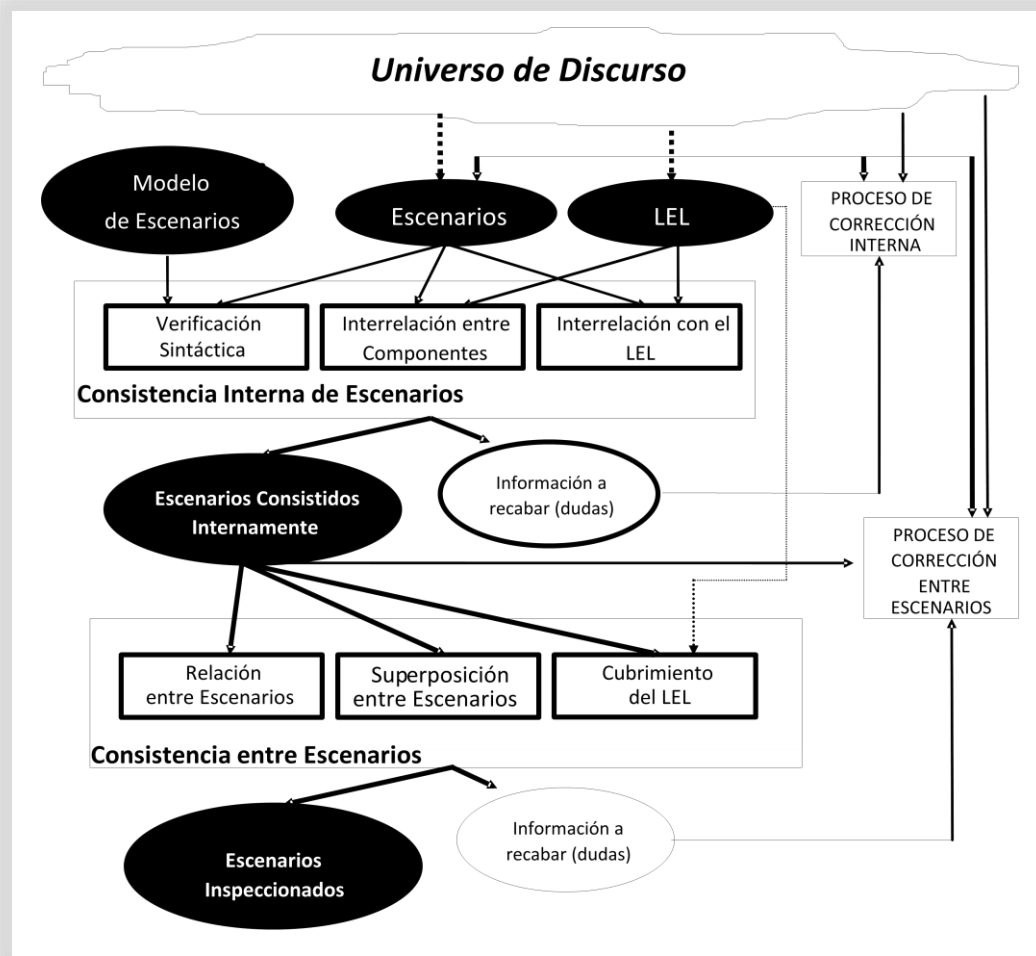


Figura 56- Verificación de Escenarios

La primera verificación, *Consistencia Interna de Escenarios*, se ocupa de analizar cada Escenario en forma individual. El proceso de consistencia interna de Escenarios utiliza para su desarrollo los siguientes elementos adicionales: el modelo de Escenarios y el LEL. Al finalizar el proceso, se obtiene un conjunto de Escenarios mejorados y un grupo de dudas que se deben resolver en el UdeD, a partir de las cuales se genera una nueva versión de Escenarios y eventualmente un nuevo proceso de consistencia interna de Escenarios. Como se muestra en la Figura 56, el análisis de cada Escenario comprende la **verificación sintáctica**, la **interrelación con el LEL** y la **interrelación de los componentes**. La verificación sintáctica comprueba que cada componente haya sido correctamente escrito, por ejemplo, el título puede ser: frase|([sujeto]+verbo+predicado). La interrelación con el LEL verifica que los símbolos del LEL estén correctamente utilizados y que toda frase destacada como símbolo del LEL efectivamente sea parte del mismo. La interrelación de componentes verifica entre otras cosas, que todos los actores de la lista de actores

cumplan un rol en algún episodio y que todo sujeto de los episodios esté en la lista de actores.

La **verificación sintáctica** se basa en la existencia de formularios y heurísticas para el llenado de los mismos. En este caso particular se utilizan tres formularios, con los siguientes propósitos:

- * Verificación sintáctica de título, objetivo, contexto, actores, recursos y excepciones.
- * Verificación sintáctica de episodios.
- * Verificación sintáctica de indicadores de tipos de episodios.

VERIFICACIÓN SINTÁCTICA DE COMPONENTES

- 1° Para cada componente del Escenario verificar la sintaxis utilizada.
- 2° Registrar los elementos faltantes en el casillero correspondiente.
- 3° Registrar los elementos sobrantes en el casillero correspondiente.
- 4° Asegurarse que corresponde eliminar todos los sobrantes detectados.
- 5° Registrar en mejoras a realizar todas las eliminaciones confirmadas.
- 6° Registrar todos los faltantes como información a recabar.

Cuadro 1: Heurística correspondiente a la verificación sintáctica de componentes.

En el Cuadro 1 se presenta una versión resumida de la heurística asociada con el primero de los formularios.

La verificación de la **interrelación de los Escenarios con el LEL** se realiza mediante la utilización de un formulario con su correspondiente heurística. En el Cuadro 2 se presenta una versión resumida de la heurística asociada con esta verificación.

La **interrelación de componentes** se ocupa de verificar que los actores y los recursos estén correctamente enumerados en los componentes correspondientes y a su vez que los mismos tengan participación en el resto del Escenario. Para lograr este efecto se utilizan tres formularios con el siguiente propósito:

- * Verificación de ocurrencia de actores en el Escenario.
- * Verificación de ocurrencia de recursos en el Escenario.
- * Comprobación pragmática de episodios.

IDENTIFICACIÓN DE SÍMBOLOS DEL LEL

- 1° Transcribir al formulario todas las palabras o frases marcadas como símbolos del LEL.
- 2° Verificar la pertenencia al LEL de cada palabra o frase transcrita.
- 3° Registrar en la columna LEL el tipo de símbolo o N si no existe.
- 4° Revisar el Escenario buscando usos sin marca de pertenencia al LEL.
- 5° Registrar en la columna SIN NEGRITA el componente donde se detectó.
- 6° Revisar el Escenario buscando símbolos usados con significado diferente.
- 7° Registrar en la columna MAL USO el componente en que se detectó.
- 8° Para las líneas con tipo N, asegurarse que no deben ser símbolos del LEL.
- 9° Para las líneas con indicación SIN NEGRITA, asegurarse que son símbolos del LEL.

Cuadro 2: Heurística correspondiente a la identificación de símbolos del LEL.

Los dos primeros formularios permiten comprobar si todos los actores y recursos enumerados tienen participación en los restantes componentes del Escenario, tales como título, objetivo, contexto y episodios. Además, se verifica si toda entidad que merece ser registrada como actor o recurso por su participación en el resto del Escenario se encuentra incluida en el componente correspondiente. El tercer formulario tiene por objetivo detectar episodios que no contienen ni actores ni recursos ni símbolos del LEL, esta situación es de por sí muy sospechosa y probablemente permita detectar alguna omisión en el componente Actores o en el componente Recursos o en ambos. En el Cuadro 3 se presenta una versión resumida de la heurística asociada con el primero de los formularios.

La segunda verificación, *Consistencia entre Escenarios*, permite verificar las **relaciones entre los Escenarios, analizar la superposición** entre los mismos y estudiar el grado de **cubrimiento del LEL**.

VERIFICACIÓN DE OCURRENCIA DE ACTORES EN UN ESCENARIO

- 1º Registrar todo candidato a actor presente en el título, en el objetivo, en el contexto o en los episodios.
- 2º Registrar la cantidad de ocurrencias en cada componente mencionado.
- 3º Registrar si el candidato a actor está presente o no en la lista de actores.
- 4º Registrar en la columna LEL el tipo de símbolo o N si no existe.
- 5º Recorrer la lista de actores y transcribir aquellos no incluidos aún.
- 6º Asegurarse que los candidatos a actores fueron seleccionados apropiadamente.
- 7º Asegurarse que los actores fueron incluidos en la lista apropiadamente.
- 8º Asegurarse que los actores con N en la columna LEL, no deben ser símbolos del LEL.
- 9º Registrar en mejoras a realizar todos los cambios confirmados.
- 10º Registrar en información a recabar todo aspecto no discernible.

Este proceso, como se muestra en la Figura 56, se realiza luego de finalizado el proceso de Consistencia interna de Escenarios. Al finalizar el proceso, se obtiene un conjunto de Escenarios mejorados y un grupo de dudas que se deben resolver en el macrosistema, a partir de las cuales se genera una nueva versión de Escenarios y eventualmente un nuevo proceso de consistencia interna de Escenarios o de consistencia entre Escenarios.

La verificación de la **relación entre los Escenarios** se realiza mediante la utilización de cuatro formularios con el siguiente propósito:

- * Comprobación de los Escenarios integradores.
- * Comprobación de la existencia de sub-Escenarios.
- * Verificación de la consistencia de precondiciones en el contexto.
- * Verificación de las precondiciones de los sub-Escenarios.

VERIFICACIÓN DE LAS PRECONDICIONES DE LOS SUB-ESCENARIOS

- 1º Identificar Escenarios que contengan Sub-Escenarios.
- 2º Registrar cada precondition del Sub-Escenario.
- 3º Verificar que la precondition lo sea también del Escenario padre.
- 4º Verificar si los episodios del Escenario padre previos al episodio Sub-Escenario satisfacen la precondition.
- 5º Identificar como omisiones las precondiciones que no cumplen ni 2º ni 3º.
- 6º Determinar si es discernible cada omisión a partir del resto de la información.
- 7º Determinar la posible existencia de errores en la precondition del Sub-Escenario o del Escenario padre.
- 8º Registrar en mejoras a realizar todas las omisiones resueltas.
- 9º Registrar en información a recabar toda omisión pendiente de resolución.

Cuadro 4: Heurística correspondiente a la verificación de precondiciones en sub-Escenarios

El primer formulario permite comprobar el nivel de jerarquía de los episodios que componen los Escenarios Integradores. El segundo tiene por objetivo detectar la inexistencia de sub-Escenarios identificados como tales. El tercero permite detectar errores o discrepancias en las precondiciones de los Escenarios u omisión de información para satisfacer dichas precondiciones. El cuarto formulario permite detectar errores en las precondiciones de los sub-Escenarios u omisión de información en los Escenarios que contienen dichos sub-Escenarios. En el Cuadro 4 se presenta una versión resumida de la heurística asociada con la cuarta verificación.

La verificación de la **superposición entre los Escenarios** se realiza mediante la utilización de un formulario con su correspondiente heurística. En el Cuadro 5 se presenta una versión resumida de la heurística asociada con esta verificación.

La verificación del grado de **cubrimiento del LEL** se realiza mediante la utilización de cuatro formularios, con el siguiente propósito:

- * Verificación de la ocurrencia de actores.
- * Verificación de la ocurrencia de recursos.
- * Identificación de símbolos del LEL no utilizados.
- * Contrastación de impactos del LEL con Escenarios y episodios.

VERIFICACIÓN DE SUPERPOSICIONES ENTRE ESCENARIOS

- 1º Determinar el conjunto de Escenarios excluyendo los Escenarios Integradores.
- 2º Confeccionar la lista de combinaciones de Escenarios tomados de a dos.
- 3º Para cada par, calcular el índice de cercanía.
- 4º Seleccionar todos los pares cuyo índice supere un valor prefijado.
- 5º Para los pares seleccionados analizar los objetivos y los episodios.
- 6º Seleccionar todos los pares con objetivos o episodios comunes.
- 7º Confirmar y registrar las superposiciones detectadas.
- 8º Registrar en mejoras a realizar los reordenamientos de Escenarios detectados.
- 9º Registrar en información a recabar las cuestiones no discernibles.

Cuadro 5: Heurística correspondiente a la verificación de superposiciones entre Escenarios

Los dos primeros formularios permiten detectar actores y recursos no incluidos en el LEL y el nivel de participación de los mismos en los Escenarios. El tercero permite específicamente determinar el grado de cubrimiento del LEL por parte del conjunto de Escenarios. El cuarto tiene por objetivo verificar que todos los impactos de los símbolos pertenecientes a la clasificación sujeto del LEL estén representados en algún Escenario o episodio. En el Cuadro 6 se presenta una versión resumida de la heurística asociada con la cuarta verificación.

CONTRASTACIÓN DE IMPACTOS DEL LEL CON ESCENARIOS Y EPISODIOS

- 1º Registrar todos los símbolos del LEL pertenecientes a la clasificación Sujeto.
- 2º Registrar todos los impactos de los símbolos seleccionados.
- 3º Identificar en los Escenarios todas las menciones a los símbolos seleccionados.
- 4º Revisar cada mención buscando detectar qué impacto está siendo considerado.
- 5º Registrar los episodios o Escenarios que se relacionan con impactos del LEL.
- 6º Registrar como omisiones los impactos no relacionados con los Escenarios.
- 7º Determinar si es discernible cada omisión, a partir del resto de la información.
- 6º Registrar en mejoras a realizar todas las omisiones resueltas.
- 7º Registrar en información a recabar toda omisión pendiente de resolución.

Cuadro 6: Heurística correspondiente a la Contrastación de impactos del LEL con Escenarios y episodios.

Actividad Validar

Los Escenarios son validados con el cliente-usuario realizando entrevistas no estructuradas. Durante la validación, el componente dudas (que puede ser agregado durante la actividad *Describir* y que luego debe ser eliminado) debe ser considerado con especial atención. Es importante reiterar que, a pesar de que se trata de una descripción estructurada, los Escenarios están escritos en lenguaje natural, empleando el vocabulario propio del cliente-usuario y describiendo situaciones específicas en la que ellos están involucrados.

La actividad *Validar* se realiza para detectar DEO en los Escenarios. Los errores son detectados principalmente leyendo el Escenario al cliente-usuario, algunas omisiones aparecen durante la lectura, pero otros preguntando sobre información faltante o detalles del Escenario. Las discrepancias pueden aparecer durante las entrevistas cuando existen diferencias entre lo que los clientes / usuarios ven acerca de la información o cuando se analiza la información obtenida.

El principal objetivo de la validación es confirmar la información elicitada y detectar DEO, y un efecto de este proceso es elicitación de nueva información. La validación de un conjunto de Escenarios, después del proceso de verificación, puede confirmar que las situaciones del UdeD ocurren, o sea que han sido registradas con la percepción de los actores (cliente-usuario) del UdeD durante la lectura o discusión de los Escenarios.

Capítulo 8.2

Construcción de Escenarios Futuros

Resumen

Si bien los Escenarios son el modelo central del Proceso de Requisitos, los Escenarios Futuros son los más representativos. Estos Escenarios describen la propuesta para el nuevo Sistema de Software y tienen empotrados los requisitos del software. Por lo tanto, tomando los aspectos comunes descritos en el **Capítulo 8**, en el presente Capítulo se describen todas las particularidades de los *Escenarios Futuros*.

Los Escenarios permiten representar situaciones del contexto, su principal diferencia es la temporal, o sea que los EA representan situaciones observables del contexto, aquellas que existen en el proceso del negocio. Mientras que los futuros describen situaciones que aún no existen. Esta complejidad es uno de los motivos por los cuales se debe tener un cuidado especial al construirlos. Además, estos EF contienen los requisitos del software, lo que hace que su relevancia sea determinante en la construcción del software, haciendo indispensable asegurar su construcción y validación. Es indispensable comprender que toda decisión que se tome en este momento será mandatorio hasta tener el Sistema de Software en producción. Sólo a través de la gestión de requisitos se podrán modificar. Por lo tanto, los EF son el cimiento para la construcción del Sistema de Software y todos los artefactos que se construyan desde este momento en adelante, deberán ser validados con la ERS que se obtiene de los EF. Por lo tanto, todo aquello que se omita, este incompleto o ambiguo, se prorrogará en todos los artefactos que se construyan, llegando indefectiblemente al producto final.

Evolución de los Escenarios

El concepto de evolución utilizado en esta sección se ciñe a la descripción del **Capítulo 3**, donde se utiliza “evolución aparente” para describir la transformación que se produce dentro de la IR. En el contexto del presente Capítulo, el termino más apropiado sería “evolución planificada” ya que refleja la Proyección del UdeD futuro. Esta evolución requiere necesariamente la aplicación de mecanismos de generalización o de abstracción en algunos momentos y de detalle o especialización en otros. Cuando por primera vez, el ingeniero/a de requisitos, se enfrenta a una situación del UdeD casi siempre observa actores realizando actividades en un cierto contexto. Habitualmente, el propósito de las mismas no es suficientemente explícito para poder enmarcarlo claramente en el UdeD. Es aquí donde interviene la abstracción ya que el ingeniero/a de requisitos debe precisar el objetivo durante la construcción del Escenario Actual que describe la situación. Al construir el Escenario Futuro debe imaginarse un futuro posible, haciendo jugar en un rol preponderante al nuevo Sistema de Software. Es más, si por cada Escenario Actual resulta posible más de un Escenario Futuro equivalente (diferentes propuestas), el ingeniero/a de requisitos junto con el cliente deben seleccionar el mejor.

Es importante resaltar que la situación futura se analiza desde la perspectiva de QUÉ debe hacerse y CÓMO el sistema interactúa con el resto del UdeD (ver Figura 57). Debe notarse especialmente que el punto de vista del CÓMO está aplicado al contexto del sistema y no al sistema mismo. La definición de CÓMO el Sistema de Software proveerá los servicios que se definen en esta etapa se realiza con posterioridad en el momento del diseño del software.

Resumiendo, cuando el Sistema de Software es el instrumento para llevar a cabo un meta-objetivo (Objetivo General del Sistema), la relación entre los objetivos del Escenario Actual y del Escenario Futuro son potencialmente diferentes y el desafío es definir el objetivo del Escenario Futuro utilizando el objetivo del Escenario Actual y el meta-objetivo del problema. Nuevamente, se indica que se está mostrando una evolución de un Escenario Actual en un Escenario Futuro. Si la transformación fuera más compleja, el apareo de los objetivos se cumple sobre un conjunto de Escenarios y no sobre cada Escenario individual.

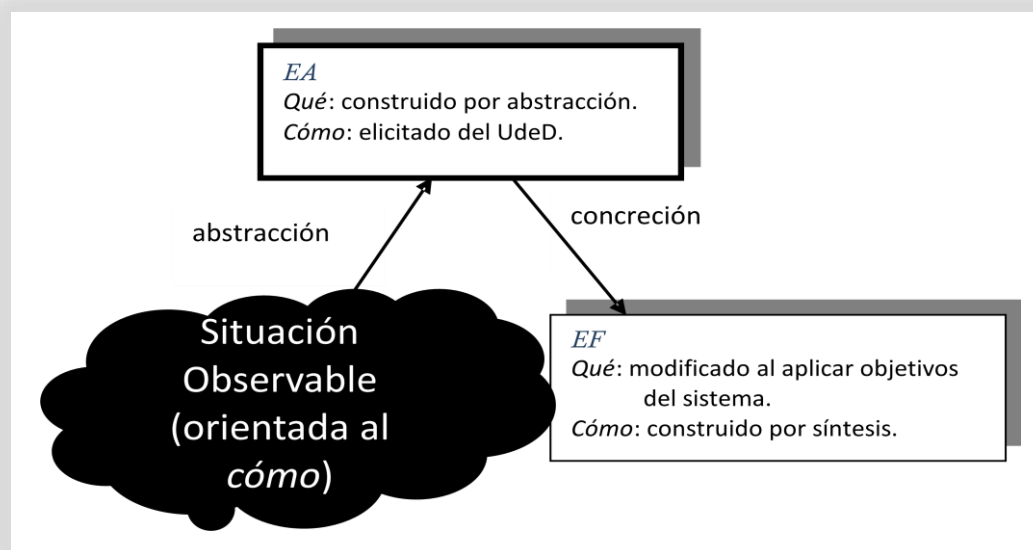


Figura 57– Abstracción y concreción en la construcción de Escenarios Futuros

Proceso de construcción de Escenarios Futuros

Durante la construcción de los EF la relación con el Objetivo General del Sistema se torna bidireccional dado que el Objetivo General del Sistema afecta a cómo los Escenarios Futuros se construyen y cuando los Escenarios Futuros están terminados se comprende mejor el Objetivo General del Sistema.

Como primer paso para construir los EF, el ingeniero/a de requisitos debe plantearse el siguiente interrogante:

¿Qué grado de cambio se espera en el proceso del negocio?

Al analizar el Objetivo General del Sistema y de ser posible, los sub-objetivos, en el contexto en estudio, el ingeniero/a de requisitos está en condiciones de conocer la respuesta. Obviamente, como se mencionó en el primer párrafo, durante la construcción de los EF se conoce más acerca del Objetivo General del Sistema, lo que podría modificar esta primera impresión. La pregunta sirve fundamentalmente, para saber cómo debe prepararse el ingeniero/a de requisitos para afrontar mejor el desafío que está por comenzar. Cuando el proyecto de software está envuelto en grandes cambios, siendo este un factor clave del proyecto, se espera que muchas situaciones que ocurren actualmente, se modifiquen durante la implantación del artefacto de software. Como consecuencia, también se espera que los Escenarios Futuros tengan poco apareo con los Escenarios Actuales. Además, el proceso de construcción de los Escenarios Futuros debe ser consciente de esta propiedad y ser consecuente con la necesidad de proporcionar un ambiente adecuado para encarar todos los detalles necesarios para proyectar los objetivos ya disponibles del software. En el otro extremo, algunos proyectos de software tienen el compromiso de mantener el proceso del negocio tan inalterado como sea posible. Esto puede deberse a varias razones, tales como regulaciones gubernamentales, políticas de la casa matriz o meramente para conservar una manera exitosa de llevar el negocio. Se espera entonces que haya mucho apareo entre los Escenarios Actuales y los Escenarios Futuros. Luego, el proceso de construcción de Escenarios Futuros debe adecuarse al manejo apropiado de esta peculiaridad. Los Escenarios Futuros de proyectos de software inmersos en un alto grado de cambio en el proceso del negocio deben construirse en un modo orientado a los objetivos, prestando menos atención a los aspectos procedurales actuales. Por el contrario, proyectos de software con un marco de trabajo de pocos cambios debe construir sus Escenarios Futuros usando un enfoque dirigido por consideraciones procedurales. Sin embargo, muchos proyectos de software se ubican entre ambos extremos. Para ellos, es necesario combinar ambos enfoques según sus características.

Tomando en cuenta lo recién mencionado, se puede analizar el proceso de construcción de EF [Doorn02] que se describe en la Figura 58.

Se puede observar una gran similitud entre las actividades del proceso de construcción de EA con el presentado en la Figura 58. Ambos utilizan el mismo modelo y, por lo tanto, como se mencionó en el **Capítulo 8**, tienen algunas actividades que se ejecutan de la misma manera. Otras actividades son muy parecidas y hay un grupo específicas del UdeD futuro. Entre estas últimas se encuentra la primera actividad del proceso. Puede observarse que la actividad *Derivar* ya no existe, en este proceso se realiza una transformación del UdeD actual al futuro, utilizando el Objetivo General del Sistema y la información de los EA, siempre con un ancla en el LEL. Se puede observar también, como se describió en el **Capítulo 8**, que las actividades **Describir** (con nuevas sub-actividades como la Negociación) y **Organizar** tienen mucha similitud, mientras que las actividades **Verificar** y **Validar** se deben adaptar al punto de vista del UdeD futuro.

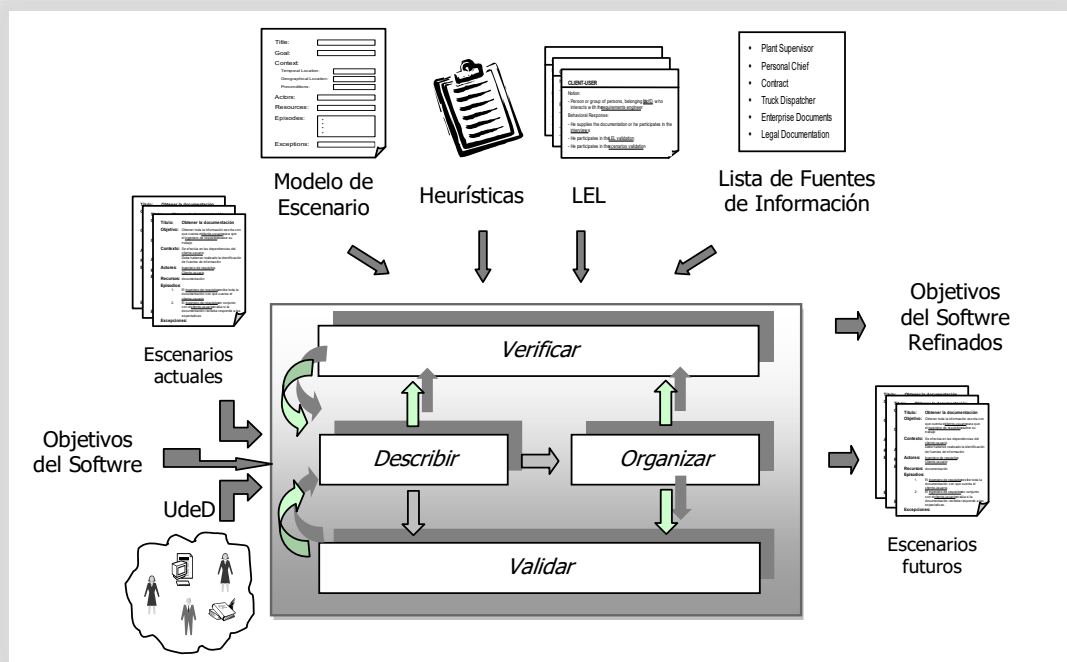


Figura 58 - Proceso de Construcción de Escenarios Futuros.

Una diferencia importante entre los EA y los EF, radica en la aparición de un nuevo actor que representa al nuevo Sistema de Software. En la Figura 59 se puede observar un EF donde participa el actor Sistema de Software. A cargo de este actor están todas las acciones que constituyen los servicios o restricciones del software a construir.

Como se mencionó en el **Capítulo 8**, los Escenarios contextualizan los servicios del nuevo Sistema de Software, esto les permite a los clientes-usuarios comprender el proceso del negocio futuro, tal como sucederá con el Sistema de Software funcionamiento.

A diferencia de la primera parte de la construcción de los EA cuyo origen es fundamentalmente el LEL, los EF se originan en el análisis de la descomposición de los EA Integradores. Cabe destacar que el esfuerzo para construir los EF es menor debido a que se parte de un conjunto de Escenarios (EA) previamente organizados y verificados.

Existen tres enfoques para construir los EF:

- Procedural,
- por Objetivos e
- Híbrido.

Es necesario remarcar que se debe seleccionar la orientación de acuerdo al nivel de cambio que se espera en el proceso del negocio futuro. A mayor cantidad de cambios la construcción de los EF se la debe pensar por Objetivos, mientras que, en un contexto con pocos cambios, como puede ser una informatización, los EF tendrán serán muy parecidos a los EA y en este caso un proceso Procedural es el más indicado. Cuando existe una solución que modifica el proceso del negocio existente (enfoque por Objetivos) pero otra parte del contexto se mantiene inalterable o con pocos cambios (enfoque Procedural), por lo tanto, es necesario utilizar un enfoque Híbrido, o sea aplicar uno u otro de acuerdo a cada caso particular.

ASIGNAR FECHA DE ANÁLISIS

Objetivo: determinar el momento del [análisis](#) de cada [insumo\(1\)](#)

Contexto:

Ubicación Geográfica: [Oficina técnica](#)

Ubicación Temporal:

Precondición: debe haber un [insumo\(1\) pendiente](#) sin [fecha de análisis](#)

Recursos: [cronograma de trabajo](#), [insumo\(1\)](#)

Actores: [jefe de control de calidad](#), [Sistema](#)

Episodios:

1. El [Sistema](#) muestra el [cronograma de trabajo](#)
2. El [Sistema](#) muestra los [insumos\(1\) pendientes](#) sin [fecha de análisis](#) ordenados por [prioridad de análisis](#)
3. El [jefe de control de calidad](#) selecciona una fecha y un [insumo\(1\)](#)
4. El [Sistema](#) guarda en el [cronograma de trabajo](#) la fecha y el [insumo\(1\)](#) seleccionados
5. El [Sistema](#) asigna el estado [asignado](#) a ese [insumo\(1\)](#)

Excepciones:

Cuando existen conflictos entre diferentes [insumos\(1\)](#) con la misma [prioridad de análisis](#), el [jefe de control de calidad](#) determina el orden según su criterio.

Figura 59 – Ejemplo de Escenarios Futuro (Caso SCC)

Unos de los aportes de la presente tesis es la identificación de la pérdida de cubrimiento que tiene el LEL en los EF. En este contexto, se presenta en el **Capítulo 15** la construcción del LEL_R que evoluciona el léxico durante la IR. Por lo tanto, en la Figura 59 se deben reemplazar los hipervínculos hacia el LEL, por vínculos hacia el LEL_R para evitar incorporar ambigüedad en la descripción de los EF.

Proceso orientado a lo Procedural

La hipótesis principal cuando se construyen EF que representan cambios menores del contexto es que hay un mapeo casi completo de los EA a los EF. También se supone que el objetivo del Escenario es invariante, es decir, puede copiarse del Escenario Actual al Escenario Futuro tal como es. Obviamente, ésta es una situación límite, los proyectos reales introducen algunos cambios y no siempre cada Escenario Actual tiene su correspondiente Escenario Futuro con exactamente el mismo objetivo. Sin embargo, éste es el punto de partida razonable en muchos casos. Luego, el Escenario Futuro se construye enfocándose en el componente episodio del Escenario Actual. Cada episodio debe analizarse en el marco de los objetivos del proyecto de software para determinar si necesita ser modificado y cómo. Una vez que todos los episodios han sido estudiados, el resto de los componentes del Escenario debe observarse para propagar los cambios introducidos en los episodios. Pueden necesitarse nuevos recursos o algunos recursos existentes pueden tornarse obsoletos. Si el Escenario Futuro está dentro del alcance del proyecto de software, un nuevo actor tiene que ser incluido: el propio artefacto de software. Finalmente, cambios menores en el título y/o el objetivo pueden requerirse.

Aunque cada Escenario Futuro se construye con un enfoque bottom-up, el conjunto de EA debe recorrerse en el modo “depth-first”. Esto significa que el primer Escenario Actual que debe considerarse es el Escenario Integrador de mayor nivel en la jerarquía. Si existe más de un nivel de Escenarios Integradores, se debe tomar un camino hasta llegar a un Escenario raíz. En un contexto de bajo grado de cambios en el proceso del negocio, se espera alguno o ningún impacto en los Escenarios Integradores. Debe recordarse que los Escenarios Integradores no contienen ninguna pieza propia de información, o sea, todos sus episodios son Escenarios Integradores de menor nivel, hasta llegar al que se compone de Escenarios raíz. Por lo tanto, los Escenarios Integradores

actúan como marco para ayudar a realizar la auténtica tarea de construcción de los EF. Una vez que se tuvieron en cuenta todos los Escenarios raíz, deben reconstruirse los Escenarios Integradores Futuros, utilizando la misma técnica aplicada en la generación de los Escenarios Integradores actuales.

Para la construcción de EA se ha propuesto un nuevo **mecanismo por proximidad**, el cual puede ser aplicado para construir los EF. De esta manera, por cada EF que se construye, se debe pensar si existen situaciones próximas relacionadas.

Proceso orientado a los Objetivos

La hipótesis principal aquí es básicamente la opuesta a la del proceso orientado a lo procedural. Los EA comprendidos por el alcance del Objetivo General del Sistema serán modificados. Esto es, Escenarios apareados tendrán cambios importantes en los objetivos. También, es de esperar que varios EA se vuelvan obsoletos y que aparezcan nuevos Escenarios.

La tarea en este caso, es estudiar el objetivo de cada Escenario Actual en el contexto del Objetivo General del Sistema, y determinar si ese objetivo permanece en el UdeD futuro. Si permanece, se debe adaptar el objetivo del Escenario en el contexto del UdeD futuro. En primera instancia, se obtiene una versión preliminar del objetivo del Escenario Futuro. Luego, se debe determinar si es posible cumplirlo con un sólo Escenario Futuro o si se necesita más de un Escenario. Si el objetivo del Escenario Actual no permanece en el contexto del UdeD futuro, el Escenario Actual se vuelve obsoleto.

Hasta aquí se han obtenido algunos esqueletos de EF, teniendo una versión preliminar de sus objetivos y quizás de sus títulos. Estos Escenarios deben completarse sintetizando otros componentes basados en el objetivo.

Cada EF y el conjunto íntegro se construyen usando un enfoque top-down. El conjunto de EA debe investigarse en un modo “breath-first”. Aquí, el primer Escenario Actual que debe considerarse es también el Escenario Integrador de más alto nivel. En este caso, se esperan muchos cambios en los Escenarios Integradores. Éstos Escenarios actúan como marco para ayudar a realizar la tarea efectiva de construcción de los EF, la cual se hace sobre los Escenarios raíz. Se aconseja, que al final del proceso, construir los Escenarios Integradores Futuros con la misma técnica usada para los Escenarios Integradores Actuales.

Proceso Híbrido

Cuando se encaran proyectos de software con un nivel intermedio de cambios en el proceso del negocio, se debe aplicar un enfoque híbrido. El proceso debe comenzar como si hubiera un nivel alto de cambios en los procesos. De esta manera, tomar los Escenarios Integradores Actuales y cuando el proceso llega a cada Escenario raíz o sub-Escenario analizar el grado de cambio esperado en el alcance del Escenario Actual bajo consideración. Según el grado de cambio, se deberá aplicar a este Escenario en particular una estrategia orientada a lo procedural o una orientada a los objetivos. Cambiando de una estrategia a otra en cada paso, se puede tratar con todos los posibles grados de cambios. Obviamente, un proceso híbrido se reduce a un proceso orientado a los objetivos si la respuesta es siempre “alto grado de cambio” para cada Escenario Actual. Esto no es precisamente verdad si la respuesta es siempre “bajo grado de cambio”. Por lo tanto, la pregunta del principio es todavía más importante, pero el peor precio a pagar por escoger un enfoque híbrido es trabajar más de lo necesario.

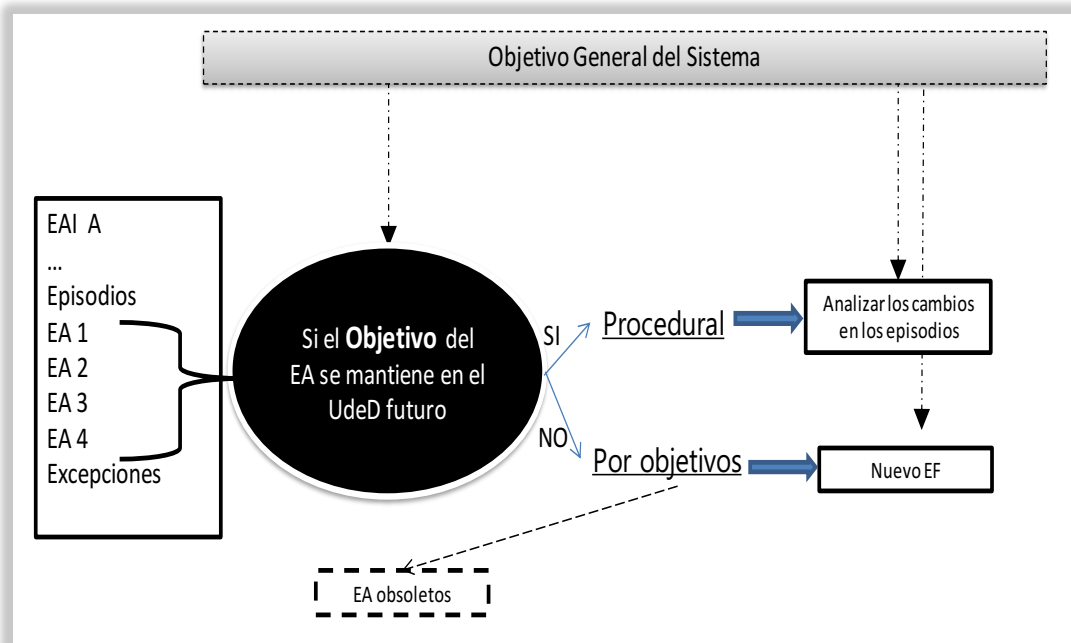


Figura 60 – Proceso de Construcción de EF

Puede observarse en la Figura 60 que cada EA (EA 1, EA 2, EA 3 y EA 4) del Escenario Integrador (EAI A) son analizados desde el punto de vista del Objetivo General del Sistema para saber si el objetivo del Escenario se mantiene dentro del UdeD futuro. De ser afirmativo, se analiza todo el EA, pero fundamentalmente los episodios, restricciones

y excepciones. De no mantenerse, se debe definir un nuevo objetivo del Escenario y describirlo completamente. Si bien en ambos enfoques (Procedural y Por Objetivos) pueden aparecer EA obsoletos, es mayor su incidencia al trabajar por objetivos.

Una vez completos los EF, deben ser Organizados, Verificados y Validados.

Durante la construcción de los EF se debe tener en cuenta aquella información que apareció tempranamente y fue registrada en las **Fichas de Información Extemporánea (FIE)** (ver **Capítulo 19**). Esta información debe ser analizada y de ser necesario, negociada (ver Figura 61).

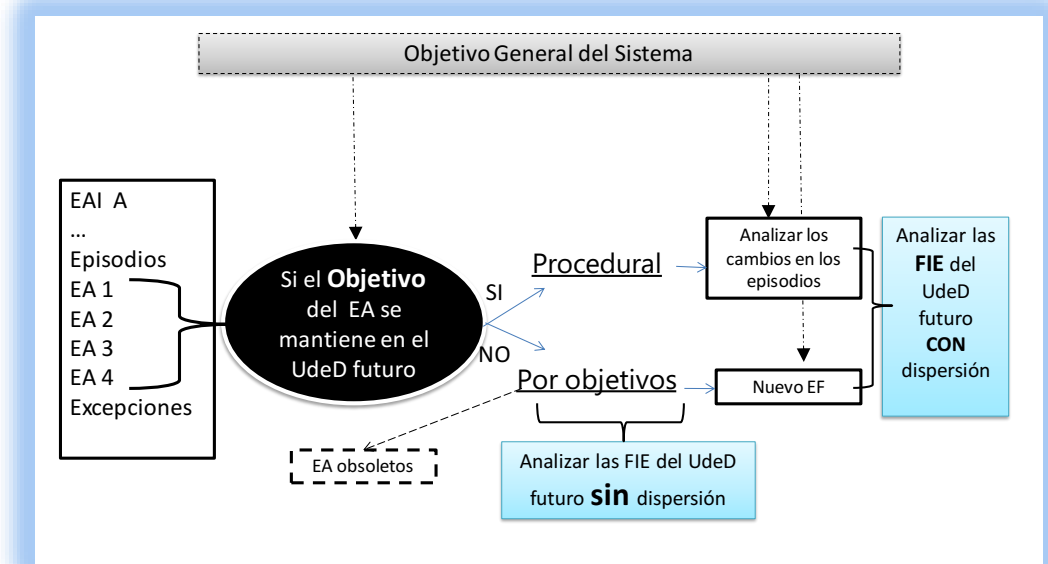


Figura 61 – Proceso de Construcción de EF con Información Extemporánea

Validación de Escenarios Futuros

Aunque los métodos, técnicas y herramientas de la validación del LEL o de los EA puedan ser coincidentes o muy similares con las que se utilizan durante la validación de los EF, el núcleo conceptual de ambas validaciones es notoriamente diferente. Cuando se realiza la validación de un EF la pregunta que subyace es ¿Es así como se desenvuelve esta situación actualmente?, mientras que al hacer lo mismo la pregunta implícita es ¿Es apropiado que las situaciones se realicen de esta manera cuando esté el sistema? o ¿Son apropiados los servicios que va a prestar el sistema en esta situación? Notoriamente cada confirmación debe ser interpretada como una cláusula de un contrato entre los

desarrolladores y los clientes y usuarios. No es necesario que este contrato se materialice en un documento legalizado para existir, ya existe informalmente.

Capítulo 9

Especificar Requisitos de Software

Resumen

Existen algunos modelos de la Ingeniería de Requisitos que contienen en sus descripciones los requisitos del software, como sucede con los Casos de Uso o los Escenarios, entre otros. Estos modelos pueden ser utilizados como documento de requisitos o decidir explicitarlos en un documento independiente. En el presente Capítulo se describe como extraer y priorizar los requisitos del software desde los Escenarios Futuros.

Especificación de Requisitos de Software

Para generar el documento de requisitos es necesario contar con los EF completos, verificados y validados, ya que el proceso busca los RF y algunos RNF en estos Escenarios. De esta manera, la lista de requisitos obtenida, también se encuentra verificada y validada, además de altamente consensuada. La mayoría de los RF se obtienen de aquellos episodios donde participa el actor *Sistema de Software*. Estos requisitos pueden estar explícitos, pero en ocasiones se encuentran implícitos en algún episodio, por lo que deben ser analizados adecuadamente. Los RNF pueden estar en las restricciones del contexto, de los recursos o de los episodios, como también en las excepciones del Escenario. También, puede aparecer RNF al analizar las condiciones de los episodios condicionales. Los RF y RNF deberán ser priorizados en reuniones que se llevarán a cabo con los clientes y usuarios [Hadad09b].

Para obtener los requisitos del software desde los EF se utiliza el meta-modelo de Especificación de Requisito [Hadad09a]. Como se puede observar en el primer cuadro de la Figura 62, este meta-modelo incluye la descripción del requisito; el tipo de requisito según la clasificación adoptada y el estado del mismo.

Requisito: descripción del requisito.

Tipo: indica si se trata de un RF o un RNF, o el tipo de RNF al que corresponde, o el tipo según alguna otra clasificación utilizada.

Estado: condición bajo la cual se encuentra el requisito, puede ser por ejemplo derivado, priorizado, etc.

Volatilidad: grado de estabilidad estimada del requisito.

Prioridad: importancia relativa que tiene el requisito para los clientes y usuarios.

Criticidad: necesidad relativa de implementación, puede indicarse si es obligatorio, deseable u opcional, o mediante un ranking de necesidad.

Factibilidad: posibilidad de implementación en el proceso del negocio, ya sea por razones sociales,

Costo de implementación: esfuerzo estimado para implementar el requisito en el Sistema de Software.

tecnológicas, ambientales, económicas, etc.

Figura 62 - Meta-Modelo de Especificación de Requisitos

De ser necesario, se pueden incorporar al meta-modelo los atributos tales como la volatilidad, prioridad, criticidad, factibilidad, riesgo y costo de implementación, tal como se muestra en el segundo cuadro de la Figura 62.

Este meta-modelo considera exclusivamente los atributos propios del requisito,

independiente del versionado y de los métodos de rastreabilidad utilizados, por tal motivo, no se han incluidos atributos como autor, fecha de creación, fecha de modificación, origen (fuente de información), versión, vinculación con modelos (LEL, EF, modelos de diseño, modelos de implementación, etc.), entre otros. Tampoco se han incluido las dependencias con otros requisitos ya que muchas de ellas, principalmente entre RF, se han tratado durante la construcción de los EF. De ser necesario tratar dependencias entre RNF, de seguridad, etc., se deberán procesar mediante una estructura de gestión propia.

El proceso para explicitar los requisitos se pueden observar en la Figura 63.

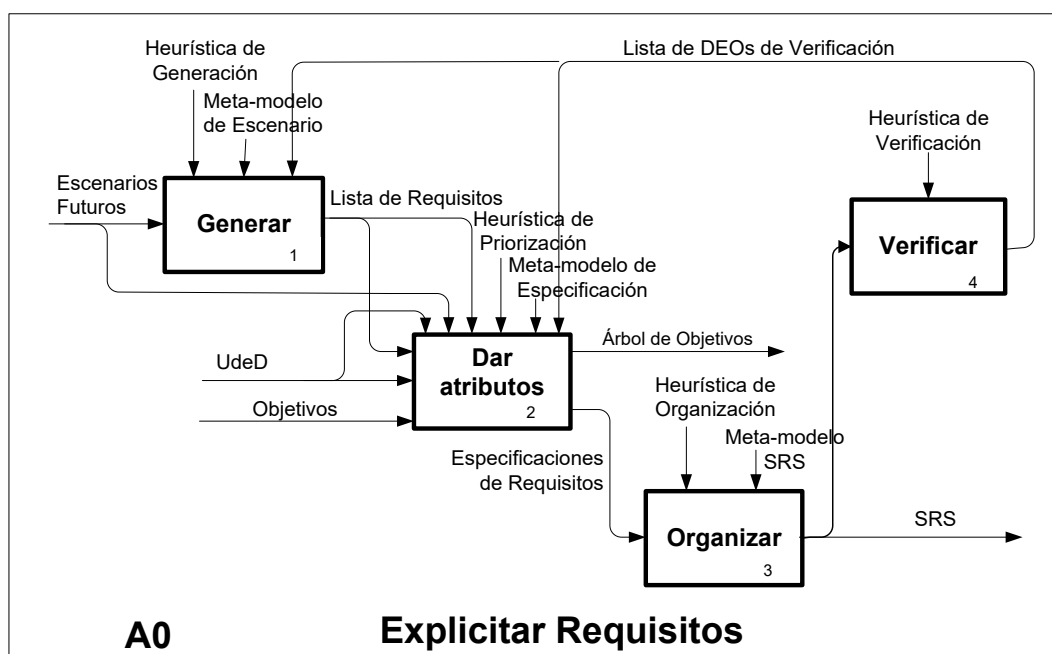


Figura 63 - SADT del Proceso de Explicitar Requisitos de Software

Para cada requisito de software, se elicitan y acuerdan los atributos para completar su especificación.

Finalmente, se redacta el Documento de Definición de Requisitos (ERS), basándose en algún estándar que la organización utilice o que se proponga de acuerdo al proyecto en particular. Cabe mencionar que este documento debe respetar el vocabulario utilizado en el UdeD, manteniendo vínculos al LEL. Se debe obtener un documento de fácil lectura para los usuarios, y que además presente muy baja ambigüedad.

Se debe recordar que, principalmente en este momento del proceso, los vínculos al glosario deben ser al LELR (ver **Capítulo 15**).

Tiene por objetivo obtener una lista de requisitos únicos del Sistema de Software, lo más completa posible. Incluye extraer los requisitos de cada EF y eliminar requisitos repetidos. Los RF se obtienen de varios componentes del Escenario, principalmente de los episodios donde el actor *Sistema de Software* actúa. No se considerarán para este propósito los episodios que sean sub-Escenarios. Además, pueden extraerse RF de las excepciones, donde la solución es una sentencia donde actúa el Sistema de Software, no siendo la solución un Escenario excepción. En algunos casos las condiciones de los episodios condicionales pueden involucrar datos, estados de datos o acciones y, por lo tanto, pueden obtenerse RF. Esto mismo ocurre en la causa de excepciones.

Cabe aclarar respecto a las condiciones, que los datos o estados de datos que participan en la misma serán aquellos que el sistema deberá administrar, y en el caso de una condición que involucra una acción debe participar en ella el sistema. Por ejemplo, existen condiciones que involucran una acción, lo que implica que se ha abreviado el discurso.

Episodio: “Si el equipo no tiene [número de pedido](#) entonces [ASIGNAR NÚMERO DE PEDIDO](#)”
Se descompone en: El sistema verifica que el equipo tenga [número de pedido](#) y Si no tiene [número de pedido](#) entonces [ASIGNAR NÚMERO DE PEDIDO](#).

En el caso que la causa de una excepción involucre un RF, con frecuencia, significa que el sistema debe realizar algún tipo de verificación. Por ejemplo:

Causa de excepción: El sistema no reconoce el [número de formulario](#).
Se obtiene el RF: El sistema debe verificar la existencia del [número de formulario](#).
Causa de excepción: El [pasaporte](#) emitido no es retirado antes de los 60 días.
Se obtiene el RF: El sistema debe controlar que el [pasaporte](#) se retire dentro de los 60 días.

Se pueden extraer un RNF básicamente de las restricciones de los episodios donde participa el *Sistema de Software*, adicionalmente pueden encontrarse en la causa de excepciones. También a partir de los recursos involucrados, pueden inferirse algunos requisitos de calidad. Si el recurso es un símbolo tipo objeto del LEL, puede buscarse en la noción del mismo alguna característica particular que pueda indicar la necesidad de un RNF. Por ejemplo:

Recurso: fecha de segunda dosis (debe haber pasado al menos 21 días de la primera)

Se obtiene el RNF: El sistema debe soportar la consulta a todas las jurisdicciones para analizar la fecha de la primera dosis.

En general la IR ha hecho cierto abuso de los RNF que tienen que terminar siendo operacionalizados. Entonces ¿cuándo operacionalizar un RNF? Cuando es un tema de interés para el cliente-usuario, cuando desea saber cómo el Sistema de Software va a atender el requisito. De lo contrario, cuando no es de su interés, sigue teniendo sentido operacionalizarlo pero es contraproducente hacerlo en la etapa de IR.

A continuación, se presentan dos ejemplos de causa de excepción que implica un RNF:

Causa de excepción: Fallos en la conexión.

Se obtiene el RNF: El sistema debe poder asegurar que todas las transacciones se puedan recuperar ante fallos del sistema o violaciones al sistema de seguridad.

Esto no significa que cada uno de estos componentes del Escenario sea siempre un requisito, es justamente en esta actividad en la que se determina cuáles de las partes presentes en los EF son efectivamente requisitos.

La Figura 64 muestra un ejemplo de lista de requisitos extraída de un solo Escenario Futuro, donde se aprecian cuatro RF y un RNF (extraído del Caso SOP”). El objetivo del EF no se ha utilizado como posible contenedor de requisitos. Estos objetivos pueden representar las grandes funcionalidades del sistema, siempre y cuando el actor Sistema de Software tenga una participación importante en los episodios del Escenario. Por lo tanto, ellos pueden abarcar varios requisitos del Sistema de Software, pudiendo utilizarse esta relación para establecer dependencias entre ellos.

Una vez extraídos los requisitos de cada EF, debe controlarse la existencia de redundancia de los mismos. Es muy frecuente que un requisito esté presente en más de un Escenario. Por ejemplo, el requisito “El sistema debe permitir la selección de un cliente por apellido, por tipo o por localidad” puede presentarse en más de un Escenario donde debe realizarse alguna tarea vinculada a un cliente. A partir de esta sub-actividad, se obtiene una lista de requisitos únicos.

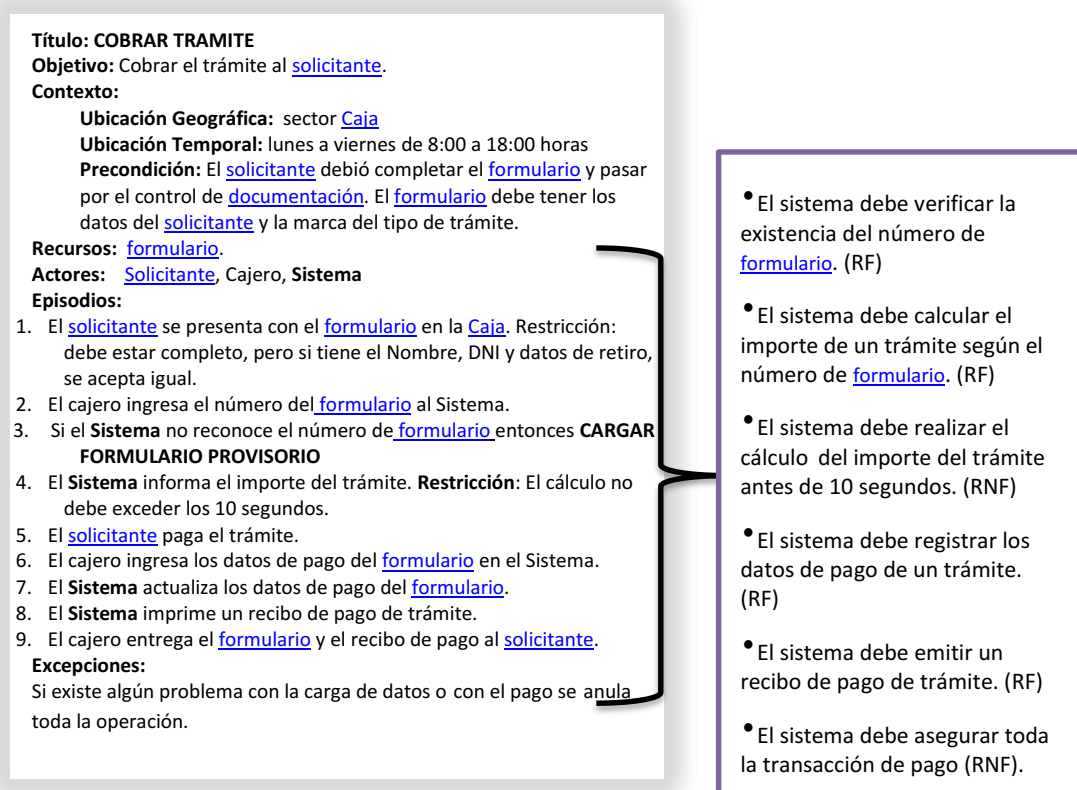


Figura 64 - Ejemplo de lista de requisitos de un Escenario Futuro (Caso SOP)

Puede observarse que en los episodios 6 y 7 aparece léxico del UdeD futuro: “datos del pago”, “recibo de pago”, “marca del trámite” y “Tipo de trámite” que serán símbolos del LEL_R (ver **Capítulo 15**).

Puede observarse en los episodios 1 y 4 que existen puntos de vista en las restricciones, los cuales se describen en el **Capítulo 16**.

Se puede observar que el Sistema de Software es utilizado por momentos como un Actor (episodio 3, 4, 7 y 8) y en otros como si fuera un recurso (episodio 2, 6). En los casos donde el Sistema de Software es visiblemente un actor, el RF es extraído sin dificultad. Cuando aparenta ser un recurso, el episodio debe ser contemplado desde la acción implícita que debe realizar. Por ejemplo, “El cajero carga los datos del pago en el sistema” se debe leer como que el sistema debe permitir ingresar y registrar los datos del pago. En la Tabla 12 se resumen los posibles tipos de requisitos a extraer de los EF, dando ejemplos de cada uno de ellos. Se puede observar en los ejemplos de esta Tabla que existen palabras o frases subrayadas, ellas representan vínculos a símbolos del LEL. Se debe tener presente que dependiendo del componente o ítem puede extraerse directamente

un requisito, puede evaluarse si es un requisito o puede inferirse un requisito.

Tipo	Componente	Ejemplo
<i>Requisito Funcional</i>	<i>Episodios</i> donde participa el Actor Sistema de Software	El Sistema de Software imprime un recibo de pago del trámite.
	<i>Solución de Excepción</i> donde participa el Actor Sistema de Software	El Sistema de Software envía mensaje sobre el límite de fecha para la incineración de un pasaporte .
	<i>Condiciones</i> donde en el episodio participa el Actor Sistema de Software	<p>Si <<el tipo de trámite es pasaporte original>> entonces el Sistema de Software ...</p> <p>Si <<es el segundo chequeo del equipo>> entonces el Sistema de Software ...</p> <p>Si <<el Sistema de Software verifica que el equipo no tiene asignado número de pedido>> entonces el Sistema ...</p>
	<i>Causa de Excepciones</i>	<p>El Sistema de Software no reconoce el número de formulario.</p> <p>Algún dato del convocado es inválido o nulo.</p> <p>El pasaporte emitido no es retirado antes de los 60 días.</p> <p>Administración modifica en el Sistema de Software algún dato cargado por Comercio Exterior.</p>
<i>Requisito No Funcional</i>	<i>Restricción</i> donde en el episodio participa el Actor Sistema de Software	El Sistema de Software debe realizar el cálculo del importe del trámite antes de 10 segundos.
	<i>Causa de Excepciones</i>	<p>Existen problemas en el servicio de internet.</p> <p>El Sistema de Software no se puede conectar al servidor donde se encuentran las estadísticas.</p>
	<i>Recursos</i>	e-mail archivo de presentación

Tabla 12 - Tipos de requisitos extraídos de Escenarios Futuros

Actividad Dar atributos a los requisitos

En esta actividad utiliza la lista de requisitos obtenida en la actividad anterior. Luego, se debe retornar al UdeD para completarla. Las especificaciones presentan vínculos al LEL, pues por un lado parte de la información que proviene de los EF, los cuales han sido descriptos usando el LEL. Luego, la nueva información que se incorpore también debe respetar el uso del LEL.

Los clientes y usuarios dan prioridades a los requisitos con apoyo de los/las ingenieros/as de requisitos. La priorización de requisitos se puede hacer según dos criterios: urgencia e importancia. La urgencia indica el orden en que atenderán y la importancia cual es el valor que tiene para el cliente. Obviamente, si se atienden ambos criterios en forma conjunta debe predominar la urgencia por sobre la importancia en el ordenamiento del trabajo.

. Se realizan reuniones para determinar la importancia relativa que tiene un requisito para los clientes y usuarios, y para organizar aquellos requisitos que deben implementarse inicialmente frente a aquellos que pueden postergarse. Se deben tener en cuenta, además de la dependencia de los requisitos, la diversidad de intereses de los clientes y usuarios, las limitaciones de recursos, las necesidades del negocio y las imposiciones del mercado, entre otros factores. Existen variadas técnicas de priorización de requisitos, como, por ejemplo: AHP (Analytic Hierarchy Process) [Saaty80] [Karlsson98], QDF (Quality Function Deployment) [Cockburn97] y EVOLVE [Hadad08].

La tarea de dar prioridades puede realizarse previamente al describirse los EF, dando prioridad a los mismos, pues puede ser una tarea más fácil que priorizar cada requisito individualmente, ya que muchos de ellos están interrelacionados y deben considerarse sus prioridades en simultáneo.

Se debe tener en cuenta que no sólo se les da prioridad a los requisitos sino también puede establecerse su criticidad, factibilidad y riesgo, que son algunos de los atributos que presentan los requisitos. El fundamento surge a partir del origen del requisito, es decir del o de los EF en donde se resuelve. En cuanto al estado, en principio es “propuesto”, pero una vez que se le asigna su prioridad pasa a estar “aprobado”. Actualizaciones al estado irán surgiendo durante el resto del proceso de desarrollo del software. Posteriormente, estas especificaciones de requisitos son validadas mediante reuniones con los usuarios.

Actividad Organizar

Los requisitos se describen en un documento según el formato de la ERS impuesto por el cliente o adoptado por el equipo de desarrollo, a partir de las especificaciones previamente generadas. El propósito principal es organizar los requisitos de manera tal de obtener el máximo de legibilidad.

Deben mantenerse en la ERS vínculos al LEL, como una forma de reducir la ambigüedad del documento escrito en lenguaje natural. Adicionalmente se mantienen vínculos a los Escenarios Futuros para mantener la rastreabilidad hacia los orígenes de los requisitos. Debe además considerarse que muchos requisitos están relacionados con otros, en muchos casos se trata de RNF con RF. El mantenimiento de estos vínculos permite tener siempre en cuenta las dependencias entre ellos.

Algunas recomendaciones para describir un requisito:

- Redactar utilizando la forma “*El sistema debe ...*”, con lo cual se identifica claramente la capacidad o la condición que el Sistema de Software brindará o acatará.
- No incluir más de un requisito en cada sentencia.
- Evitar el uso de la frase “de ser necesario”, pues debe quedar claro bajo qué condiciones opera el requisito.
- Evitar palabras ambiguas como “generalmente”, “frecuentemente”, “rápido”, “flexible”, “amigable”, “preferible”.
- Evitar deseos ilimitados en las capacidades del Sistema de Software, por ejemplo “bajo cualquier plataforma”, “independiente del motor de base de datos”, “100% confiable”, “sin fallas”.
- Para RNF, incluir valores de medición. Se puede describir el valor deseado y el valor máximo o mínimo aceptable.

Si se considera el formato presentado por [IEEE830], entonces gran parte de sus secciones pueden completarse con la información obtenida, como se describe a continuación. La sección 1.2 de la ERS “Alcance del Proyecto” incluye el nombre del proyecto que puede obtenerse del título del Escenario Futuro integrador de nivel 0, y el objetivo del producto que se obtiene del documento de objetivos del sistema o de los objetivos de los Escenarios Integradores. En la sección 1.3 de la ERS “Definiciones, acrónimos y abreviaturas” puede hacerse referencia al LEL. En la sección 1.4 “Referencias” se indicarán referencias al

conjunto de EF y al LEL. La sección 2.2 “Funciones del producto” se puede completar con los objetivos de los EF con el sistema como actor importante. En la sección 2.3 “Características de los usuarios” se puede completar con los actores de los EF que interactúan con el software y vincularlos con los respectivos símbolos tipo sujeto del LEL. La sección 3 “Requisitos Específicos” incluye todos los requisitos extraídos de los EF, donde podrán organizarse por su tipo, por el EF origen, por prioridad, por actor del EF.

Actividad Verificar

La actividad de verificación debe comprobar la correcta generación del ERS, determinando si la extracción de requisitos desde los EF fue realizada exhaustivamente, si la priorización es consistente y si al ERS contempla todos los requisitos explicitados y priorizados. La técnica de verificación sugerida por su alta efectividad [Leite05] es la de inspección. Se genera a partir de esta actividad una lista de discrepancias, errores y omisiones para corregir el ERS. Con la lista generada se vuelve a las actividades Generar la lista y Dar Atributos, según el tipo de defectos encontrados. Los aspectos principales que cubre la verificación son:

1. Verificar la sintaxis:
 - Detectar el uso de más de un verbo en la descripción del requisito.
 - Detectar el uso de adjetivos subjetivos.
 - Detectar el uso de frases ambiguas o subjetivas (cuyo significado dependa del interlocutor).
2. Verificar contra los componentes del EF:
 - Controlar que todo requisito proviene de un componente del EF: objetivo, episodio, restricción, causa de excepción, solución de excepción, recurso o condición de episodio.

Se deben analizar aquellas **Fichas de Información Extemporánea** donde se indicó que debe ser incluido en los requisitos (ver **Capítulo 19**).

3. Verificar dependencias entre requisitos:
 - Detectar relaciones no establecidas según sus orígenes en EF.
4. Verificar atributos:
 - Controlar que prioridad, criticidad, factibilidad y riesgo asignados a un

requisito estén dentro de los rangos establecidos.

- Controlar que prioridad, criticidad, factibilidad y riesgo del requisito sean consistentes según su dependencia con otros requisitos.
- Controlar que se haya fundamentado y clasificado el requisito.

5. Verificar referencias al LEL:

- Controlar que todo símbolo del LEL es identificado al mencionarse en un requisito.
- Controlar el uso correcto de símbolos del LEL.

Se deben analizar los **episodios ubicuos**, los cuales pueden contener RF y RNF (ver **Capítulo 16**).

Estrategia para Asignar Prioridades

Con el objetivo de cerrar el ciclo de especificación, se presenta una estrategia de asignación de requisitos obtenidos desde los EF [Hadad09b]. En ella se entiende que el fundamento del requisito se establece teniendo en cuenta su origen, es decir, su participación en uno o más EF ayudando a cumplir un objetivo.

Para el resto de los atributos, y probablemente para otros atributos que se deseen agregar, es necesario definir una escala de valores admisibles. Estas escalas podrán tener unos pocos niveles cualitativos o podrán ser escalas numéricas arbitrarias. En particular, de conservarse el atributo relacionado con el costo de implementación del requisito, es preferible que el mismo se exprese en forma cualitativa o en una escala numérica con pocos valores.

En cuanto al estado, en principio se debe partir de un estado tal como “propuesto”, pero una vez que se le hayan asignado todos sus atributos se modificará a un estado “aceptado” o similar. Actualizaciones al estado irán surgiendo durante el resto del proceso de desarrollo del software.

Es difícil definir la volatilidad de los requisitos en una etapa tan temprana del proceso de desarrollo del software. Es cierto que de algunos requisitos se puede conocer su condición de inestable por el contexto en que el mismo fue creado, en virtud de muchos indicadores tales como la dificultad para definir con precisión todo o parte del EF que le dio origen. Desafortunadamente los requisitos con aparente baja volatilidad pueden ser muy

afectados por avatares del proceso del negocio o del contexto en el que el mismo se desenvuelve.

De todos los atributos restantes: prioridad, criticidad, riesgo, factibilidad y costo de implementación, el que desencadena decisiones acerca de cómo seguir en el proceso es la prioridad. Los otros atributos influyen sobre la prioridad y ayudan a su definición.

Cabe aclarar que al haberse obtenido la lista de requisitos de un conjunto de EF ya validados y negociados con los clientes y usuarios, la tarea de asignación de prioridades se debe realizar con una visión macroscópica. Es decir, se aspira a asignar prioridades a conjuntos consistentes de requisitos y no a requisitos individuales.

En este punto es importante reexaminar el propósito de asignar prioridades a los requisitos. Este atributo es necesario para resolver la implantación de los requisitos en el futuro Sistema de Software y eventualmente la división del proyecto en etapas donde los requisitos de mayor urgencia serían materializados en las primeras etapas. Esta visión del posible uso de la urgencia pone en conflicto todo el proceso de software en que la elicitación de requisitos se halla inmersa. Se ha adoptado una modalidad atómica para la obtención de requisitos y repentinamente se cambia a un proceso incremental y/o iterativo. En otros términos, en un proceso en cascada o similar se implementarían todos los requisitos elicitados en forma monolítica, sin puestas en servicio parciales, por lo que la asignación de prioridad no es una actividad necesaria. El aparente o real conflicto entre la elicitación de requisitos en forma total y su implementación por etapas no es necesariamente un problema, ya que puede provenir de una decisión tomada en forma explícita. Si este fuera el caso, la asignación de prioridades se realimenta sobre los EF, los cuales deben ser vueltos a estudiar para evaluar en su contexto cada implementación parcial. Si se hubiera seguido estrictamente un proceso iterativo y/o incremental, tampoco sería necesaria la asignación de prioridad, ya que la misma fue asignada al definir el alcance de la iteración.

En el caso en que se trate de una situación en la que se ha definido una única lista de requisitos a partir de la cual se seleccionará un subconjunto que será materializado en una implantación parcial, es necesario precisar la estrategia con la que se definirán las prioridades.

Como primera actividad para el establecimiento de prioridades, se debe seleccionar la técnica de asignación de las mismas. Independientemente de la técnica elegida, se propone a continuación un mecanismo nuevo para la compatibilización de las prioridades,

en virtud de que en este proceso la lista de requisitos ha sido obtenida de un modelo del proceso del negocio coherente. Estrictamente se propone a continuación un mecanismo que asigne las prioridades en forma consistente desde el primer momento, haciendo innecesaria toda actividad posterior de compatibilización.

Esta propuesta se basa en asignar prioridades a conjuntos de requisitos que satisfacen algunos de los objetivos o sub-objetivos del Sistema de Software. Es importante enfatizar que durante la construcción de los EF fue necesario establecer con claridad, si aún no lo estaba, el Objetivo General del Sistema para seleccionar la estrategia de construcción.

Diversos autores [Mylopoulos92] [Lamsweerde98] [Dardenne93] [Potts95] [Cockburn97] han propuesto, con otro propósito, abstracciones y refinamientos de objetivos generales u objetivos estratégicos para encontrar sub-objetivos o super-objetivos según sea el caso. En estas propuestas se ha descrito en detalle un proceso que identifica objetivos del negocio y construye una jerarquía de objetivos y sub-objetivos. En la presente tesis se adopta el uso de la misma estrategia, pero sobre el Objetivo General del Sistema. Las abstracciones fueron realizadas durante la definición del Objetivo General del Sistema, antes de construir los EF [Doorn02], mientras que los refinamientos deben realizarse en este punto, al asignar prioridades a los requisitos.

Luego, la tarea de asignar prioridades a los requisitos consiste en:

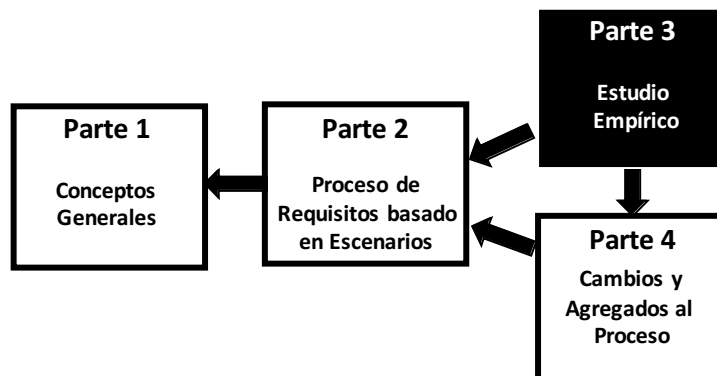
- Dividir el Objetivo General del Sistema en sub-objetivos, utilizando las técnicas orientadas a objetivos mencionadas.
- Continuar la división de los sub-objetivos en sub-objetivos más detallados hasta que se pueda asegurar que cada una de las hojas del árbol admita una única prioridad sin incoherencia interna. En general, es suficiente con uno o dos niveles de descomposición.
- Asignar prioridades a las hojas del árbol.
- Relacionar los EF y los sub-objetivos, considerando qué EF satisfacen cada sub-objetivo. No se consideran los EF donde no interviene el Sistema de Software bajo estudio. Puede ocurrir que un EF participe en el cumplimiento de más de un sub-objetivo y que un sub-objetivo sea satisfecho por más de un EF.
- Para todos los EF que materializan un solo sub-objetivo o más de un sub-objetivo con la misma prioridad, trasladar la prioridad del o de los sub-objetivos a los requisitos que dio lugar el EF.

- Para todos los EF que participan en dos o más sub-objetivos, con al menos dos prioridades diferentes, distinguir qué requisitos del EF están involucrados con qué sub-objetivo y asignarle su prioridad.
- Para RNF no asociados a EF, asociar, de ser posible, a sub-objetivos y trasladar su prioridad. Si se asocia a más de un sub-objetivo con distinta prioridad, asignarle la prioridad más alta.
- Cuando un EF participa en dos o más sub-objetivos con al menos dos prioridades diferentes, es necesario rearmar el EF en uno o más Escenarios de transición, que describan la parte correspondiente al proceso del negocio que existirá en la iteración correspondiente.

Cabe aclarar que, aunque en las pautas indicadas hacen mención sólo al atributo prioridad, se debe tener en cuenta que, dependiendo del criterio de cálculo de la prioridad, éste atributo podrá estar representando la importancia o beneficio del requisito para los clientes y se asignarán en simultáneos valores a atributos tales como criticidad, factibilidad, costo de implementación y riesgo, entre otros, de acuerdo al método adoptado.

Parte 3

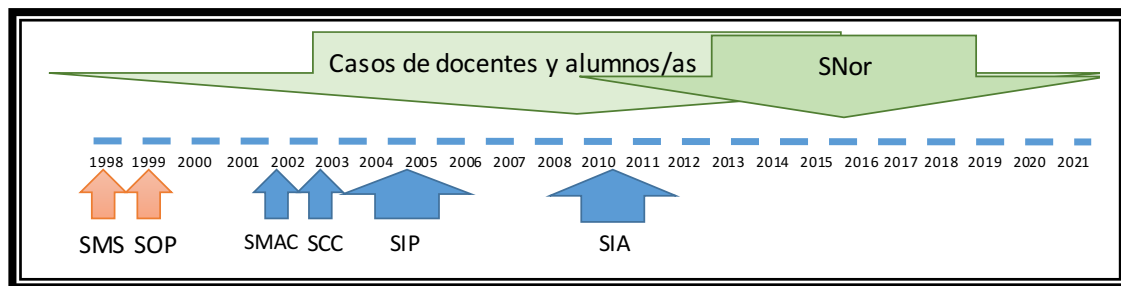
Estudio Empírico



Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Introducción

El estudio empírico que se presenta a continuación, se ha sustentado en un base de conocimiento que comienza a gestarse en el año 1997 y sigue hasta la fecha. Los casos suman aproximadamente 200, con una gran variedad de temas. En la Figura 65 se puede observar un resumen cronológico de los mismos. Las flechas de arriba corresponde a trabajos realizados por docentes y alumnos/as de carreras de grado y posgrado de diferentes Universidades. Las flechas de abajo corresponden a trabajos realizados por profesionales. En la parte superior, el grupo más extenso ha sido realizado en empresas reales, pero cabe aclarar que, en la mayoría de los casos, el cliente sólo actuó como un colaborador sin ninguna expectativa relacionada con la construcción del nuevo Sistema de Software. El otro caso de las flechas superiores, SNor, corresponde al caso Norpack que es una fábrica de cajas de cartón corrugado el cual tiene la particularidad de ser un caso documentado. Con respecto a las flechas de abajo, el primer grupo (SMS y SOP) corresponde a casos reales realizados por un grupo de investigación donde participó la tesista (Meeting Scheduler y Sistema Nacional para la Obtención de Pasaportes) en la Universidad de Belgrano y el segundo grupo (SMAC, SCC, SIP y SIA) son casos verdaderos generados en la actividad profesional de la tesista. En estos casos no solo existió una necesidad real sino también una gran expectativa en la construcción del Sistema de Software. Si bien las diferencias entre ambos grupos son sustanciales, a la hora de construir los requisitos del software para realizar el estudio empírico, no se ha detectado ninguna influencia.



- SNor – Sistema Norpack. Fábrica de cajas de cartón corrugado.
- SMS – Sistema del Meeting Scheduler – Universidad de Belgrano
- SOP – Sistema de Obtención de Pasaportes – Sistema Nacional de la República Argentina
- SMAC- Sistema de Mesa de Ayuda para Cajeros Automáticos. Empresa de venta y soporte de cajeros.
- SCC - Sistema de Control de Calidad de Productos Farmacéuticos. Laboratorio con tercerización de la producción.
- SIP - Sistema Integral de Producción. Empresa exportadora de productos de cueros.
- SIA - Sistema de Gestión Integral de Alumnos. Institución educativa de nivel superior.

Figura 65– Cronología de los casos que sustentan el estudio empírico

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Introducción

A partir de 2010 hasta la fecha, cuando se da comienzo al presente estudio empírico, se realizaron diferentes Proyectos de Investigación que se detallan en la Tabla 13.

Período	Institución	Nombre del Proyecto	Investigadores/as
01/01/2010 31/12/2011	UNLaM	Consolidación de Requisitos de Software	Director: Jorge Doorn Línea 1: Validación de Requisitos a cargo de Gladys Kaplan Línea 2: Gestión de Requisitos a cargo de Jorge Doorn Línea 3: Especificación de Requisitos a cargo de Graciela Hadad
01/01/2012 31/12/2013	UNLaM	Contextualización de Procesos de Requisitos	Director: Jorge Doorn Co-directoras: Graciela Hadad y Gladys Kaplan Investigadores/as: Claudia Litvak, Nora Gigante, Renata Guatelli, Andrea Vera
01/01/2014 31/12/2015	UNLaM	Estabilidad de los Documentos en el Proceso de Requisitos	Director: Jorge Doorn Co-directora: Gladys Kaplan Investigadores/as: Nora Gigante, Renata Guatelli
01/01/2016 31/12/2017	UNLaM	Ingeniería de Requisitos para Sistemas CRM	Directora: Gladys Kaplan Co director: Gabriel Blanco Asesor externo: Jorge Doorn Investigadores/as: Guillermo Hindi, Nora Gigante, Claudia Litvak, Gabriel Pousadas, Andrea Vera
01/01/2015 31/12/2018	UNLu	Generación semi automática de casos de prueba a partir de Escenarios	Directora: Gladys Kaplan Asesor externo: Jorge Doorn Investigadores/as: Walter Panisi, Claudia Ortiz, David Petrocelli, Eugenia Céspedes y Julián Massolo
01/01/2019 31/12/2020	UNLaM	Aspectos No Funcionales de los Procesos de Requisitos	Directora: Gladys Kaplan Co director: Gabriel Blanco Investigadores/as: Andrea Vera, Renata Guatelli y Laura Pepe Investigador Externo: Jorge Doorn
01/01/2021 31/12/2022	UNLaM	Agrupamiento selectivo de Escenarios	Directora: Gladys Kaplan Co director: Gabriel Blanco Investigadores/as: Juan Pablo Mighetti Investigador Externo: Jorge Doorn Alumnos: Yonatan Salguero y Candela Santander

Tabla 13 – Proyectos de Investigación relacionados con la Parte 3

Capítulo 10

Estudio empírico del LEL

Resumen

La calidad del LEL ha sido un tema de estudio por muchos años en los cuales se han encontrado algunas respuestas satisfactorias las cuales se han incorporado al proceso de construcción del LEL existente. Otros problemas, como ser la completitud del LEL, permanecen sin solución. En el presente documento se realiza un estudio con el objetivo de determinar el origen de estos problemas y darles una solución definitiva.

Tal como ocurre en prácticamente todos los modelos de la Ingeniería de Software, y de la Ingeniería de Requisitos en particular, maximizar la completitud de los modelos es un desafío constante. Si bien la calidad del LEL ha sido mejorada en varias ocasiones con la incorporación o modificación de diferentes mecanismos, como la heurística de construcción, la verificación con inspecciones, los patrones sintácticos, etc., aún persiste un problema crucial que está relacionado con la impericia de lograr que diferentes personas, en igualdad de condiciones, obtengan glosarios equivalentes. Esto ha llevado a que en muchas ocasiones se pronuncie el siguiente interrogante:

¿Afecta al Proceso de Requisitos un glosario incompleto?

Existe un nivel de incompletitud que puede resultar tolerable en el contexto del objetivo original del LEL, donde es utilizado para asegurar la comunicación y mejorar la precisión de los documentos generados. Pero se ha vuelto perjudicial en el contexto de los otros usos que se le han ido incorporando al LEL, como ser la derivación de Escenarios, analizar el cubrimiento del LEL en los Escenarios para detectar omisiones, desambiguar los requisitos del software, etc. Por tal motivo, la problemática de la completitud del LEL es relevante.

Estudios relacionados con la completitud del LEL

En esta sección se resumen las principales conclusiones de diferentes estudios realizados sobre el proceso de construcción del LEL. Dichos estudios pusieron de manifiesto algunas debilidades estructurales del modelo y ciertas dificultades cognitivas con las que se debe enfrentar el ingeniero/a de requisitos. En esta sección se presentan dichos resultados y en las secciones siguientes, se detallan los posibles orígenes de estas debilidades.

Cabe aclarar que, entre todos los estudios descriptos en esta sección y el Proceso de Requisitos presentado en esta tesis, ha existido una superposición o paralelismo de trabajos realizados por diferentes grupos de investigación que amerita una presentación cronológica.

La Figura 66 reproduce textualmente los problemas y conclusiones del primer estudio realizado sobre el LEL [Hadad07]. Cabe destacar que este estudio es contemporáneo a la definición original del *Proceso de Requisitos basado en Escenarios* y justamente, esa es su importancia para esta sección. Algunos de los problemas mencionados en la Figura 66,

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

ya estaban asociados a la completitud. Las soluciones de la Figura 66, tan tempranas en la creación del proceso, han sido modificadas con el tiempo. Tal es caso del Problema 5, detección de sinónimos, donde se formuló inicialmente realizar más entrevistas, pero luego se pudo observar que esto es poco apropiado, ya que por un lado es justamente la realización de varias entrevistas una de las principales causas de ocultamiento de sinónimos tanto en el caso en que participen diferentes ingeniero/as de requisitos como cuando se accede a diferentes fuentes de información; por otro lado la reiteración de entrevistas por cuestiones de segundo orden en un proceso que involucra una fuerte interacción con los clientes y usuarios no es recomendable. Además, es altamente probable que participen varios ingeniero/as de requisitos en la construcción del glosario, los cuales pueden incluir símbolos aparentemente diferentes, pero que en realidad no lo son. La presencia de este problema está muy relacionada con las fuentes de información consultadas. Para asegurar que no quedan sinónimos sin detectar, se incluyó la Verificación del LEL basada en una inspección guiada por formularios, específicamente el Formulario IV. ANÁLISIS DE LA NOCIONES E IMPACTOS que ayuda a detectar sinónimos analizando semánticamente todas las nociones e impactos (ver Anexo A).

En la Tabla 14 se ha realizado una breve descripción del lugar en el Proceso de Requisitos existente, donde fueron incluidas las mejoras de la Figura 66 y otras mejoras posteriores, abarcando un periodo de tiempo aproximado desde 1996 hasta el 2006. Sólo para enfatizar, se han agrupado los problemas de la Figura 66 en dos conjuntos: los de completitud y los de proceso. Cabe aclarar que los de completitud están incluidos en los de proceso, pero se han separado para marcar su importancia.

Problema 1:

- Determinar las entradas al léxico: cuáles debían ir, cuáles no eran necesarias, cuáles no correspondían al dominio del problema.

Criterio Adoptado:

- Incorporar solo frases o palabras correspondientes al dominio de la aplicación. Excluyendo éstas:
 - entradas obvias, ya que son de conocimiento general de todas las personas (usuarios, ingenieros de sistemas y personas externas al Universo de Discurso).
 - entradas excesivamente específicas, por considerarlas que no forman parte central del problema en cuestión.
 - entradas con un significado amplio (ambiguas), porque su definición puede resultar muy genérica

Problema 2:

- Describir símbolos afines con la misma estructura. Este problema se mostró como crítico cuando participaron varias personas.

Criterio Adoptado:

- Clasificar las entradas y establecer para cada tipo de entrada la forma de cómo se debe definir, tanto la noción como el impacto.

Problema 3:

- Repetir o no impactos comunes a distintos símbolos.

Criterio Adoptado:

- Repetir los impactos comunes en todos los símbolos.

Problema 4:

- Determinar el nivel de detalle al definir las entradas.

Criterio Adoptado:

- Volcar en cada definición de un símbolo todo el conocimiento relevante que sobre él se posee.

Problema 5:

- Dificultad en la detección de sinónimos.

Criterio Adoptado:

- Realizar varias entrevistas con diversos usuarios ayuda a obtener sinónimos o clarificar sinónimos confusos.

Identificar posibles sinónimos comparando símbolos con nociones o impactos similares.

Problema 6:

- Al seleccionar símbolos, determinar la utilización del “Verbo” o del “sustantivo” como nombre del símbolo. Ejemplo: *Controlar las Huellas Digitales* o *Control de Huellas Digitales*.

Criterio Adoptado:

- Utilizar el “sustantivo” como nombre del símbolo.

Problema 7:

- Determinar el nombre de la entrada en caso de frases largas.

Criterio Adoptado:

Incluir la frase completa como nombre de la entrada, ya que se debe respetar el vocabulario propio del usuario

Problema 8:

- Dificultad para establecer si una frase corresponde a una noción o a un impacto en un determinado símbolo.

Criterio Adoptado:

- Identificar el símbolo dentro de la clasificación de entradas. De acuerdo al tipo de símbolo, reconocer la frase como una noción o como un impacto.

Figura 66 – Lista de problemas y soluciones publicada en 1996

Nro. y descripción del Problema (ver Figura 66)		Tipo	Incluido en ...
1	Determinar las entradas al léxico: cuáles debían ir, cuáles no eran necesarias, cuáles no correspondían al contexto del futuro sistema.	Compleitud	Se menciona en la Heurística de construcción.
2	Describir símbolos afines con la misma estructura. Este problema fue crítico debido a la participación de varias personas.	Compleitud	Se menciona en la Heurística de construcción. (Se considera muy baja la probabilidad de que sea necesario agregar tipos).

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

3	Repetir o no impactos comunes en distintos símbolos o hacer referencia a un nuevo símbolo que contenga dichos impactos aunque éste no sea parte del contexto del futuro sistema.	Proceso	Se menciona en la Heurística de construcción.
4	Determinar el nivel de detalle al definir las entradas.	Complejidad	Patrón sintáctico y actividad de Verificación
5	Dificultad en la detección de sinónimos.	Proceso	Se menciona en la Heurística de construcción y en la actividad de Verificación
6	Al buscar símbolos, determinar la utilización del “verbo” o del “sustantivo” como nombre del símbolo.	Proceso	Se menciona en la Heurística de construcción. (Aún se lo sigue discutiendo).
7	Determinar el nombre de la entrada en caso de frases largas.	Proceso	Se menciona en la Heurística de construcción.
8	En algunos casos, dificultad para establecer si una frase corresponde a una noción o a un impacto en un determinado símbolo.	Proceso	Patrón sintáctico y en la actividad de Verificación

Tabla 14 – Resolución de los problemas en el proceso de construcción del LEL existente

Puede observarse en lo que sigue, que la completitud del LEL ha sido una preocupación recurrente. En 2003 se realizó un estudio cuantitativo referido a la cantidad de símbolos identificados y a su forma de agrupamiento [Doorn03] [Ridao06]. En estos estudios se utilizó el método de captura y recaptura denominado DPM (Detection Profile Method) aplicado a 9 LEL construidos por diferentes grupos. Estos LEL tienen la particularidad que su fuente de información ha sido un caso escrito sobre “*Planes de Ahorro Previo para la Adquisición de Vehículos 0km*”. El objetivo de este estudio fue estimar la cantidad de omisiones existentes. Puede observarse en la Tabla 15 que la cantidad de símbolos encontrados por cada grupo es bien disímil, lo que indica que cada grupo observó una parte diferente del léxico del macrosistema o desarrolló su trabajo con una calidad diferente de la de los otros grupos. La dispersión en la cantidad de símbolos detectados desde el máximo de 65 hasta el mínimo de 24 es por sí notoria, pero si se observa que aún en el grupo que detectó 65 símbolos omitió por lo menos otros 53 que sí fueron observados por algún otro grupo surge que el problema de completitud es muy importante. El número total de símbolos diferentes detectados fue de 118, por lo tanto, ningún grupo identificó más del 55% de los términos.

Nro. de LEL	Cantidad total de Símbolos
1	52
2	29
3	30
4	33
5	65
6	27
7	44
8	24
9	54

Tabla 15 – Resultados del estudio [Doorn03]

En base a estos datos, otro estudio continuó avanzando sobre los 9 mismos LEL. Ahora en una aproximación cualitativa [Litvak13] sobre la semántica de los símbolos. Se dividió el macrosistema en categorías y se determinó la cantidad de símbolos que había identificado cada grupo en cada categoría. De esta manera se pudo determinar la cantidad de símbolos relevantes y no relevantes obtenidos por todos los grupos. Se pudo establecer estadísticamente que los grupos tuvieron distintas percepciones del macrosistema según las categorías y que delimitaron el contexto en estudio de forma diferente. Estos resultados corroboran los anteriores, dejando nuevamente en evidencia un serio problema con respecto a la completitud del LEL.

Un estudio para identificar las buenas prácticas en la construcción del LEL [Antonelli13] convocó a un grupo de 50 personas para escribir el LEL sobre un mismo contexto de aplicación y así identificar puntos débiles en las descripciones. El resultado ha sido una serie de guías a modo de buenas prácticas, las cuales toman al LEL como un modelo contenedor de información para especificar los requisitos del software o para avanzar en el diseño, asumiendo que el LEL contiene información acerca del nuevo Sistema de Software.

Todos los estudios para estimar la completitud del LEL arribaron a la conclusión de que el problema existe, pero no lograron avanzar más allá que justamente eso, estimar la completitud. El hecho de realizar estos estudios con un caso escrito, dejaba un camino por experimentar, referido a la construcción del LEL con usuarios reales. Por tal motivo y en el marco de esta tesis, en 2019 se realizó un caso controlado en un contexto real,

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

utilizando entrevistas y validando los modelos construidos con los usuarios. También en estas condiciones los datos obtenidos fueron similares. Trabajaron dos grupos de alumnos de grado en el Sistema de Biblioteca de la Universidad Nacional del Oeste en el año 2018. La particularidad de este caso es que ambos grupos realizaron las entrevistas en forma simultánea y, por lo tanto, contaron con la misma información. Su entrevistada fue la bibliotecaria. Coincidiendo con los datos de los estudios anteriores, cada grupo observó de manera diferente el contexto. La cantidad de símbolos obtenidos fue de 60 y 68. La Tabla 16 unifica estos resultados, presentando en la primera columna el total de símbolos por tipo, y luego describe cuántos de ellos coinciden en ambos LEL. Finalmente, el porcentaje de coincidencia correspondiente.

Tipo	TOTAL de Símbolos (sin repeticiones)	Cantidad de Símbolos iguales en ambos LEL	Porcentaje de coincidencia
SUJETO	6	4	67%
OBJETO	29	13	45%
VERBO	37	21	57%
ESTADO	10	7	70%

Tabla 16– Resultados de los LEL del Sistema de Biblioteca

El número total de símbolos diferentes detectados fue de 82 y de símbolos que vieron ambos grupos de 45, por lo tanto, estos dos grupos coincidieron en sólo el 54,8% de los símbolos. Se puede observar una importante similitud de los resultados con los registrados en la Tabla 15.

En resumen, existe un 67% de símbolos sujeto observados por ambos grupos y la pregunta inmediata es ¿Qué sucedió con el 33% que alguno de los grupos no vio? ¿Y con el 55% de objetos? ¿Y con el 43% de verbos? ¿Y con el 30% de estados? Una respuesta posible está en la dificultad para establecer los límites del sistema y otra es la subjetividad de cada grupo. El resultado de este estudio consolidó la hipótesis acerca de que las notorias diferencias existentes se refieren en gran parte a cuestiones relacionadas con la percepción de las personas.

En otro trabajo de investigación llevado a cabo por Oliveros et. al., centrado en la enseñanza de las técnicas de elicitación [Oliveros12] propuso la observación utilizando una cámara Gesell. Si bien el objetivo estuvo centrado en la elicitación de requisitos y no los aspectos de calidad del LEL, este trabajo muestra un grado de incompletitud en los requisitos obtenidos similar a los detectados en [Doorn03] para la construcción del LEL.

Tomando todo el conocimiento expresado con anterioridad y la experiencia en el uso, se analizan a continuación diferentes aspectos del proceso de construcción del LEL presentado en el **Capítulo 7**, que afectan su calidad. Para una mejor comprensión se los dividió en Factores internos y Factores externos.

Factores internos que afectan la calidad del LEL

Los factores internos son aquellos elementos que el proceso de construcción del LEL no ha definido correcta o completamente y que, por lo tanto, afectan su completitud y consistencia. El primer factor interno y el que más perturba su calidad, se refiere a la generación de una *lista de símbolos candidatos* como primera actividad del proceso de construcción del LEL. La heurística de construcción propone realizar entrevistas no estructuradas para identificar palabras o frases relevantes del macrosistema [Leite90]. Esto ya supone un esfuerzo cognitivo importante, porque se debe entender el contexto del negocio para identificar qué parte del léxico es relevante en dicho macrosistema, con la gran dificultad que en general no se tiene ningún conocimiento previo o peor aún, se tienen algunos preconceptos no necesariamente acertados que suelen ocultar las peculiaridades del UdeD que está siendo estudiado. Mientras esto sucede, se debe prestar atención a mantenerse dentro del Objetivo General del Sistema y a confirmar en cada paso los límites del contexto en estudio. Este esfuerzo pone al ingeniero/a de requisitos en una postura inadecuada fundada en la propia inhabilidad del ser humano de manipular muchos conceptos a la vez, lo cual coincide con la propuesta de Wirth para la programación estructurada [Wirth71]. Pero existe otro problema con la lista candidata de símbolos, ya que actúa como un riel¹² que ayuda en la construcción del LEL, pero simultáneamente limita su evolución [Basili75]. Es importante resaltar que esta lista inicial suele ser muy incompleta y sesgada. Incompleta, por la ya mencionada inhabilidad del ser humano de considerar muchos aspectos al mismo tiempo y sesgada porque usualmente esta lista se construye con personal de línea media o alta en la pirámide organizacional y, por lo tanto, representa una mirada particular del macrosistema. Cuando se avanza en el proceso de construcción y comienza el refinamiento del LEL, la lista inicial rara vez es refutada. En muchos casos se la acepta sólo por desconocer su origen. En la gran variedad de casos analizados, se ha podido observar una alta coincidencia entre

¹² Según RAE “Carril de una vía férrea”

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

esta lista inicial y la lista final. De esta manera, se puede afirmar que construir una lista inicial tiene un alto riesgo de convertirse en una limitación que atenta contra la calidad del modelo, ya que congela los símbolos muy tempranamente y desde miradas parciales, como puede ser la visión léxica de la línea media o alta en la pirámide organizacional.

Casos	Cant. Símbolos Lista Inicial	Cant. de Símbolos Lista final	Cant. De Símbolos iguales
SIA	70	93	62
SCC	121	91	77
SNor	34	36	31
	43	46	43
SMAC	39	54	35

Tabla 17 - Comparación entre la lista inicial de símbolos del LEL y la final

En la Tabla 17 se puede observar que en algunos casos la cantidad de símbolos candidatos inicial puede tener más símbolos que la lista final, como es el Caso SCC. En este caso se definió incorrectamente el UdeD y se incluyeron términos que no correspondían. Se puede observar que los Casos SNor, por ser un caso que utiliza una fuente de información textual, tienen una coincidencia aún mayor. En general, se puede apreciar que las listas iniciales suelen ser menores a las finales, esto se debe, por lo general, a símbolos no detectados, a homónimos que se juntaron, a límites incorrectos del área en estudio, etc. Este último punto sucede porque establecer los *límites del UdeD* es uno de los primeros desafíos con los que se encuentra el ingeniero/a de requisitos, pero es recién al finalizar la Ingeniería de Requisitos cuando se tiene certeza del mismo. Por tal motivo, es de esperar que sea el LEL, por ser el primer modelo que se construye, el más perjudicado en este sentido y el más afectado por la percepción del ingeniero/a de requisitos. En este sentido, el alcance del LEL, y por lo tanto su completitud, pueden verse perjudicados por esta situación.

Este problema no se presenta con la misma fuerza para el LEL_R (ver **Capítulo 15**) ya que los límites están claramente establecidos debido a que este glosario se construye una vez completos los EF.

Un factor interno no estudiado adecuadamente, tiene que ver con el *empotramiento de símbolos*. El empotramiento surge en el contexto de pares o grupos de más símbolos en los que al menos uno de ellos está constituido por dos o más palabras, y uno de ellos contiene a los restantes. Si bien podría imaginarse una situación de empotramiento con

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

más de tres símbolos, los casos reales usualmente presentan situaciones de empotramiento entre dos o tres símbolos. En el caso de un par de símbolos la única situación posible es que un símbolo esté contenido en el otro. En el caso de ternas es posible encontrar tres situaciones. La más común es la de anidamiento puro, un símbolo está contenido en un segundo y este a su vez en un tercero, pero puede ocurrir que dos símbolos estén incluidos en un tercero siendo completamente disjuntos entre sí o teniendo una superposición parcial. En este caso, la narrativa, tanto escrita como oral, evita las redundancias y por lo tanto omite uno de los dos símbolos, usualmente el más corto. Esto es muy similar a lo que ocurre con las inclusiones de tuplas u objetos en bases de datos. Los símbolos empotrados no ocurren en el vacío o en forma arbitraria, usualmente están fuertemente relacionados. Cuando el empotramiento es entre símbolos existentes en el LEL, permite visualizar la fuerza de los vínculos, pero su principal aporte es cuando alguno de los símbolos involucrados en el empotramiento ha sido omitido, en ese caso considerar el empotramiento aumenta la posibilidad de identificar nuevos símbolos. Puede observarse en la Figura 67 un ejemplo del Caso SNor donde se omitió el símbolo “Oficina de Planificación de la Producción” el cual fue detectado al analizar el empotramiento del símbolo “Persona de guardia de la oficina de planificación de la producción”.

Persona de guardia en la oficina de planificación de producción (S)

Noción

- Es una persona que trabaja en la oficina de planificación de la producción.

Impacto

- Puede [emitir el programa de fabricación extraordinario](#).

Figura 67 – Ejemplo de símbolo con empotramiento aparente (Caso SNor)

Para asegurar que realmente era una omisión, se analizaron diferentes LEL. En la Tabla 18 se puede observar que existe un conflicto con este símbolo, debido a que dos grupos lo identificaron y los otros tres grupos, no. En el análisis posterior del caso, se puede deducir que dicho símbolo, por su relevancia, es un símbolo del LEL.

LEL del Caso SNor	¿Aparece?
Grupo 1	No
Grupo 2	No
Grupo 3	Si
Grupo 4	No
Grupo 5	Si

Tabla 18 – Ejemplo de uso de empotramiento (Caso SNor)

Para abordar el análisis del empotramiento es conveniente revisar los mecanismos lingüísticos con los que se introducen palabras o frases en nuevos dominios asignándoles significados especiales. Estos son la *metonimia*, el *préstamo* y la *construcción de sintagmas*. En la Figura 68 se pueden ver estos elementos en un ejemplo.

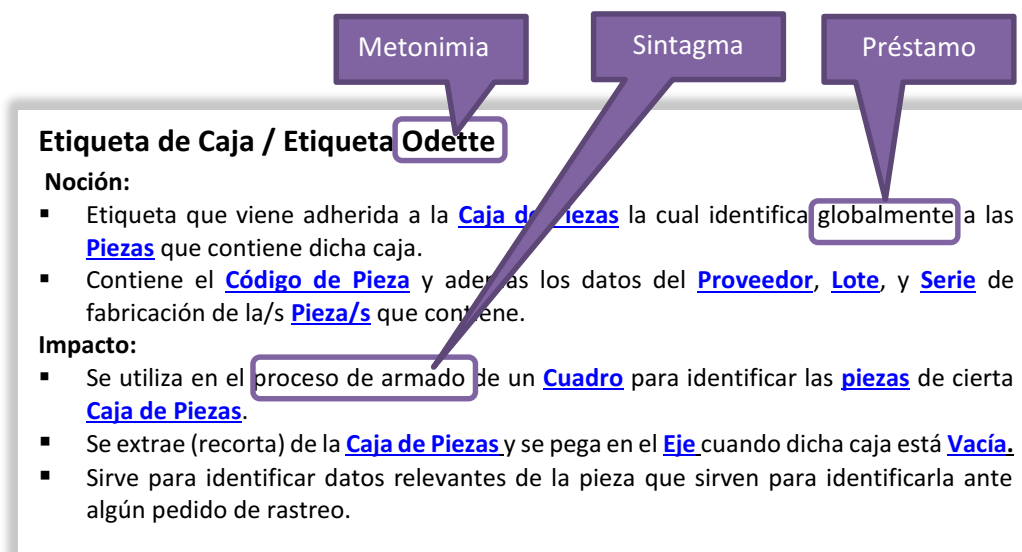


Figura 68 - Ejemplo modificado de Metonimia, Préstamo y Sintagma

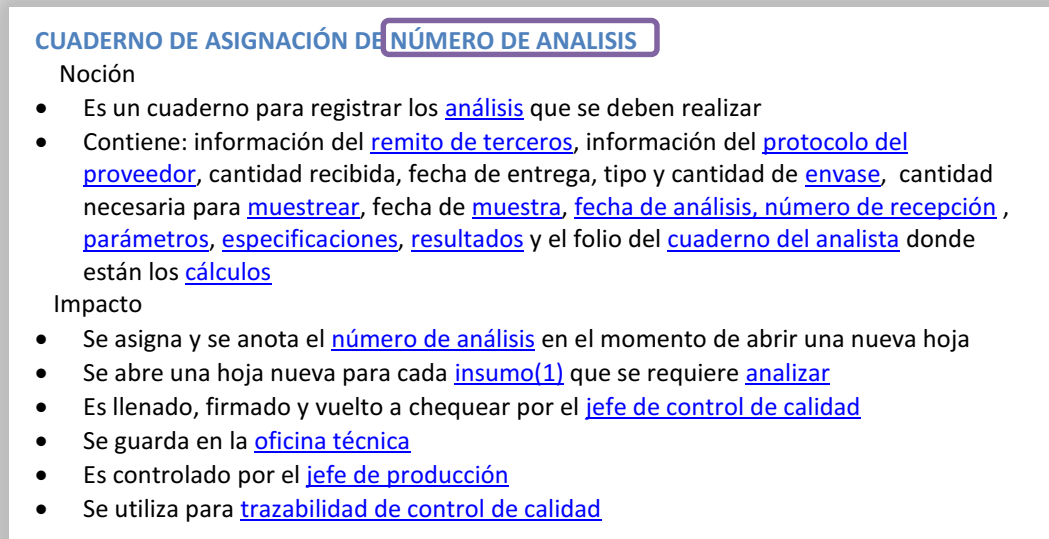
La metonimia consiste en tomar una palabra o frase de uso corriente y utilizarla con un sentido diferente. El préstamo consiste en adaptar una palabra o frase del lenguaje corriente, de otro dominio o eventualmente de otro idioma modificándola levemente y asignándole un significado diferente. Los sintagmas son frases que constituyen una unidad sintáctica en las sentencias y que cumplen una función determinada con respecto a otras palabras de la oración. Los sintagmas pueden ser verbales, nominales, preposicionales o adjetivos. Los sintagmas nominales son registrados en el LEL como

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

objetos o como sujetos, los sintagmas verbales como verbos, mientras que los sintagmas adjetivos definen estados. Los sintagmas preposicionales no son de especial interés. Tanto los símbolos creados por metonimia como los creados mediante sintagmas son los más propensos a errores. Los primeros porque no son fáciles de reconocer y los segundos porque suelen conformar empotramientos.

Para ejemplificar la aparición e importancia del empotramiento, se presenta en la Figura 69 un caso de empotramiento que marca claramente el estrecho vínculo semántico entre el objeto “Cuaderno de Asignación de Número de Análisis” y el símbolo “Número de Análisis”. En este caso, dicha información también está contenida en la descripción del verbo, pero marcar el empotramiento permite visibilizar rápidamente los vínculos semánticos más fuertes. Además, para quienes no están familiarizados con el LEL de ese macrosistema, permite una manera diferente de navegación, desde otra perspectiva. La navegación del LEL se realiza a través de los hipervínculos entre los símbolos. La navegación desde el empotramiento es por la fuerza intrínseca de los vínculos en los nombres de los símbolos y permite una revisión rápida del núcleo principal de cada símbolo.



CUADERNO DE ASIGNACIÓN DE NÚMERO DE ANALISIS

Noción

- Es un cuaderno para registrar los [análisis](#) que se deben realizar
- Contiene: información del [remito de terceros](#), información del [protocolo del proveedor](#), cantidad recibida, fecha de entrega, tipo y cantidad de [envase](#), cantidad necesaria para [muestrear](#), fecha de [muestra](#), [fecha de análisis](#), [número de recepción](#), [parámetros](#), [especificaciones](#), [resultados](#) y el folio del [cuaderno del analista](#) donde están los [cálculos](#)

Impacto

- Se asigna y se anota el [número de análisis](#) en el momento de abrir una nueva hoja
- Se abre una hoja nueva para cada [insumo\(1\)](#) que se requiere [analizar](#)
- Es llenado, firmado y vuelto a chequear por el [jefe de control de calidad](#)
- Se guarda en la [oficina técnica](#)
- Es controlado por el [jefe de producción](#)
- Se utiliza para [trazabilidad de control de calidad](#)

Figura 69 - Ejemplo de símbolos con empotramiento (Caso SCC)

Se debe prestar atención a los casos particulares donde existe una superposición textual pero no corresponde a un empotramiento del LEL. Una causa de este problema sucede cuando aparece empotrada una palabra o frase que es utilizada con otra acepción diferente a la descrita en el LEL. Por ejemplo, en un contexto de Biblioteca, el símbolo “libro” se refiere a todo lo que puede ser consultado por un socio. En el símbolo “Libro de

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

visitantes”, “Libro” no es un empotramiento porque su significado es diferente. En este caso, no existe empotramiento a menos que se defina un homónimo. También se debe prestar una adecuada atención al uso de la nominalización de los verbos, o sea utilizar un sustantivo para referirse a un verbo (ejemplo, utilizar “autorización” en vez de “autorizar”). Si la nominalización existe en el LEL se puede estar en presencia de un empotramiento, pero si en el LEL sólo existe el verbo, se debe analizar particularmente. Por ejemplo, el uso de “documentación” para expresar el verbo “documentar”. En este caso se debe asegurar que no se esté en presencia de una omisión. De ser un símbolo omitido se lo debe incorporar al LEL y si, existe empotramiento. De lo contrario, no hay empotramiento. Por lo tanto, cada vez que se identifica un posible empotramiento primero se debe asegurar que su uso corresponda con la definición existente en el LEL. Se puede observar en la **Tabla 19** que existen empotramientos que sólo pueden ser clarificados retornando al UdeD.

Símbolo	Tipo en el LEL	Símbolos empotrados	Observaciones
Auxiliar de Ventas	S	Auxiliar Ventas	¿Ventas hace referencia al Dpto. de Ventas o al verbo Vender?
Pendiente de devolución	E	Devolución	Si existiera un homónimo entre un sustantivo y un verbo ¿hace referencia al objeto o al verbo?
Etiqueta de Control de Calidad	E	Etiqueta Control de Calidad	Control de Calidad ¿es un verbo, un objeto o hace referencia al Dpto. de Control de Calidad?

Tabla 19 – Ejemplos de empotramientos que deben ser analizados en el UdeD

Por todo lo dicho, es necesario analizar cada empotramiento de la siguiente manera:

- Los símbolos empotrados deben ser utilizados con el significado del LEL. Por ejemplo, “Libro de visitantes” en un entorno de una biblioteca tiene a “libro” como empotrado. Pero “libro” en este contexto tiene otra acepción, por lo tanto, no es un empotrado excepto que exista el homónimo correspondiente.
- Analizar todos los nombres de símbolos que sean frases, especialmente en el caso de los verbos.
- Evaluar si una frase más breve o una palabra puede tener un significado específico en el UdeD para detectar nuevos símbolos.

Finalmente, detectar el empotramiento tiene las siguientes ventajas:

- Mejora la comprensión del léxico del macrosistema.
- Permite detectar símbolos, jerarquías conceptuales ocultas, etc.
- Robustece los vínculos semánticos del léxico.
- Mejora la completitud del glosario.

Un tema preexistente en la construcción del LEL que tiene el mismo peso de representación que los sinónimos, son los homónimos, pero no están incluidos en el modelo LEL. La homonimia es una forma de ambigüedad léxica [Ting15]. Puede observarse en la Figura 70 que existen dos grupos de homónimos, pero en el LEL sólo interesan los del tipo homógrafos. Los homófonos pueden tener alguna dificultad para ser identificados, pero de ser detectados deben ser representados como símbolos independientes.

HOMÓNIMAS	
HOMÓFONAS	HOMÓGRAFAS
Suenan igual	Suenan igual
Significado distinto	Significado distinto
Se escriben distinto	Se escriben igual
<i>Ejemplo: Vaca de animal y Baca portaequipajes</i>	<i>Ejemplo: Traje de vestir y Traje de traer</i>

Figura 70 – Clasificación de homónimas

La existencia de homónimos en el LEL es variada. Por un lado, como ya se mencionó, existen los símbolos homógrafos. Por el otro lado, existen símbolos que tienen un significado general igual, pero con algunas variaciones especialmente en el impacto. Por ejemplo, el estado “autorizado” para un presupuesto significa lo mismo, o sea que la proyección de gastos del área es aprobada por la gerencia, pero el comportamiento de esta autorización en el contexto de la atención al cliente es diferente de la del área de stock. Estos casos deben ser representados como homónimos en el LEL.

Se han utilizado los *patrones sintácticos* para dar un marco de referencia sobre qué información debe contener la noción y el impacto de cada símbolo dependiendo de su tipo. Su objetivo principal es homogeneizar las descripciones del LEL y ayudar al ingeniero/a de requisitos acerca de qué información debe contener cada componente. Los patrones sintácticos mejoran la completitud del LEL ya que orientan en la búsqueda de información, especialmente aquella no tan visible. Además, mejoran sensiblemente la legibilidad del glosario y son indispensables para asegurar el Principio de Circularidad. Pero paradójicamente, en vez de asegurar los vínculos puede generar vínculos débiles, ya

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

que en ocasiones se fuerza la incorporación de información en la noción o en el impacto para cumplir con lo propuesto en el patrón sintáctico, generando vínculos artificiales. Esto se debe a que los patrones sintácticos sugieren la incorporación de cierta información, la cual debe ser buscada en el contexto e incorporada en el componente que corresponde, siempre y cuando exista.

NOCIÓN			
TIPO	[Leite90]	[Kaplan00]	[Hadad07]
SUJETO	Quién es el sujeto.	Dejar claro quién es el sujeto.	Definir quién es.
OBJETO	Definir el objeto e identificar los otros objetos con los que se relaciona.	Definir el objeto e identificar otros objetos con el cual tiene relación.	Definir <u>qué representa, sus características</u> e identificar otros objetos con los que se relaciona.
VERBO	Decir quién ejecuta la acción, cuando ocurre, y los procedimientos implicados en la acción.	Indicar quien la ejecuta, cuando sucede y la información necesaria para la ejecución de la acción.	<u>Describir concisamente la actividad que representa, su propósito</u> e identificar quién lo ejecuta, cuándo <u>(cronológicamente o relativo a otras actividades) y dónde se realiza.</u>
ESTADO	Aclarar lo que significa y cuáles son las acciones que llevará a esta situación.	Dejar claro qué significa y qué <u>actividades</u> y acciones llevaron a este estado.	Tiene que hacer claro lo que significa y los <u>estados y actividades</u> o acciones que llevaron a este estado.

Nota: se han destacado las diferencias con respecto a la definición original [Leite90], con subrayado e itálica.

Tabla 20 – Evolución del componente Noción

Un ingeniero/a de requisitos con poca experiencia puede transformar el patrón sintáctico en una exigencia y forzar la incorporación de todo lo que contiene el patrón sintáctico. También cabe destacar que esta presión que sienten algunos/as ingeniero/as de requisitos puede estar sustentada en que la verificación del LEL inspecciona la clasificación en función del contenido de la noción y del impacto, alertando en caso de sobrantes o faltantes. Pero este mecanismo de verificación tiene utilidad para detectar información incorrectamente ubicada y la omisión de información, pero sólo aquella que realmente existe en el macrosistema.

El uso de los patrones sintácticos en el LEL está ampliamente aceptado, aun así, quedan algunas dudas sobre su uso. Puede observarse en la Tabla 21 que existen varias versiones de los patrones sintácticos. Éstas han surgido de la experiencia en el uso del LEL. Existe otra definición de los patrones sintácticos que se mantiene muy cercana a la original, la cual está orientada a construir un LEL colaborativo [Antonelli16].

IMPACTO			
TIPO	[Leite90]	[Kaplan00]	[Hadad07]
SUJETO	registrar cuáles son las acciones que realiza	registrar las <u>responsabilidades y actividades que realiza o recibe</u>	debe registrar las <u>responsabilidades o actividades que ejecuta o recibe</u>
OBJETO	las acciones que se pueden aplicar al objeto deben estar declaradas	describir las <u>actividades</u> / acciones que pueden ser aplicadas al objeto.	debe describir las <u>actividades</u> o acciones que se aplican al objeto <u>o que se realizan con él</u>
VERBO	Identificar las situaciones que impiden la ocurrencia de la acción, qué reflejos de la acción en el ambiente (otras acciones que deberán ocurrir) y cuáles las nuevas situaciones de la acción.	registrar las <u>operaciones / procedimientos envueltos en la acción</u> e identificar las situaciones que impiden que la misma se realice, qué otras acciones desencadenan en el ambiente <u>y qué situaciones son causadas por la acción. Indicar que estados se desencadenan a partir de esta acción.</u>	debe registrar las <u>acciones, operaciones o procedimientos involucrados en la actividad</u> y debe identificar las situaciones que impiden la ocurrencia de la actividad, qué otras actividades se desencadenan en el UdeD y <u>qué situaciones ocurren debido a esta actividad</u>
ESTADO	debe identificar otras situaciones. y acciones que pueden ocurrir desde esta situación específica	<u>identificar otros estados y actividades</u> /acciones que pueden ocurrir a partir de este estado	debe identificar <u>otros estados y actividades</u> o acciones que pueden ocurrir a partir de este estado

Nota: se han destacado las diferencias con respecto a la definición original [Leite90], con negrita, subrayado e itálica.

Tabla 21 – Evolución del componente Impacto

En la Tabla 20 y Tabla 21 se presentan tres definiciones de los patrones sintácticos que se han publicado a lo largo de 17 años. Se toma como referencia la versión original del LEL que corresponde a Leite et. al. [Leite90] y luego, dos versiones posteriores [Kaplan00] y [Hadad07], donde se publicaron con algunas diferencias. Puede observarse que algunos de los cambios realizados son fundamentalmente aclaratorios.

En el estudio empírico de la presente tesis se ha utilizado el patrón sintáctico de [Kaplan00] por ser la versión que corresponde con la Inspección de LEL y la más utilizada en los casos de estudio existentes. Sobre estos patrones sintácticos, se proponen mejoras en el Capítulo 12, las cuales contribuyen a la completitud del LEL.

Factores externos que afectan la calidad del LEL

Los factores externos son aquellos elementos que existen en el UdeD y que su inadecuado tratamiento perjudica la calidad del LEL. Un factor al que se le debe prestar una especial atención está referido con los ***discursos retóricos o artificiales de los clientes-***

usuarios, los cuales no utilizan tan efectivamente el léxico real del macrosistema. Es durante las entrevistas donde los clientes y usuarios pueden utilizar un léxico forzado o artificioso por temor a que no se les comprenda adecuadamente. En una entrevista, principalmente durante la etapa de Comprensión del UdeD actual, puede asumir que el ingeniero/a de requisitos no le comprende, ya que recién está estudiando el contexto de la aplicación. En estos casos, el cliente-usuario se considera obligado a sobre-explicar todo aquello que considere importante. En este discurso exagerado, pueden aparecer expresiones con léxico artificial, empotramientos falsos y que los símbolos que el ingeniero/a de requisitos identifique no lo sean y queden ocultos los reales. Obviamente, el proceso de construcción debe eliminar estos errores, pero se encuentra con una lista inicial de símbolos que dificulta percibirlos y, por lo tanto, exige un importante esfuerzo adicional.

Existen varios autores que reclaman una definición más clara hacer de qué es un *Estado* para la Ingeniería de Software [Ryan14] [Al-Fedagui20], afirmando que existen inconsistencias en la definición y uso. Tampoco se proporciona un marco para trabajar con ellos. En el Proceso de Requisitos que se presenta en esta tesis, incluye los estados desde el primer modelo que se construye, lo que muestra un interés particular en su identificación y tratamiento desde el inicio del proceso. Si bien esto es un logro, su tratamiento se va desvaneciendo al pasar de un modelo a otro. Pueden observarse las siguientes dificultades a la hora de incluir los estados en el LEL (en los **Capítulos 15 y 16** se tratan los estados en los Escenarios):

- 1) Pueden haberse interpretado erróneamente como partes del nombre del símbolo.
- 2) Deben ser detectados, requiere pericia de parte del ingeniero/a de requisitos.
- 3) Debe ser analizado su significado particularmente en el contexto en estudio.
- 4) Se deben identificar los estados homónimos.

La Figura 71 muestra un ejemplo de un símbolo objeto y sus estados.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

<p>Insumo(1) Noción</p> <ul style="list-style-type: none"> Definición del Sector Control de Calidad Es todo aquello que ingresa a la empresa : las materia prima, los granel, los producto semielaborado (1), los producto terminado, el material de acondicionamiento , los envases y los productos importados Tiene un número de recepción Puede tener el estado amarillo, verde o rojo <p>Impacto</p> <ul style="list-style-type: none"> Se le toma una muestra Se le realiza control de calidad Se lo envía a elaborar Puede ser enviado al depósito de cuarentena, al depósito de aprobados, al depósito de rechazados, al depósito de devoluciones según corresponda Si es producto o producto importado, se le puede tomar una contramuestra 	
<p>Amarillo / En Cuarentena / Pendiente Noción</p> <ul style="list-style-type: none"> Indica que un insumo(1) no fue analizado <p>Impacto</p> <ul style="list-style-type: none"> El insumo(1) se analiza El insumo(1) puede pasar a aprobado o rechazado El insumo(1) es almacenado en el depósito de cuarentena 	<p>Rojo / Rechazado Noción</p> <ul style="list-style-type: none"> Indica que el insumo(1) no es apto para su uso Sucede después de analizar <p>Impacto</p> <ul style="list-style-type: none"> El insumo(1) puede ser destruido El insumo(1) es almacenado en el depósito de rechazados
<p>Verde / Aprobado Noción</p> <ul style="list-style-type: none"> Indica que un insumo(1) es apto para su uso <p>Impacto</p> <ul style="list-style-type: none"> El insumo(1) es utilizado para elaborar El insumo(1) es almacenado en el depósito de aprobados 	

Figura 71 – Ejemplo de un símbolo base y sus Estados (Caso SCC)

Los estados pueden aparecer solos, como estados terminales, o dentro de una cadena de estados que aplican a un mismo símbolo base. El pasaje de un estado a otro depende de ciertas condiciones que se producen en el contexto en estudio, por lo que deben existir verbos que realicen dichas acciones. Por ejemplo, en el sistema de Plan de Ahorro, el Adherente puede ser Adjudicado o Sorteado, ambos estados se asignan dentro del UdeD y su presencia depende de haber ejecutado las acciones Adjudicar y Sortear. En este ejemplo aparece un segundo problema, no asociado directamente a los estados, pero sí al LEL. Puede observarse que el Adherente no participa en los verbos Adjudicar y Sortear, sino que se utilizan sus datos para realizar dichas acciones. La ejecución del verbo tiene como consecuencia que algunos adherentes pasen a ser adjudicados o sorteados. Se debe tener especial cuidado cuando se utilizan indicadores de sujetos y no el sujeto en sí mismo

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

(por ejemplo, en los verbos Adjudicar y Sortear no se utiliza al Adherente sino el código o número que lo identifica). Se sugiere prestar una atención particular a estos casos para no generar inconsistencias en las descripciones.

Un caso particular asociado a los estados es que en algunas expresiones de los clientes-usuarios aparecen unidos al símbolo base, por ejemplo “Factura anulada”. En estos casos, puede suceder que realmente sea un único símbolo o que se trate de un problema enunciativo, ya que el cliente-usuario está tratando de describir lo que hace de la mejor manera posible. Si realmente es un único símbolo en el contexto, no es un estado del LEL, en este ejemplo es un objeto. Si el ingeniero/a de requisitos detecta que se trata de un estado, debe separar el símbolo base del estado. Como ya se mencionó, es indispensable identificar como es utilizado en el contexto. Por ejemplo, en un Sistema Contable “factura anulada” es un objeto, mientras que, en un Sistema de Facturación, existe el símbolo objeto “Factura” y el símbolo estado “Anulado”. Es necesario identificar si se está en el primer caso o en el segundo para no cometer errores y que reflejen el léxico del contexto real.

Cómo mínimo los estados ocurren en pares, aunque pueden ser ternas, cuaternas o grupos más numerosos. Es posible, especialmente cuando se trata de dos estados que uno de ellos no esté visible. Por ejemplo, puede ocurrir que sólo sea perceptible el estado “vacío”, pero para que este estado exista debe existir el estado “lleno”, aunque no se lo identifique como tal.



Figura 72 – DER¹³ de símbolos Estados

En la Figura 72, existe un caso que requiere un tratamiento particular y es el caso de que un mismo estado aplique a más de un símbolo base. Como ya se mencionó, deben ser descriptos como homónimos, pero es interesante analizar por qué. Cabe destacar que es posible que en determinados contextos exista un estado único que aplique a varios símbolos base, pero es una excepción y debe ser considerada como tal. En general, cada símbolo estado tiene particularidades que le corresponden específicamente a cada

¹³ El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos [Chen76]

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

símbolo base con el que se relaciona. Por ejemplo, las razones, la forma y los símbolos verbos involucrados en el pasaje al estado anulado de una “Factura” seguramente son muy diferentes del pasaje al estado anulado de un “Remito”. Tratarlos como un único estado obliga a ignorar esas particularidades y requiere levantar el nivel de abstracción de la definición del estado para generalizarlo. Este mecanismo se debe evitar ya que se pierde información valiosa del contexto. Lo contrario a generalizar es incluir todos los casos posibles en la misma definición. Este caso puede generar confusión y una gran dificultad de lectura, además de esconder la realidad del contexto. Por todo lo mencionado, para asegurar la calidad del LEL, es necesario describir cada estado en forma independiente dejando muy claro el vínculo con su símbolo base y las formas en que se producen los cambios de estado.

Los Estados toman relevancia en el LEL_R (ver **Capítulo 15**) por la cercanía de los requisitos del software y su incorporación aseguran que han sido contemplados a la hora de especificarlos.

Otro factor externo son los *puntos de vistas de contexto*, los cuales pueden ser relacionados con los utilizados en el género literario donde un punto de vista significa la postura desde la cual se desarrollan los acontecimientos o desde qué posición, tanto espacial como temporal, y con qué nivel de conocimiento se narran los hechos que conforman la ficción. En el caso del Proceso de Requisitos, es la postura que adopta la fuente de información para describir como se conforma el UdeD.

El LEL es una unidad que determina una dimensión semántica del proceso del negocio e incluye los aspectos y componentes cognitivos conscientes del vocabulario utilizado para describirlos.

La importancia de los *puntos de vista* es la posibilidad de visibilizar conflictos en la ejecución de las tareas. De esta manera, con un tratamiento apropiado se podrá generar un sistema más adecuado a las necesidades del contexto.

Los puntos de vista existen principalmente en las fuentes de información: personas, bibliografía del dominio, sistemas informáticos, etc. Los puntos de vista que se analizan en esta sección son:

- el “*Es*”,
- el “*Deber ser*” y
- el “*Próximo*”.

El punto de vista “es” y “deber ser” se diferencian por “cómo se debe” realizar una tarea y “como se realiza realmente”. No identificarlos suele deberse a que se esconde la falta de cumplimiento de normativas organizacionales o la falta de aclaraciones acerca de situaciones excepcionales, que pueden ser significativas para el éxito y hasta para la supervivencia de la organización o para el éxito o fracaso del futuro Sistema de Software. De no explicitarlos en el LEL, puede hacer muy dificultoso visibilizar estos problemas. El ejemplo de la Figura 73 muestra la dificultad de identificar un conflicto en un caso real acerca de tomar una muestra en un laboratorio farmacéutico. Este caso data del año 2003 donde existía una disposición de A.N.M.A.T. (Asociación Nacional de Medicamentos, Alimentos y Tecnología Médica) que determinaba los pasos obligatorios que se debían realizar para tomar una muestra farmacológica. El procedimiento determinaba que se debía etiquetar el envase de la muestra en el mismo lugar físico donde se encontraba el tonel con la droga y escribir a mano alzada los datos de identificación de la muestra copiando del tonel el nombre del producto farmacéutico, número de partida (o número de lote), sustancia o estándar de referencia, sustancia o estándar de referencia primaria, sustancia o estándar de referencia secundaria, concentración, fecha de vencimiento, etc., en la etiqueta. Una vez tomada la muestra se debía completar el libro de muestras, también manuscrito, agregando datos como la fecha del día, la hora y el nombre de quién realizó el procedimiento. Esta guía indicaba que por cuestiones de seguridad, sólo se debía tomar una muestra por vez, tal como se enuncia en el primer impacto de la Figura 73.

TOMAR MUESTRA

Noción

- Lo realiza el [técnico](#) cada vez que llega al laboratorio un [insumo\(1\)](#) para [elaborar](#)

Impacto

- El [técnico](#) toma un único [envase de muestra](#)
- le pega una [etiqueta de análisis](#) vacía
- Va junto al tonel
- Junta una [muestra](#) en el [envase de muestra](#)
- Completa de forma manuscrita la [etiqueta de análisis](#) copiando los [datos del tonel](#)
- Llena el [libro de muestra](#)
- Almacena el [envase de muestra](#) en el [armario de muestras](#)

Figura 73 – Ejemplo del LEL con punto de vista (Caso SCC)

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

Esto fue lo que describió el Director de Control de Calidad del Laboratorio. Luego, se comprobó que esto no se cumplía debido a que la distancia entre el lugar donde estaban los envases (oficina técnica) y el lugar donde estaba el tonel (depósito) era de aproximadamente 100 m. Una vez tomada la muestra el técnico debía retornar al punto de partida ya que en la misma oficina técnica también se almacenaban las muestras, provocando un desvío importante entre lo operativo y lo normativo. El responsable de las muestras solía ir con varios envases previamente etiquetados y tomar más de una muestra por vez. Este conflicto entre lo que *se hacía* y lo que *se debía hacer*, ponía al laboratorio en un serio riesgo ya que de saberse serían seriamente multados, llegando en el peor de los casos a clausurar el laboratorio, con el agravante de que la dirección desconocía lo sucedido. Esto deja muy en claro la importancia de detectar y atender correctamente estos puntos de vista.

El punto de vista “**Próximo**” se refiere a algo que sucederá de manera inminente. Es una suerte de presente extendido, analizado desde la perspectiva del tiempo y condiciones de desarrollo. Se refiere a todo aquello que no existe todavía y que puede o no caer dentro del ámbito del uso del nuevo sistema informático. Si algo va a cambiar en dos semanas y los desarrolladores recién están con el LEL, debe tratarse como presente. Cabe destacar que no está referido a cosas que se esperan para el nuevo Sistema de Software, ya que esas afirmaciones deben estar en las Fichas de Información Extemporáneas (ver **Capítulo 19**), sino que son cuestiones referidas a aspectos que se conocen. Por ejemplo, si el LEL describe acciones realizadas sobre un pizarrón con tiza, pero la organización ya compró pizarras electrónicas, esa información es de utilidad cuando se describan algunos procesos futuros o servicios del nuevo Sistema de Software. En este caso, este punto de vista se debe tratar como si fuera presente. También puede ocurrir que se refiera a ciertas características que se planifican en el contexto para cuando el Sistema de Software esté en servicio. Por ejemplo, el cliente indica que un local de ventas se clausurará y el nuevo recién estará listo para cuando se instale el nuevo Sistema de Software, el cual tendrá privados para la atención individual de los clientes. Esto claramente será un condicionante para los Escenarios Futuros, pero no tiene nada que ver con los Escenarios Actuales. Incluir los puntos de vista del contexto en el LEL completa la información existente y ayuda a identificar conflictos.

Se espera que durante la etapa de *Planificar el UdeD futuro* los puntos de vista del contexto sean resueltos para clarificar los requisitos del software. Por lo tanto, la aparición en el LEL_R puede actuar como una alerta de aspectos no tratados adecuadamente o decisiones pendientes.

El último factor externo que se describe en esta sección se refiere a las **jerarquías conceptuales** propias del léxico [Kaplan17]. Como en todos los símbolos, deben existir como tal en el UdeD. Estas jerarquías del contexto y omnipresentes en todos los léxicos, no suelen ser percibidas fácilmente.

En algunos contextos, las jerarquías conceptuales por taxonomía pueden ser confundidas con estados. Esta confusión se genera por la falta de claridad en distinguir una definición de la otra. Tal es el caso de la Figura 74 donde un hombre o una mujer enviudan. La diferencia depende de cómo es utilizado en el contexto. Por tal razón, se dan a continuación, dos textos donde se trata la viudez de forma diferente dando lugar a diferenciar la jerarquía conceptual del estado. En el primer texto se trata la viudez como una taxonomía:

*“Entonces acababa de cumplir 20 años, era **viudo** reciente y rico, estaba deslumbrado por la coronación de Napoleón Bonaparte, se había hecho masón, recitaba de memoria en voz alta sus páginas favoritas de Emilio y La Nueva Eloísa, de Rousseau, que habían sido sus libros de cabecera durante mucho tiempo, y había viajado a pie, de la mano de su maestro y con su morral a la espalda, a través de casi toda Europa”*

Gabriel García Márquez

En este caso el hecho de por qué es viudo es totalmente irrelevante. Lo que importa son las características que se adicionan a la generalización de HOMBRE y corresponde a los 20 años del hombre y a su condición de rico. El peso para identificar una jerarquía conceptual está en la complementación de información (estructuras de datos, acciones, etc.).

Sin embargo, otra forma de pensar la viudez es la siguiente:

*“Ser **viuda** en India significa estar muerta en vida. En el instante que muere el marido empieza una terrible condena; les raparán el pelo, las vestirán siempre de blanco, no llevarán joyas, les despojarán de todas sus posesiones, de su estatus social, serán repudiadas por toda su familia: dicen que traen mala suerte y que incluso su sombra da mal augurio: son una maldición. Y todo este calvario por el simple hecho de haberse quedado viudas”*

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

En este caso, es el hecho de la muerte del marido es lo que marca que las mujeres sean discriminadas. Por lo tanto, en los estados, el hecho que determina la existencia del estado es absolutamente relevante. El peso está puesto en el contexto, en este caso, el cultural. Ambos casos puede observarse en la Figura 74 y determinar las diferencias requiere de un análisis más profundo que la sola identificación del símbolo.

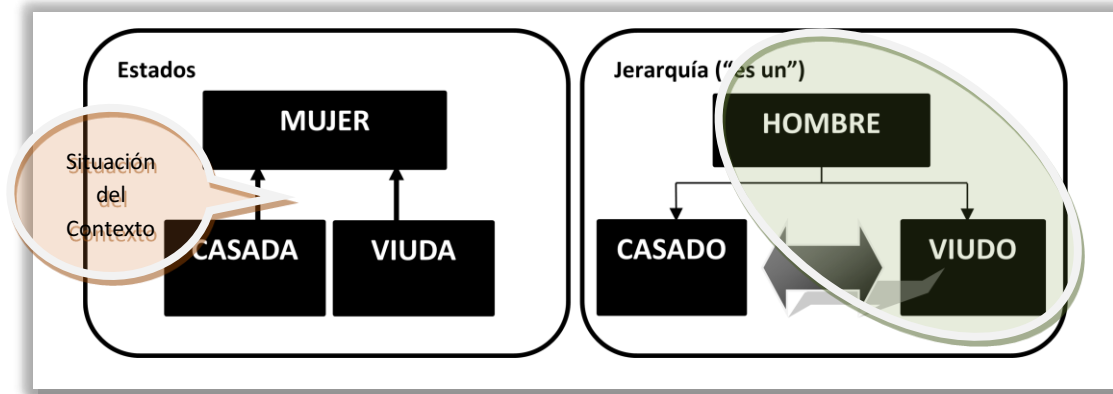


Figura 74 – Ejemplo de Estado y Jerarquía del Contexto

Es interesante pensar cuál es el grado de dependencia entre estados o especializaciones y su símbolo base o generalización respectivamente. En el caso de los estados, la omisión del símbolo base hace desaparecer automáticamente a todos los estados. Los estados no tienen oportunidad de vivir si no existe el símbolo base. En el caso de las jerarquías, la omisión de la generalización es absorbida por las especializaciones, por lo tanto, no existe dependencia. Sin embargo, es importante insistir que eso es una representación errónea del glosario que se está construyendo.

La omisión de los estados no afecta la consistencia del LEL, pero sí perjudica su completitud. Con lo cual existen peculiaridades del contexto donde se desempeñará el futuro sistema que se ignoran.

A diferencia de los estados, los elementos de las jerarquías tienen un menor grado de dependencia entre todos sus elementos. Si la jerarquía se aplana omitiendo, por ejemplo, el genérico, se debe repetir la información del genérico en cada especialización y la información del glosario se mantiene inalterable.

Por otro lado, si las especializaciones se omiten, todas las descripciones de las especializaciones deben ser absorbidas por el genérico. En cualquiera de estos casos se produce una representación confusa del léxico que es perjudicial para su comprensión.

Las jerarquías pueden provenir de relaciones de composición, jerarquías mereológicas, o de relaciones esenciales o taxonomías [Smith98] [Varzi94].

Un estudio preliminar de las jerarquías observables realizadas sobre aproximadamente 200 LEL de diferentes contextos, ha mostrado una preeminencia notoria de las taxonomías (“es un”) sobre las jerarquías mereológicas (“es parte de”). Un ejemplo de la existencia de estas jerarquías del contexto se puede observar en la Figura 75, donde en una primera lectura del Caso de Cartón Corrugado¹⁴ se encontraron términos que no tienen relación entre ellos. El análisis de estos términos permitió visualizar una taxonomía con un genérico próximo “producto” y las diferencias específicas “caja” y “accesorio” y las sucesivas diferencias específicas “caja de tapa y fondo”, “caja de solapa”, “divisor” y “separador”.

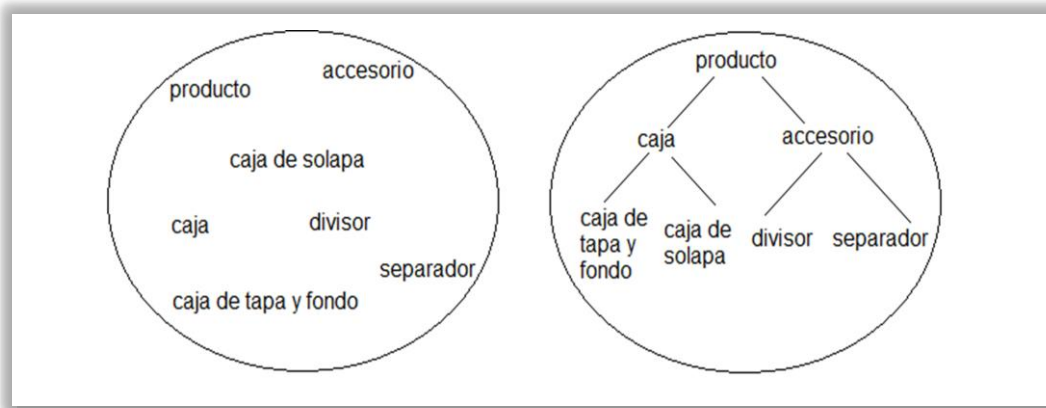


Figura 75 – Ordenamiento del léxico sin y con jerarquías (Caso SNor)

La omisión o el desconocimiento de la existencia de las jerarquías en el léxico del macrosistema pueden incorporar imprecisión y ambigüedad a los modelos construidos ya que son un elemento necesario de clasificación y ordenamiento semántico y por lo tanto de completitud. Es necesario contar con un mecanismo que las visibilice y permita registrarlas en el glosario lo antes posible. En muchos casos, al avanzar en el desarrollo del software, estas jerarquías aparecen claramente visibles en las clases del software. Lamentablemente en este momento ya es muy tarde, porque la información que se puede obtener en ese momento es menos confiable y posiblemente extemporánea, con las consecuencias que eso provoca.

Las jerarquías del contexto se presentan de dos maneras extremas: completas e incompletas. Las más preocupantes son las incompletas, donde se puede presumir la omisión de información necesaria para asegurar la calidad de los requisitos del software.

¹⁴ Caso de Cajas de Cartón Corrugado es un caso de estudio presentado en un documento escrito que describe la planificación de la producción de una empresa que vende cajas y accesorios de cartón.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

En la Figura 76 se analizan todos los casos posibles de jerarquías en el LEL, completas e incompletas. Estrictamente idea de jerarquía completa utilizada en este punto es bastante restrictiva ya que se dice que una jerarquía es completa cuando se cumplen las siguientes condiciones:

- i) El genérico de la raíz es alguna de las hojas de la jerarquía propia de los símbolos del LEL, o sea que corresponde a su clasificación (sujeto, objeto, verbo o estado).
- ii) Todos los nodos de la jerarquía poseen al menos dos hojas o no poseen ninguna.

Los ejemplos de la Figura 76 han sido simplificados ya que eluden a representar jerarquías de más de dos niveles y con más de dos especializaciones. Poca información nueva puede deducirse del caso representado en la Figura 76a) que representa una jerarquía completa, mientras que los restantes casos ameritan un estudio más profundo. La Figura 76b) muestra la situación donde se presume que los símbolos del LEL tienen un genérico próximo común no identificado. Subyace la pregunta ¿existe alguna palabra o frase que deba ser identificada como símbolo del LEL que generalice a este subconjunto de símbolos? Por su parte el esquema contenido en la Figura 76c) visualizan la situación que se produce cuando se detecta el genérico y sólo algunas de las especializaciones. Nuevamente subyace una pregunta ¿faltan elementos de la especialización que sean símbolos del LEL no identificados? Finalmente en los casos Figura 76d) y Figura 76e) se presentan cuando se detectan símbolos aislados y la pregunta es ¿son parte de alguna jerarquía del LEL? En síntesis, la pregunta que subyace es si existe algún símbolo del LEL que complete una diferencia específica o una generalización no detectada.

Cuando se trata de una taxonomía (“es un”), el caso de la Figura 76c) se manifiesta por la presencia de sinónimos parciales; en otras palabras uno de ellos se puede utilizar en reemplazo de otro en algunos casos pero no en todos. Esto se debe a que el grado de especialización de uno de ellos es superior al del otro. En este caso no es necesario haber percibido la existencia de otra especialización ya que la sola condición de sinónimo parcial es suficiente para caracterizar el caso. Si se trata de una relación mereológica (“es parte de”), el caso Figura 76c) se manifiestan cuando se detecta un componente pero faltan otros.

Obviamente cualquiera de los casos incompletos de la Figura 76 puede ser un indicio de omisiones en la construcción del LEL y ser un mecanismo de alerta para revisar la información disponible o para buscar más información en el macrosistema. Este último

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 10 - Estudio empírico del LEL

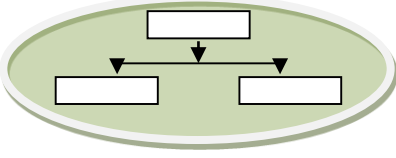
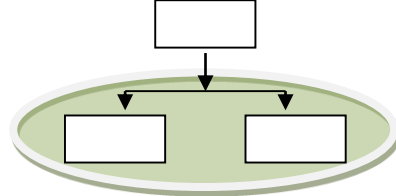
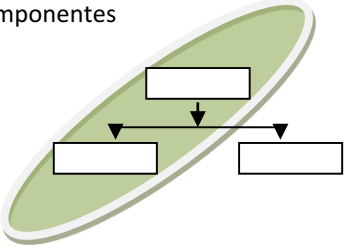
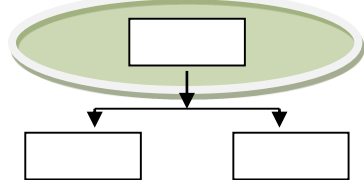
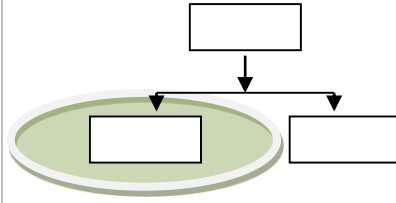
caso es donde puede existir una fuente adicional de información cuya lectura no es obvia que indique la necesidad de una apertura del alcance del macrosistema no identificado hasta este momento.

Es importante tener en cuenta que una persona entrevistada tendrá la perspectiva del macrosistema según el *rol o función que desempeñe*. En el caso de los documentos, estos tendrán la perspectiva del objetivo para el cual fueron creados y para quienes está destinado. Estas particularidades del origen de la información, suelen ser el desencadenante de la aparición de genéricos o de especializaciones. Es de esperar que en los roles directivos o gerenciales aparezcan con mayor frecuencia los genéricos, mientras que en un rol operativo la descripción se concentre más en el detalle.

Las **jerarquías incompletas** se deben en ocasiones, a una determinación de los límites del contexto que ha dejado afuera aspectos que no son de interés para el Sistema de Software a construir.

Se debe prestar una atención singular a la aparición de jerarquías, ya que a veces se esconden en el resto del léxico. Aparecen durante las entrevistas o lectura de documentos, pero también puede ser necesario hacer un análisis posterior para asegurarse que no quedaron jerarquías ocultas. Por lo tanto, existen dos momentos donde detectar las jerarquías: durante la construcción del LEL y post construcción del LEL.

Representar las jerarquías conceptuales del contexto tempranamente ayuda a mejorar la precisión de las descripciones de todos los modelos, pero incorporarlas en el LEL_R puede ser utilizado como un mecanismo de verificación acerca de su incorporación en los requisitos del software, asegurando el apropiado nivel de detalle de los mismos.

<p>a) Jerarquía Completa</p> 	<p>Todos los componentes de la jerarquía fueron identificados como símbolos del LEL. Este caso puede surgir naturalmente del contexto o llegar a él a través de un refinamiento de cualquier de los casos de jerarquías incompletas.</p>
<p>b) Jerarquía Incompleta: ausencia del genérico o agregado más próximo</p> 	<p>Se tienen dos símbolos propios del contexto representando palabras o frases con significado especial que resultan ser especializaciones o componentes de una palabra o frase que no se usa en el contexto o se usa sin un significado particular. Este caso ocurre probablemente como consecuencia de la forma en que se ha delimitado el contexto para realizar el Proceso de Requisitos o la omisión del uso, intencional o involuntario de un agregado o un término genérico por parte de los clientes-usuarios.</p>
<p>c) Jerarquía Incompleta: omisión de otras especializaciones o componentes</p> 	<p>Se han identificado dos o más símbolos con una relación jerárquica en la cual se deduce la existencia de otras palabras o frases del contexto no identificados como símbolos. Este caso ocurre probablemente como consecuencia de la forma en que se ha delimitado el contexto para realizar el Proceso de Requisitos. Es probable que este caso requiera especial atención si en algún momento del Proceso de Requisitos es necesario ampliar los límites del sistema.</p>
<p>d) Jerarquía Incompleta: ausencia de todas las especializaciones o componentes</p> 	<p>Se ha identificado una palabra o frase con significado particular en el contexto que describe una composición semántica con otras palabras o frases no identificadas como símbolos del LEL. Este caso ocurre probablemente como consecuencia de la forma en que se ha delimitado el contexto para realizar el Proceso de Requisitos o de la irrelevancia de ciertos componentes o términos específicos para el Proceso de Requisitos.</p>
<p>e) Jerarquía Incompleta: ausencia del genérico más próximo o del agregado y de otras especializaciones o componentes</p> 	<p>Se está en presencia de una palabra o frase con significado especial en el contexto que idéntica en su noción un elemento genérico o que es parte de un agregado. Este caso tiene lugar como consecuencia de la combinación de los factores que dan lugar a los casos b) y c).</p>

El óvalo sobre un rectángulo indica que el elemento es un símbolo del LEL detectado. Los otros rectángulos (sin marcar por el ovalo) indican la posible existencia de otros elementos de la jerarquía no identificados en el LEL.

Figura 76 – Tipos de jerarquías conceptuales del contexto

Capítulo 11

Estudio empírico de los Escenarios

Resumen

A diferencia del LEL, no existen estudios previos que identifiquen defectos en los Escenarios. Sin embargo, el estudio empírico que se presenta a continuación, ha determinado algunos problemas de gran relevancia, principalmente en lo referido a la completitud.

Como ya se mencionó, la *elicitación guiada por los modelos* puede verse afectada cuando existe un gap conceptual entre las características del contexto, la visión intrínseca del modelo y la perspectiva del ingeniero/a de requisitos. Las características del contexto están muy influenciadas por la naturaleza del proceso del negocio, siendo en ocasiones dominante los procesos, en algunas los productos, en otras las interacciones internas, entre varias más. En este aspecto pueden existir dos posibilidades, que todas las visiones sean compatibles o que discrepen. El primer caso es el ideal, ya que la información fluye con mayor facilidad y la representación no sufre alteraciones. Usualmente, la visión del ingeniero/a de requisitos se adapta razonablemente al enfoque del modelo, algo que no ocurre con la narrativa del proceso del negocio. Cuando existen diferencias, se debe realizar un esfuerzo adicional debido a la necesidad de transformar la visión del macrosistema al enfoque del modelo. Si bien los modelos del Proceso de Requisitos que se utilizan en esta tesis tienen su propia perspectiva, en algunos aspectos permiten soportar cierta diversidad de enfoques. Esto sucede porque todos los modelos, y los Escenarios en particular, son construidos en lenguaje natural. Esta flexibilidad puede ser vista en primera instancia como una ventaja, pero en realidad incrementa la responsabilidad del ingeniero/a de requisitos ya que debe tener especial cuidado en la redacción.

Factores internos que afectan la calidad de los Escenarios

Los factores internos son aquellos que se encuentran dentro del proceso de construcción de los Escenarios.

Como ya se mencionó en el **Capítulo 8**, los EA y los EF comparten muchas actividades del proceso de construcción. La principal diferencia radica en la perspectiva con el cual se construye cada conjunto de Escenarios, siendo que el primero de ellos observa el proceso del negocio en el momento de realizar la definición de los requisitos y el segundo, planifica cómo será el proceso del negocio con el Sistema de Software en ejecución. Cada perspectiva tiene una heurística de construcción propia, aun así, comparten el modelo y varias actividades. En ambos casos, existe una primera actividad de suma relevancia, ya que es donde se construye el primer conjunto de Escenarios. En el caso de los EA, esta primera actividad se denomina Derivación desde el LEL (ver **Capítulo 8.1**), mientras que para los EF consiste en una transformación de los EA a EF (ver **Capítulo 8.2**). En ambos

casos se construye una primera versión del conjunto de Escenarios a partir de la información existente en los modelos construidos, garantizando el avance del proceso sobre información validada. A pesar de haber tomado en cuenta todos estos aspectos, los resultados obtenidos no han sido los esperados, en particular en la actividad Derivar desde el LEL. Esta derivación genera una primera versión de EA denominados “Escenarios candidatos derivados” (ver Figura 52). En esta actividad se propone generar una lista de situaciones del contexto obtenidas directamente desde el LEL. Utilizar el glosario sin ningún tipo de procesamiento, produce un salto conceptual entre una representación declarativa (LEL) y una procedural (Escenarios). Por otro lado, la granularidad y ordenamiento del LEL suele ser muy diferente de la que existe en el contexto real, ya que los glosarios tienen una gran dispersión de la información, adecuado para su propósito, pero inapropiado para describir el proceso del negocio.

Esta actividad se realiza muy temprano en el proceso de construcción del software, por lo cual, si muestra alguna debilidad ante el cliente-usuario, hará muy difícil de sostener la credibilidad durante el resto del proceso, con las consecuencias negativas que esto puede originar.

Diferentes comparaciones realizadas entre los Escenarios candidatos derivados y los Escenarios Actuales finales han dado una coincidencia cualitativa y cuantitativa muy elevada, la cual se atribuyó en un primer momento, al éxito de la actividad Derivar. En la presente tesis, retrospectivamente y desde un análisis cualitativo de estos dos conjuntos de Escenarios, se sostiene que esta actividad, en vez de ser una ventaja para el proceso de construcción de EA, resulta ser un corsé bastante difícil de aflojar. Como ya se mencionó, la información que proviene del glosario es expresada de manera declarativa y no procedural, como requieren los Escenarios. Pero además, este procedimiento genera una *lista inicial de Escenarios*, cuyos efectos nocivos son similares a los descriptos para la lista inicial de símbolos del LEL (ver **Capítulo 10**).

Utilizar el LEL casi como única fuente de información para generar la lista inicial de Escenarios candidatos derivados, supone igualar el concepto de “actividades” y “tareas” con “situaciones”. Esto sucede porque los impactos de los sujetos, los cuales describen las actividades y tareas que realizan, son considerados en la derivación existente como potenciales Escenarios, asumiendo que una actividad es igual a una situación. Esta diferencia es observable en la poca información utilizada del LEL en los Escenarios. El

resultado de la derivación produce un conjunto de Escenarios incompletos y desordenados, obligando al ingeniero/a de requisitos a un retrabajo importante. Para obtener los Escenarios reales se debe reorganizar el conjunto de Escenarios candidatos en un nuevo conjunto de Escenarios, obligando a contrastar cada situación obtenida desde el LEL con los procesos reales del negocio. Esta tarea somete al ingeniero/a de requisitos y a los clientes y usuarios a concentrarse más en los desvíos de granularidad y ordenamiento que en la comprensión y refinamiento de cada situación del contexto. De esta manera, la derivación existente termina perjudicando los objetivos originales de la actividad, ya que, en vez de optimizar el proceso, lo perjudica.

La derivación existente ha omitido sistemáticamente la incorporación de los estados en los Escenarios construidos, desaprovechando una información de gran valor para los requisitos del software. Para la presente tesis, además de tomar en cuenta los estados, se debe incorporar la nueva información que contiene del LEL (ver **Capítulo 11**), la cual debe ser ahora analizada desde la perspectiva de los Escenarios. Este es el caso de las jerarquías conceptuales y de los puntos de vista. Las debilidades de la actividad Derivar pueden ser sintetizadas en:

1. La generación de una lista inicial,
2. El salto conceptual de un modelo a otro,
3. La exclusión de los estados y
4. La omisión de los puntos de vista y de las jerarquías conceptuales.

Otro factor interno a considerar en los Escenarios, es la necesidad de ***aumentar la expresividad de los episodios***. Los tipos de episodios que existen son los denominados: acciones simples, condicionales, opcionales, sin secuencia y los que representan la jerarquía de Escenarios (sub-Escenarios) (ver **Capítulo 8**). Estos episodios abarcan gran número de casos, pero no permiten representar algunas particularidades que son necesarias para comprender cada situación. Por lo tanto, estas limitaciones en la representación de los episodios, puede generar confusiones y en el peor de los casos, interpretaciones erróneas acerca del proceso del negocio (actual y futuro). Siendo que, los EF contienen los requisitos del software, es necesario que los mismos sean correctamente expresados y comprendidos.

Una limitación que tienen los Escenarios es que no se puede indicar cuando un episodio puede ser realizado en cualquier momento del Escenario, ignorando la secuencia de los episodios. Este caso es diferente de los episodios sin secuencia identificados con los numerales (#) ya que estos indican paralelismo o concurrencia, mientras que aquellos son episodios que se pueden realizar en cualquier momento del Escenario.

Finalmente, se debe analizar si es necesario identificar los símbolos del LEL que aparecen en el título de cada Escenario, o sea el *empotramiento en el título del Escenario*.

Puede observarse en el estudio empírico que algunos autores de los casos existentes han marcado estos vínculos internos del título. El título del Escenario es un nombre propio y, por lo tanto, podría prescindir de este marcado. Pero no hacerlo sería ignorar la relación estrecha de los Escenarios con el LEL y esconder información valiosa que puede aportar este vínculo mejorando la comprensión del Escenario.

Existe un problema que se presenta al describir los Escenarios y está asociado al *uso correcto del LEL*. Esto sucede porque la redacción de cada episodio puede arrastrar algunos inconvenientes propios del uso del lenguaje natural. Como ya se mencionó, si por cada símbolo que se desee relacionar se consulta su definición, su incorporación será segura. Pero también, a esta altura del proceso donde ya se construyó el LEL, el ingeniero/a de requisitos se siente confiado/a para utilizar el léxico del macrosistema y eso lo pone en una posición más riesgosa. Entre los errores que pueden suceder se encuentran los mencionados en el **Capítulo 10** y desde ahora, con las mejoras incorporadas al LEL, algunos más:

- No recordar que un símbolo es parte de una jerarquía. En este caso la inclusión puede ser parcialmente correcta, pero con un riesgo muy alto de cometer errores graves.
- Que el símbolo tenga homónimos y se utilice uno equivocado.
- No recordar los puntos de vista y omitirlos en los Escenarios.
- No recordar los estados asociados y omitir detalle o episodios.
- Incluir por aproximación fonética sin tener en cuenta que pueden existir símbolos diferentes muy parecidos, usualmente de distinto tipo.

A continuación, se ejemplifica el último ítem, que es un problema muy común en los Escenarios. Si en un contexto (ver Figura 77) existe el verbo “Autorizar”, el objeto “Autorización” y el estado “Autorizada” se suelen encontrar usos incorrectos o dudosos. Puede observarse que en el uso de “autorizada” se refiere en el último episodio hace referencia al estado cuando debe hacerlo al verbo o al objeto. En este caso puede suceder que sea sólo un error de redacción o que sea un error de vinculación entre el Escenario y el LEL. En cualquiera de los casos, es necesario contar con algún mecanismo que permita visibilizar, de una manera rápida y simple, si el uso de los símbolos del LEL es correcto.

Liberar PO

Objetivo: Garantizar que [gerencia](#) recibe las [PO](#) sin observaciones

Contexto:

Ubicación Geográfica: PC

Ubicación Temporal:

Precondición: Debe existir [PO](#) sin [autorización](#)

Recursos: [PO](#) (no debe estar [liberada](#))

Actores: Sistema

Episodios:

- El **Sistema** selecciona las [PO](#) sin [autorizar](#)
- Selecciona una [PO](#) y controla si tiene algún [modelo observado](#)
- Si no existe ninguna observación entonces [libera](#) la [PO](#) para ser [autorizada](#) en [Gerencia](#)

Excepciones:

Figura 77 – Ejemplo de Escenarios con usos incorrectos o dudosos del LEL (Caso SMAC)

Finalmente, existe una omisión con respecto a las *restricciones a los actores*. Se puede observar que las Restricciones de los Escenarios aplican al Contexto, a los Recursos y a los Episodios. Con respecto a las Restricciones de los recursos sucede algo muy particular, ya que se debe elegir si poner la restricción asociada directamente al recurso o describirla como una Precondición del Contexto. La decisión depende de la visibilidad que se le desee dar. En una mirada rápida a un Escenario, es más fácil visibilizar la Restricción en un recurso que comprenderlo desde el análisis de la Precondición del Contexto. Lo mismo sucede con los actores. Se han detectado casos donde es necesario describir una Restricción al actor, pero el modelo de Escenario no lo contempla. Por ejemplo, si se desea restringir la realización de un Escenario a los enfermeros que están capacitados y certificados para tratar con pacientes Covid, se debe incluir en la Precondición, perdiendo la asociación entre el componente Actores y los Episodios y de esta manera, se muestra claramente que la restricción aplica a todos los

episodios donde el actor tiene un rol. Por otro lado, debe recordarse que los actores que aparecen en la Precondición no son necesariamente actores del Escenario, o sea que pueden no tener participación en los episodios por lo que puede perder fuerza.

Factores externos que afectan la calidad de los Escenarios

Los factores externos son aquellos que influyen en la comprensión de los Escenarios y que se encuentran fuera del proceso de construcción de los mismos.

Los *Estados*, como ya se mencionó, deben ser contemplados en la derivación, pero luego, deben ser analizados con detenimiento durante la construcción de los Escenarios. Esto implica que el ingeniero/a de requisitos debe estar atento a la aparición de posibles nuevos estados y a la inclusión de los estados existentes en el LEL en todas las situaciones pertinentes. Cuando aparece un estado nuevo se debe retornar al LEL para incorporarlo. Por lo tanto, esta relación estrecha con el LEL genera un reciclaje entre ambos modelos que no sólo incluye a los símbolos tipo estados sino a cualquier símbolo nuevo que se identifique durante la construcción de los Escenarios.

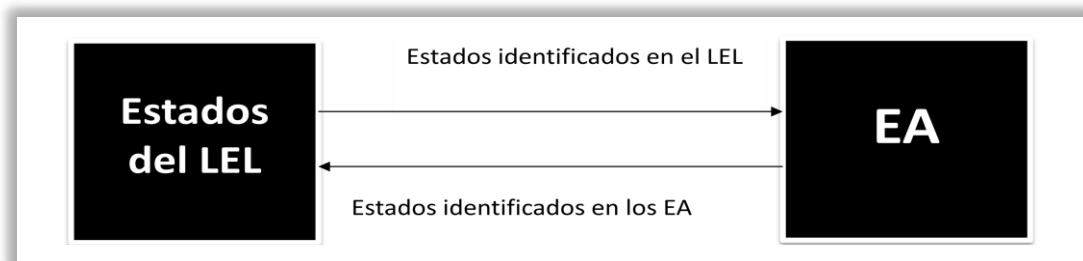


Figura 78 – Relación entre estados del LEL y los EA

En la Figura 78 se puede observar que la presencia de estados debe estar equilibrada entre el LEL y los EA. Los estados del LEL deben estar reflejados en los EA, mientras que los nuevos estados que se detecten, deben ser, en primera instancia, incluidos al LEL y luego, se deben analizar todos los EA existentes para determinar en qué situaciones influye.

Diferente es el tratamiento de los estados en los EF ya que por cada estado de un EA se debe analizar su continuidad en el UdeD futuro (ver Figura 79). También es esperable que aparezcan nuevos estados durante la construcción de los EF, propios del UdeD futuro.

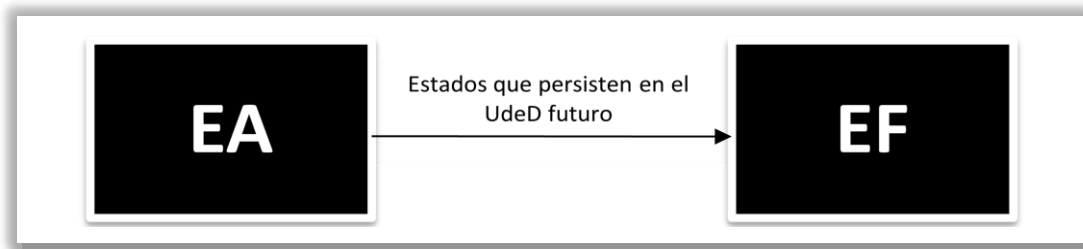


Figura 79 – Relación entre los estados del UdeD actual y del UdeD futuro

La construcción del LEL, además de ser un elemento indispensable para reducir la ambigüedad, actúan como el primer delimitador del alcance del contexto. Cabe recordar que las primeras situaciones que se describen los EA se obtienen del LEL y que sus descripciones deben maximizar su uso. Esta relación estrecha determina que las omisiones en el LEL repercuten severamente en los EA, principalmente en la primera versión de estos. Luego, los EF se construyen tomando como referencia a los EA, por lo tanto, las omisiones o malos entendidos en los EA seguramente llegarán a los EF y de éstos a los requisitos del software.

La reciprocidad entre el LEL y los EA se controla en la Verificación de Escenarios en el formulario de inspección “Pág. X. SÍMBOLOS DEL LEL NO UTILIZADOS” (Ver Anexo A) que detecta omisiones en los Escenarios a partir cubrimiento del LEL.

En muchos casos los estados son incorporados a los EA luego de la Derivación, ya que este proceso no los incluye. Luego, se verifican al analizar el cubrimiento del LEL en los Escenarios.

A continuación, se describen algunas observaciones obtenidas del estudio empírico, acerca de los estados en los Escenarios:

- Se debe cuidar que la atención del ingeniero/a de requisitos se concentre en la situación que describe el Escenario, procurando que se distraiga lo menos posible.
- Cada vez que aparece un estado en un Escenario, se debe analizar todo el símbolo estado para saber con qué otros estados se relacionan. De esta manera se pueden detectar episodios y restricciones omitidas.
- Un recurso o un actor puede comenzar un Escenario en un estado particular o dominante. Este estado debe ser incorporado como una restricción para dejar clara esta situación particular (en el caso de los actores es una mejora de la presente tesis que se describe en Restricciones a los actores en el **Capítulo 16**).

Se ha realizado una pequeña experiencia durante la construcción del Caso Caja de Cartón Corrugado, donde se les ha advertido a los ingenieros/as de requisitos acerca de la existencia de un estado específico. A partir de esto, se observó una mejora en el tratamiento de estados en otros Escenarios. Este episodio advierte que, además de la importancia de los estados para los requisitos, el saber de ellos mejora su tratamiento. De esta manera, se incrementa la importancia de incluir los estados en la Derivación.

Otro aspecto a considerar son los *puntos de vista del contexto* que muestran algunos conflictos que deben ser representados en los modelos del UdeD actual. Estos puntos de vista, comienzan siendo identificados y representados en el LEL (ver **Capítulo 10**), y luego, se los traslada a los EA para que, finalmente, se tome alguna decisión durante la construcción de los EF, siempre y cuando sea posible. Cabe destacar que durante la construcción de los EA no se toma ninguna decisión, sólo se representan los conflictos existentes (ver Figura 80). Las decisiones durante la etapa *Planificar el UdeD futuro* dependerán, por un lado, de los servicios del Sistema de Software y por el otro, de las reglas organizacionales que el software deba satisfacer. En algunos casos los puntos de vista se podrán resolver y en otros, cuando aún no se pueda tomar una decisión, se los mantendrán visibles en los EF.

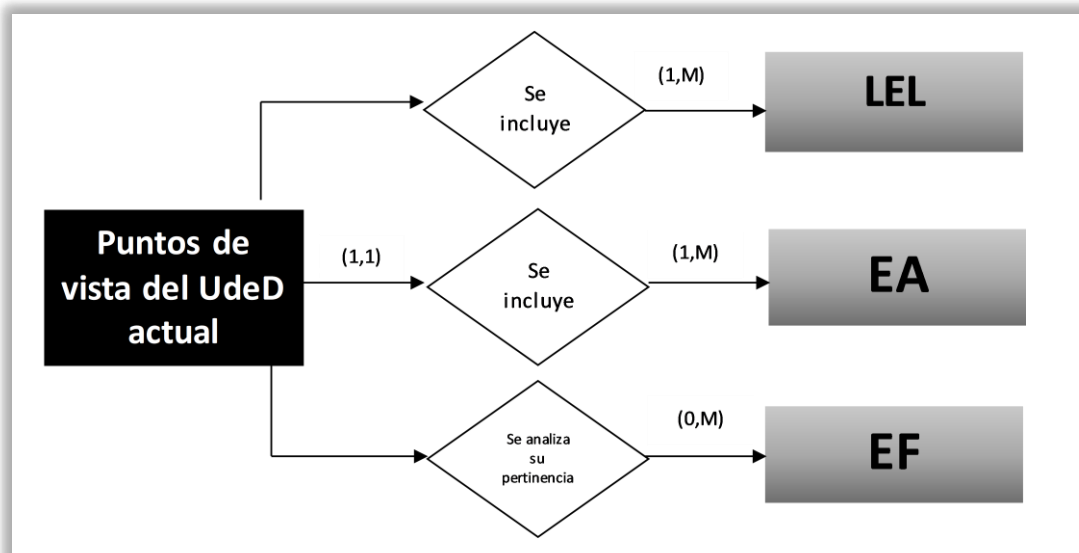


Figura 80 – Modelo DER de la relación “Puntos de Vista- Modelos”

Se puede observar en la Figura 81 que los Puntos de Vista del contexto repercuten en diferentes actividades del Proceso de Requisitos. En principio, se deben tratar en el LEL,

como fue descrito oportunamente en el **Capítulo 11**. La actividad Derivar existente se ha descartado, por lo que no se propone ningún cambio. El tratamiento de los puntos de vista en los Escenarios, se detallan en el **Capítulo 16**.

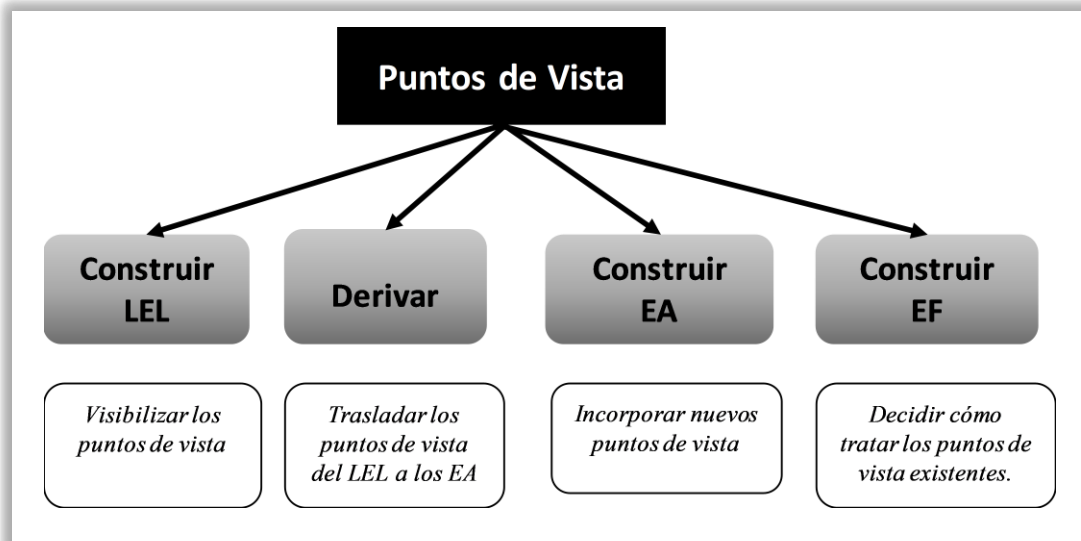


Figura 81 – Tratamiento de los Puntos de Vista del contexto en el Proceso de Requisitos

Los puntos de vista durante la construcción del LEL descritos en el **Capítulo 10** se han incluido en la nueva heurística de *Construcción del LEL por Proximidad* en el **Capítulo 14**. De la misma manera, se han incluido en la nueva heurística de *Derivación de Escenarios Actuales* en el **Capítulo 18**.

Finalmente, se deben analizar las **jerarquías conceptuales**, las cuales ya fueron visibilizadas en el LEL, según lo descrito en el **Capítulo 11**. Igual que sucede con los puntos de vista, durante la descripción de los Escenarios, deben ser tenidos cuenta en la actividad Derivar. Se debe tener en cuenta que la omisión de las jerarquías conceptuales entorpece la comprensión de detalles esenciales del contexto. Por ejemplo, omitir un símbolo genérico de una jerarquía obligará a mencionar todas las especializaciones existentes con el riesgo de olvidarse alguna, además de hacer dificultosa la comprensión del contexto que describen. Por el contrario, utilizar un genérico cuando se debe indicar una especialización, generará malos entendidos ya que, el alcance otorgado no se corresponde con el contexto observable. Esta distorsión de la realidad o falta de precisión, baja la calidad de los Escenarios, lo que repercute directamente en la calidad de los

requisitos del software. En la Figura 82 se muestra un ejemplo de un símbolo del LEL que representa un genérico de una jerarquía que no fue identificado como tal. Luego, en la Figura 83 se puede observar cómo repercute esta omisión en la descripción de un Escenario.

Caja con solapas

Noción

- Son cajas de cartón corrugado cuya tapa y fondo se conforman por cuatro solapas
- Sus laterales son de una sola pieza
- Pueden llevar [accesorios](#)

Impacto

- Se venden
- Pueden ser [a medida](#) o de [tamaño fijo](#)

Figura 82 – Ejemplo de una jerarquía léxica omitida (Caso SNor)

Se puede observar que la omisión de la jerarquía conceptual en la Figura 82 se trasladó al Escenario de la Figura 83 generando una descripción ambigua en el Escenario con efectos no deseados. Debe tenerse en cuenta que además de la falta de precisión visiblemente expuesta, puede haber pérdida de información. En el ejemplo del Escenario se puede observar que en el Título del Escenario y en los episodios se utiliza el término “Caja”, el cual no está definido haciendo correcta alusión a un genérico. Este uso parcial suele ser un claro indicativo de la omisión de una jerarquía. Un problema serio que enfrenta este tipo de omisión, es que en la validación de los Escenarios no suelen ser detectadas, ya que los usuarios comprenden el significado de “caja”. Lo que se desconoce es si algún usuario comprendió que el Escenario era solo para las *Cajas con Solapa*, solo para las *Cajas Tapa y Fondo* o para ambas.

Este ejemplo pone en evidencia como la falta de precisión y completitud del LEL afecta la calidad de los Escenarios. Un LEL pobre obliga a profundizar en las descripciones de los Escenarios, aclarando detalles que deben estar en el glosario. De no ser identificadas estas omisiones del LEL podrían afectar la consistencia de todos los modelos que se construyan.

Fabricar Caja
Objetivo: Elaborar las **cajas** para vender
Contexto:
Ubicación geográfica: Sección de máquinas.
Ubicación Temporal: Turnos de 6 a 14, de 14 a 22 y de 22 a 6 hs.
Precondición: El [Programa de fabricación](#) debe estar elaborado.
Recursos: [Orden de producción](#)
Actores: [Operario](#)
Episodios:

- El [Operario](#) única las maquinas según el [despliegue](#) de la [orden de producción](#)
- corta el cartón en piezas
- marca dobleces al cartón
- imprime laterales de los textos solicitados por el [cliente](#)
- Arma paquetes de 50 a 100 **cajas**

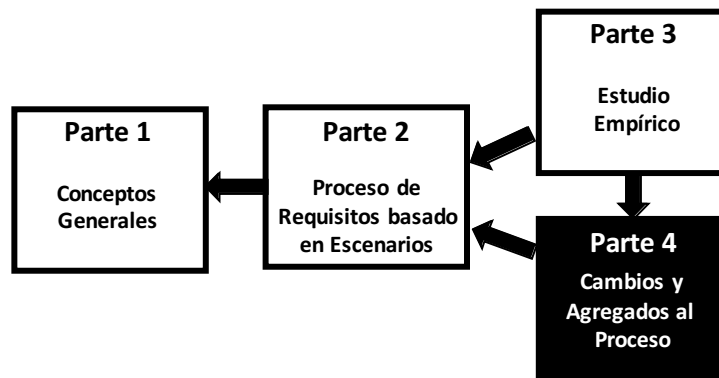
Excepción:

Figura 83 – Ejemplo de uso incorrecto de una jerarquía léxica (Caso SNor)

Las jerarquías conceptuales descritas en el **Capítulo 10** se han incluido en la nueva heurística de *Construcción del LEL por Proximidad* (**Capítulo 14**) y en la nueva heurística de *Derivación de Escenarios Actuales* (**Capítulo 18**).

Parte 4

Cambios y Agregados al Proceso



Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Introducción

En la Parte 3 se describieron un conjunto de debilidades y defectos cuyas soluciones se presentan en los Capítulos que siguen. Por la variedad de los temas mencionados, se presentan a continuación tres tablas las que van de lo general del proceso (Tabla 22) a lo particular de cada modelo (Tabla 23 y Tabla 24). En cada caso se determina el grado de desarrollo que tienen en la presente tesis. En los casos *propuestos*, donde no se han probado aún, los resultados preliminares observados son muy promisorios.

Puede observarse en la Tabla 22 que se han separado en tres grupos. Sin embargo, cabe mencionar que, en la práctica, un cambio en cualquier parte del proceso repercute en otro y esta cadena de mejoras incide en la calidad de todo el proceso.

Capítulos	Cambios	Agregados	Grado de desarrollo
Cambios y Agregados al LEL	X		Parcialmente probado
Actualización de la Inspección del LEL	X		Propuesto
Construcción del LEL por Proximidad	X		Propuesto
LEL de Requisitos		X	Probado
Cambios y Agregados a los Escenarios	X		Parcialmente probado
Actualización de la Inspección de Escenarios	X		Propuesto
Derivación de Escenarios Actuales por Proximidad	X		Propuesto
Información Extemporánea en los procesos de requisitos		X	Probado

Tabla 22 – Cambios y Agregados al Proceso de Requisito

En la Tabla 23 se detallan los cambios y agregados específicos del LEL, mientras que los cambios y agregados aplicados a los Escenarios se detallan en la Tabla 24.

Secciones del Capítulo 12	Cambios	Agregados	Grado de desarrollo
Vista Clasificación		x	Propuesto
Vista Empotramiento		x	Propuesto
Puntos de Vista		x	Propuesto
Sinónimos	x		Parcialmente probado
Homónimos	x		Parcialmente probado
Tratamiento de estados	x		Parcialmente probado
Jerarquías conceptuales		x	Probado
Patrones Sintácticos	x		Propuesto

Tabla 23 – Cambios y agregados al LEL

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Introducción

Secciones del Capítulo 16	Cambios	Agregados	Grado de desarrollo
Nueva Actividad Derivar		X	Propuesto
Vinculación con el LEL	X		Propuesto
Puntos de Vista		X	Propuesto
Restricciones a los Actores		X	Propuesto
Modificaciones al Modelo Escenarios	X		Propuesto

Tabla 24 – Cambios y agregados a los Escenarios

Finalmente, se presenta en la Tabla 25 las publicaciones que se relacionan con las soluciones propuestas en los Capítulos que siguen.

Año	Título	Autores	Lugar	Tipo	Cita
2009	Handling Extemporaneous Information in Requirements Engineering	Kaplan G., Doorn J.	IGI Global Books (Mehdi Khosrow-Pour, Information Resources Management Association, USA)	Encyclopedia of Information Science & Technology	[Kaplan09]
2010	Ambigüedad Léxica en los Modelos de Requisitos en Lenguaje Natural	Kaplan G., Doorn J., Hadad G.	WICC 10	Congreso nacional	[Kaplan10]
2011	Granularidad de la Información Extemporánea en los Procesos de Requisitos	Kaplan G., Doorn J., Hadad G.	WICC 11	Congreso nacional	[Kaplan11]
2013	Construcción del LEL de Requisitos	Kaplan G., Doorn J., Gigante N.	WICC 2013	Congreso nacional	[Kaplan13a]
2013	Evolución Semántica de los Glosarios en los Procesos de Requisitos	Kaplan G., Doorn J., Gigante, N.	CACIC 13	Congreso nacional	[Kaplan13b]
2014	Información Extemporánea en los procesos de requisitos	Kaplan G.	UNLaM	Tesis de Maestría	[Kaplan14a]
2015	Advanced & Delayed Information in Requirements Engineering	Kaplan G., Doorn J.	IGI - Global Books (Mehdi Khosrow-Pour, Information Resources Management Association, USA)	Encyclopedia of Information Science & Technology	[Kaplan15]
2017	Jerarquías Conceptuales en los Procesos de Requisitos	Gladys G., Jorge J.	WER 17	Congreso Internacional	[Kaplan17]

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Introducción

2020	Construcción de grafos de glosarios guiada por el estilo del discurso	Ridao M., Doorn J., Kaplan G.	WICC 20	Congreso nacional	[Ridao20]
2020	Reducción de la Subjetividad en los procesos de requisitos	Kaplan G., Doorn J.	WICC 20	Congreso nacional	[Kaplan20]
2020	Desagregación de la información en los Escenarios Futuros	Kaplan G., Tua D., Blanco G.	WICC 20	Congreso nacional	[Kaplan20]

Tabla 25 – Publicaciones relacionadas con la Parte 4

En la aplicación de los cambios y agregados al proceso se puede observar una mejora sustancial en la calidad de los artefactos obtenidos, especialmente en cuanto a la completitud y consistencia. Estas mejoras tempranas tendrán un efecto dominó (Domino Show) en los modelos que siguen hasta llegar finalmente a los propios requisitos del software.

Capítulo 12

Cambios y Agregados al LEL

Resumen

A partir del estudio empírico del LEL, se han podido determinar algunos cambios y agregados para mitigar los defectos encontrados. Por tal motivo, a continuación, se presenta un conjunto de soluciones que permiten obtener un LEL de mayor calidad, haciendo un especial énfasis en la completitud. Si bien no todas las propuestas han sido probadas, se puede adelantar que todas son altamente confiables.

Síntesis de los cambios y agregados

Todas las soluciones de la presente tesis, tanto para el LEL como para los Escenarios, se mantienen alineadas a dos premisas que se han sostenido desde la creación del Proceso de Requisitos hasta la fecha. Una de ellas se refiere a facilitar la lectura y comprensión de los modelos construidos, principalmente, por parte de los clientes y usuarios. La otra, mantener la simplicidad de la construcción y preservar así la atención del ingeniero/a de requisitos en cada modelo a construir, procurando que interfieran lo menos posible los aspectos de representación. Para facilitar la lectura de los modelos construidos y asegurar la comprensión, es necesario reducir a su mínima expresión la curva de aprendizaje. Este es uno de las ventajas de estos modelos. Este último punto ayuda a eliminar la resistencia natural a lo desconocido y permite una mayor colaboración de los involucrados en todo el Proceso de Requisitos.

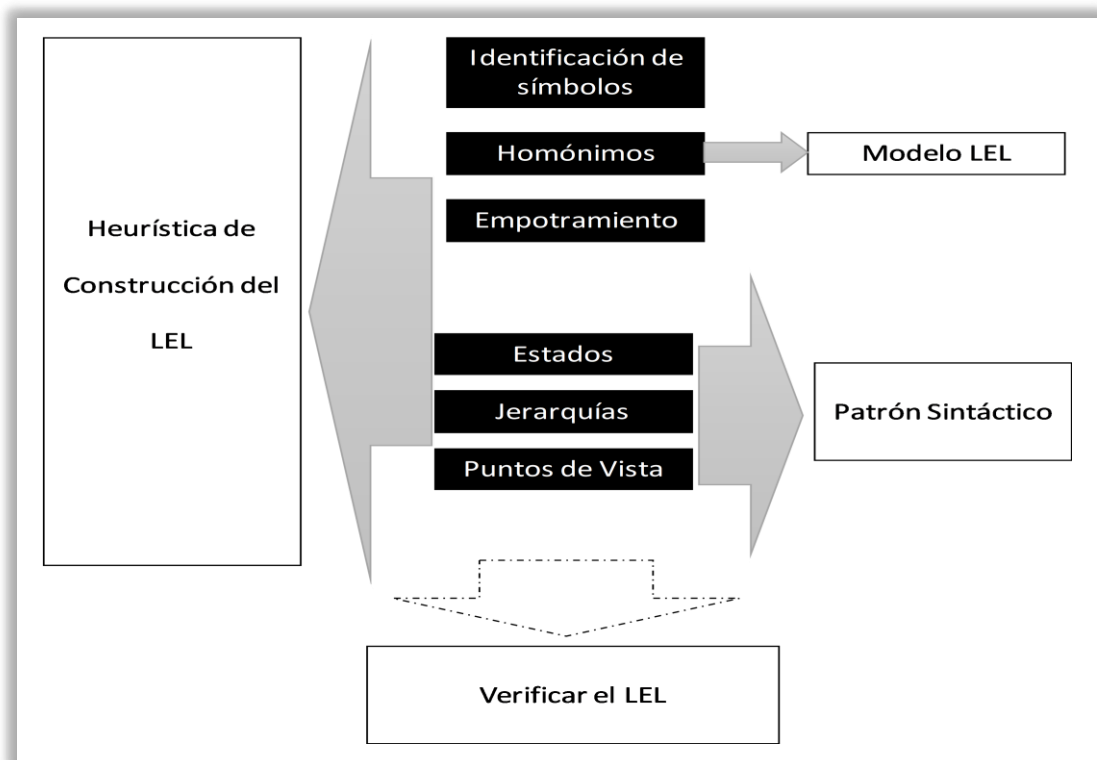


Figura 84 – Síntesis de los cambios y agregados al LEL

Antes de describir cada uno de los cambios y agregados, se realizan dos observaciones referidas a la Figura 84. En primer lugar, aclarar que la Identificación de Símbolos, por la naturaleza del problema que presenta, obligó a repensar toda la heurística. Por tal motivo se describe en el **Capítulo 14** la actividad *Construcción del LEL por Proximidad* la cual reemplaza a la actividad existente. La segunda observación se refiere a la

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

Actividad Verificar, donde las modificaciones necesarias son de tal magnitud que, por cuestiones de espacio y de claridad, no se incluyen en la presente tesis. A modo de ejemplo, se describen en el **Capítulo 13**, algunos formularios de inspección nuevos sólo con el objetivo de presentar la naturaleza de los cambios necesarios.

Vista Clasificación

El Principio de Circularidad del LEL propone que durante la descripción de cada símbolo se maximice el uso de otros símbolos. Esto obliga al ingeniero/a de requisitos a navegar por el LEL para recordar cada definición o estar muy familiarizado con todo el léxico existente. En el primer caso, acceder a cada definición, agrega un esfuerzo adicional en la construcción del glosario. Se debe tener en cuenta que los símbolos más alejados del núcleo del léxico tienen usualmente una sola mención, mientras que los del corazón del léxico suelen rondar los 10. Acceder a otros símbolos puede distraer la atención del ingeniero/a de requisitos, y en el peor de los casos, desorientarlo, un efecto muy conocido en los entornos hipertextuales. Por otro lado, estar familiarizado con el LEL es comprender el léxico del macrosistema, pero no necesariamente conocer con precisión cada definición. Por lo tanto, el primer camino es seguro pero costoso, mientras que el otro, tiene el riesgo de insertar errores. En la práctica, en la vorágine del trabajo, aparece una tercera opción, donde las menciones a otros símbolos suelen ser decididas apelando a la memoria e inclusive, a lo intuitivo. Entre los problemas más frecuentes se encuentra la inclusión de símbolos por la interpretación de su nombre, aludiendo a una apreciación espontánea, intuitiva o conocida. Por ejemplo, en el Caso SOP el Sector “Caja” es utilizado en el contexto como sujeto:

La [caja](#) emite el comprobante de pago

En este ejemplo, no existe ningún error, pero el siguiente uso del símbolo “caja” presenta una inconsistencia:

El [solicitante](#) paga el trámite en la [caja](#)

En este caso, “caja” está mal utilizado, pero no será fácilmente percibido ni por los desarrolladores ni por los clientes-usuarios, ya que se le asigna el significado más

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

conocido que es como un objeto. Este error podría estar ocultando un homónimo. Para evitar estos riesgos y asegurar la consistencia del LEL, se propone la incorporación de la Vista Clasificación, la cual permite mostrar información específica y selectiva de cada símbolo durante la edición del LEL y, de esta manera, alertar al ingeniero/a de requisitos de posibles usos incorrectos u omisiones. En síntesis, esta Vista permite:

- 1) Reforzar el principio de circularidad al hacer explícita parte de la información de cada símbolo. De esta manera, existe una alta probabilidad de detectar errores de uso.
- 2) Informar acerca de la presencia de una jerarquía conceptual, identificando si se trata de una generalización o una especialización. De esta manera, aumenta significativamente la precisión de las descripciones.
- 3) Ante la presencia de estados, informar la cadena de estados existente en el LEL, o sea, con qué otros estados se relaciona el símbolo en cuestión. De esta manera, se mejora el uso de los estados y se pueden detectar usos incorrectos y omisiones.

La Vista Clasificación selecciona un símbolo, lo delimita y le incorpora información. Se han analizado diferentes maneras de hacerlo, teniendo siempre en cuenta generar el menor grado de cambio posible en la sintaxis existente. Siendo que la forma de marcación de hipervínculos más utilizada en internet es el color combinado con el subrayado (“[hipervínculo de texto](#)”) y que la heurística de construcción del LEL no ha definido una forma única de hacerlo, sólo se ha sugerido utilizar el color, el subrayado o ambos. Por lo tanto, la decisión de la forma de marcación ha quedado a criterio de quien construye el LEL o si existe una herramienta, de quien la haya diseñado. En la presente tesis se ha utilizado preferentemente el subrayado. De esta manera, para el símbolo “Actualizar Ficha del Libro” la sintaxis conocida es la siguiente:

[Actualizar Ficha del Libro](#)

Para incorporar la información de la Vista Clasificación, es necesario decidir la mejor representación posible, principalmente para que no entorpezca la lectura. Se propone alguna de las siguientes sintaxis:

[Actualizar Ficha del Libro](#)_(v) o [Actualizar Ficha del Libro](#)_v

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

Si bien cualquiera de estas sintaxis es apropiada para la Vista Clasificación, fue necesario analizar también la Vista Empotramiento para unificarlas. Sólo para comprender el motivo de esta unificación, se adelanta que el empotramiento es la presencia de otros símbolos en el nombre de un símbolo (Ejemplo, “Actualizar Ficha de Libro” tiene “Ficha de Libro” y “Libro”). Esto implica que para marcar el empotramiento con el subrayado es necesario utilizar más de un nivel de subrayado para marcar todos los empotramientos existentes, dificultando la legibilidad del LEL. A partir de esto, se decidió el uso de corchetes para delimitar cada símbolo y la sintaxis utilizada es la que sigue:

El [solicitante]_s paga el trámite en la [caja]_s

Retomando el ejemplo del uso incorrecto del símbolo “caja”, se puede observar que al activar la Vista Clasificación se puede reconocer con gran facilidad la inconsistencia en la oración. La propuesta de la Vista Clasificación es acceder rápidamente a información sensible de cada símbolo, es un llamado de atención para el ingeniero/a de requisitos que le permite analizar información del LEL que, hasta este momento, sólo se podía acceder al navegar por los símbolos, con el ya mencionado esfuerzo que conlleva. La detección del uso incorrecto de caja trajo otro análisis, referido a decidir qué símbolo era el correcto y si el mismo había sido detectado previamente. En este caso, el símbolo “Mostrador” ya existía en el LEL y sólo fue necesario reemplazar un símbolo por otro:

El [solicitante]_s paga el trámite en el [mostrador]_o

La Vista Clasificación no modifica el LEL, sino que expone información existente de los símbolos y deja a consideración del ingeniero/a de requisitos efectivizar los cambios pertinentes. La Vista puede ser activada en cualquier momento, cuando el ingeniero/a de requisitos lo considere necesario. De esta manera, no se afecta la sintaxis del LEL y resulta transparente para el cliente-usuario. La misma Vista puede ser utilizada durante la edición de los Escenarios (ver **Capítulo 16**). En la presente tesis, esta herramienta no se describe, sólo se mencionan algunas funcionalidades cuando estas benefician la mejora que está siendo considerada.

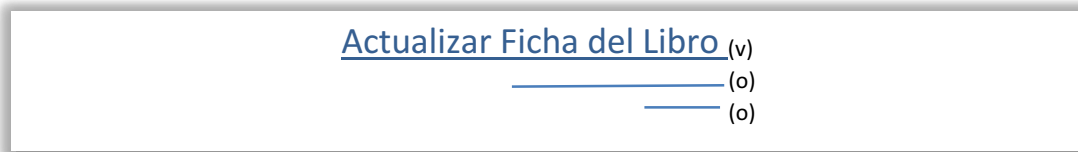
Vista Empotramiento

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

Como ya se mencionó, esta Vista es un caso especial de la Vista Clasificación y pueden ser activadas juntas o separadas. Esto se debe a que la Vista Empotramiento actúa en el nombre del símbolo mientras que la de Clasificación en la noción y el impacto. A pesar de ello, se describen por separado para explicarlas mejor y porque esta Vista no es accesible desde el entorno de edición de los Escenarios, ya que incorporar más información del LEL al Escenario dificulta su lectura.

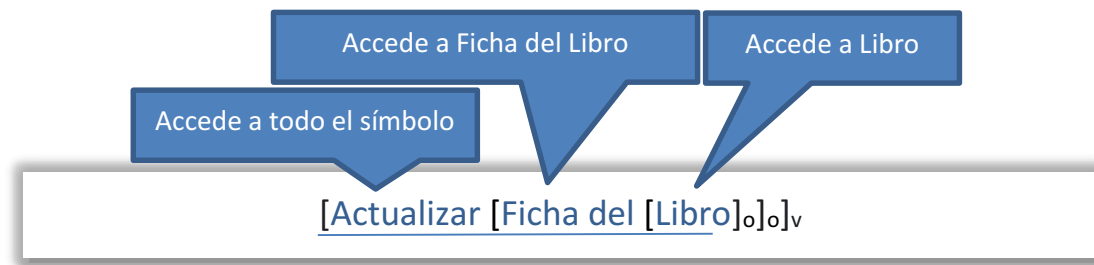
Como se mencionó en el **Capítulo 10**, el empotramiento existe en todos los léxicos y se presenta como una superposición sintáctica entre símbolos. Además, como también se describió oportunamente, mejora la visualización de los vínculos estrechos que existen en el nombre de un símbolo (por ejemplo, en “Actualizar Ficha del Libro” existe “Ficha de Libro” y “Libro”). Su mayor aporte es la posibilidad de detectar palabras o frases relevantes del macrosistema que no han sido identificadas aún como símbolos del LEL. Como se mencionó en la Vista Clasificación, la sintaxis más conocida para los hipervínculos es con subrayado. Para visibilizar el empotramiento, esta modalidad entorpece la legibilidad cuando existe más de un empotramiento en el mismo símbolo:



Esta sintaxis llevada a un documento con todo el LEL, lo hace verdaderamente complejo de leer y comprender. Esto fue lo que determinó que para la Vista Clasificación se decidiera el uso de corchetes, ya que permite representar varios empotramientos en un mismo renglón:



Cabe destacar que cada símbolo empotrado deberá tener su hipervínculo al símbolo que representa. En el caso de Actualizar Ficha del Libro:



En los casos estudiados no se han encontrado casos cuando los anidamientos no cumplen la propiedad de que un símbolo sea un subconjunto propio de otro. Por lo tanto, la propuesta de los corchetes anidados se ha considerado la mejor representación. Pero cuando este anidamiento no se cumple, los corchetes pueden presentar problemas. Por tal motivo, se propone una representación alternativa, menos legible pero irrefutable. En este caso se propone generar un conjunto de los símbolos empotrados al lado del símbolo, como sigue a continuación:

$[Actualizar Ficha del Libro]_v [Ficha del Libro]_o [Libro]_o$

La herramienta debe identificar estos casos y cambiar automáticamente su sintaxis. Los tipos de los símbolos participantes en un empotramiento pueden ser diferentes. O sea, los empotrados pueden ser del mismo tipo que el símbolo que los contiene o no, pudiendo ser cada uno de un tipo distinto. Por ejemplo, en “Lista de proveedores habituales” y “Lista de proveedores invitados” se puede observar que “habitual” como “invitados” son estados y el símbolo “Lista de proveedores” es un objeto. Este ejemplo es interesante ya que “habitual” e “invitado” no aparecían como estados en este ejemplo de LEL y después incorporar la visualización del empotramiento se hicieron claramente visibles.

Puntos de Vista

Los puntos de vista deben estar representados en el LEL y pueden ser detectados espontáneamente en el discurso del cliente-usuario o al indagar más profundamente en las tareas realizadas. Para identificar un punto de vista, se deben realizar preguntas del tipo:

“¿es la única manera de hacerlo?” “¿se está haciendo actualmente?”

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

El punto de vista del “Es” y “Deber ser” identifican las actividades o tareas que no siempre se realizan de la forma en que se las planeó. La propuesta es procurar determinar cuál es la forma de trabajo habitual y la importancia que tiene para quien solicita el Sistema de Software. Cuando se comiencen a describir los servicios que proveerá el Sistema de Software, al construir los Escenarios Futuros, se debe tener en claro como deberá tratar el sistema estas situaciones. Cuando la forma de trabajo habitual no es la prescrita por el “Deber ser”, es correcto escribir el punto de vista del “Es” y continuar con una nueva oración con punto seguido, con la palabra “Debería ...” para identificar el punto de vista del “Deber ser”. Es necesario mantenerse dentro de la misma noción o impacto para dejar claro que se trata del mismo concepto. Si el apartamiento del “Deber ser” es poco frecuente, debe describirse primero el punto de vista del “Deber ser” y tratar la visión del “Es” como formas excepcionales.

La sintaxis propuesta es la siguiente:

El [vendedor](#) tira el triplicado del [ticket](#) a la basura. **Debiera** guardarlo en el [cajón de triplicados](#) para su posterior control.

En este ejemplo, lo habitual es que los triplicados se tiren a la basura (“Es”), a pesar de que existe la obligatoriedad de guardarlos para controlarlos (“Deber ser”).

Para explicar con mayor detalle estos puntos de vista, se presenta un ejemplo. En una organización existe una regla del negocio que dice que “todas las solicitudes de materia prima deben quedar registradas, o sea deben realizarse personalmente en el mostrador del Stock para firmar la solicitud o en su defecto, por mail”. Puede suceder que el que realiza la solicitud es el jefe de producción y por cuestiones operativas, lo hace telefónicamente, ignorando la regla del negocio. También se puede suponer que todos en la organización saben y aceptan que no se cumple la regla del negocio, pero para no entorpecer los tiempos de producción permiten esta omisión. En este contexto supuesto, podría aparecer alguno de los siguientes impactos:

- 1) La materia prima es solicitada telefónicamente
- 2) La materia prima es solicitada personalmente o por mail.
- 3) La materia prima es solicitada personalmente, por mail o telefónicamente.
- 4) La materia prima debe ser solicitada personalmente o por mail.
- 5) En ocasiones, la materia prima es solicitada telefónicamente. Debe ser solicitada personalmente o por mail
- 6) En ocasiones, la materia prima es solicitada telefónicamente, pero debe ser solicitada personalmente o por mail.
- 7) En ocasiones, la materia prima debe ser solicitada personalmente o por mail, pero se hace también telefónicamente.
- 8) En ocasiones, la materia prima suele ser solicitada telefónicamente.
- 9) La materia prima se solicita personalmente, por mail y telefónicamente. Esta última se realiza cuando hay una necesidad operativa.

Figura 85 – Ejemplo de Impactos con Puntos de Vista

En la Figura 85 se puede observar que cualquiera de estos impactos son factibles de encontrar en un símbolo, pero cada uno de ellos tiene una interpretación diferente:

- En el 1 y 2 no hay manera de comprender que existe otro punto de vista.
- En el 3 se asume a ambos puntos de vista son iguales. Es el más peligroso.
- En el 4 se deja ver que existe otro punto de vista.
- En 5, 6 y 7 queda claro que hay dos puntos de vista.
- En el 8 se deja ver que hay otras formas de hacerlo, sin saber si se trata de otros puntos de vista o sólo de otras opciones.
- En el 9 se puede observar otra opción para expresar una realidad diferente.

La manera correcta de representar dos puntos de vista es que ambos estén claramente descriptos, como es el caso de los ítems 5, 6 y 7 de la Figura 85.

Puede observarse en lo antes descrito, que existe una gran posibilidad de encontrar información en los impactos que no fue visibilizada. También, puede ocurrir que la información no esté completa. Por tal motivo, se sugiere que el ingeniero/a de requisitos formule preguntas que permitan profundizar acerca de otras opciones o puntos de vista existente. Las preguntas sirven para identificar puntos de vista:

¿Existe alguna normativa? ¿Existen otras formas de hacerlo?

¿Todas las formas de hacerlo son de igual importancia?

¿Desde niveles superiores, se espera que se realice de alguna forma particular?

¿Sin importar lo establecido, cómo lo suelen hacer?

En resumen, el ingeniero/a de requisito debe estar atento a las siguientes opciones:

- 1) Que exista un “Deber ser” hace sospechar que pueda haber un “Es” que lo contradiga.
 - a) Esto debe ser verificado.
- 2) Si coexisten un “Deber ser” y un “Es” contradictorios, entonces, puede pasar que:
 - a) Existen las dos conductas.
 - b) Existe sólo el “Es”.
 - c) La obligatoriedad del “Deber ser” no es absoluta.
 - d) Esta duda debe ser resuelta antes de dar por cerrado el LEL.
- 3) Sería ideal que todo “Deber ser” sea complementado con las intenciones acerca del futuro, por lo tanto:
 - a) Hay que prestar especial atención a si aparece en el UdeD algún “Próximo” relacionado.

Es necesario recalcar que todo “Deber ser” debe ser registrado en el LEL, aunque sea obsoleto o se desea erradicarlo, porque contribuye a la comprensión del término o los términos relacionados.

El punto de vista del “**Próximo**” suele aparecer como una observación referida a un aspecto del contexto que se conoce, pero que aún no ha sido materializado. Estas observaciones afectarán a los EA y luego, serán de utilidad al definir los servicios del nuevo Sistema de Software. El ingeniero/a de requisitos debe estar alerta a observaciones que aparecen al describir un símbolo y que tienen carácter aclaratorio. Por ejemplo, al preguntar por los pizarrones la respuesta es “Bueno, tenemos pizarrones, pero ya se compraron pizarras electrónicas”. Aparecen por lo general, de forma totalmente espontáneas. La sintaxis propuesta es incorporar la oración en el mismo renglón anteponiendo la palabra “Será” o semejante:

- El profesor dibuja una tabla con dos columnas para el grupo y los temas
- Escribe los temas para que los alumnos seleccionen
- Un alumno de cada grupo pasa al frente y escribe su numero de grupo a la derecha del tema elegido
- El profesor copia en su cuaderno de clase los grupos los temas asignados. Los pizarrones **serán** reemplazados por pizarras electrónicas que permiten guardar una imagen del pizarrón.

Sinónimos

Dos dudas frecuentes se presentan ante los sinónimos del LEL:

- 1) Cómo se debe representar el componente Nombre
- 2)Cuál de los sinónimos se debe utilizar al mencionarlo en otro símbolo, en un Escenario o en la ERS.

Con respecto a la primera duda, se puede observar en el meta modelo del LEL (ver Figura 34) que no se menciona su representación, sin embargo se ha utilizado una barra (“/”) para separar los diferentes nombres de un mismo símbolo. Con respecto a la segunda duda, y atendiendo a visibilizar la complejidad del léxico, es preferible utilizar la entrada completa del símbolo, o sea el nombre completo del símbolo con todos sus sinónimos. Se debe recordar que cualquier decisión que se tome debe ser aplicada tanto en el LEL como en los Escenarios. Puede observarse en la Figura 86 que si bien la descripción del “uso no sugerido” es correcta, puede generar gran dificultad en la lectura, principalmente si existen muchos sinónimos. Se sugiere utilizar solo uno de los nombres del símbolo para asegurar la legibilidad del modelo.

USO NO SUGERIDO

Insumo(1)
Noción

- ...
- Puede tener el estado amarillo/en cuarentena/pendiente, verde/aprobado o rojo/rechazado

USO SUGERIDO

INSUMO (1)
Noción

- ...
- Puede tener el estado amarillo. verde o roio

Figura 86 – Ejemplo del uso sugerido de sinónimos (Caso SCC)

Utilizar expresiones más breves significa elegir uno de los sinónimos en representación de todos. En general, el primer nombre que aparece suele ser el más mencionado en el contexto. Si bien utilizar este nombre es correcto, utilizarlo en todos los casos puede ocultar la complejidad léxica del símbolo. Siendo que el objetivo del LEL es visibilizar la complejidad y los conflictos léxicos, se recomienda, siempre que sea posible, utilizar indistintamente todos los sinónimos, de a uno por vez. Esta manera de trabajar puede tener un esfuerzo adicional de tener que acceder a la definición por no reconocer un sinónimo, pero no interfiere en la lectura y asegura una buena percepción de la complejidad léxica.

Homónimos

Como ya se mencionó en el **Capítulo 10**, los homónimos deben ser identificados en el LEL porque son parte constitutiva del léxico. Cabe recordar que la homonimia se puede presentar en cualquiera de los tipos del LEL (sujeto, objeto, verbo y estado) y que no existe una limitación cuantitativa. Sólo se debe tener presente que deben incluir en el LEL sólo aquellos que son usados en el contexto en estudio. Para identificarlos, cada homónimo se debe representar por separado y tener una enumeración secuencial que permita relacionarlos:

{SímboloHomónimo(1), SímboloHomónimo(2), ..., SímboloHomónimo(n)}

o

{SímboloHomónimo_1, SímboloHomónimo_2, ..., SímboloHomónimo_n}

Siempre que sea posible, se sugiere incorporar en la noción de cada símbolo una indicación específica que diferencie cada homónimo. Por ejemplo, en un símbolo incorporar “Para Producción ...” y en el homónimo “Para Calidad ...”.

Para ejemplificar la importancia de la identificación de los homónimos, en la **Figura 87** se puede observar la dificultad que se presenta cuando no se identifican los homónimos del contexto, tal es el símbolo de la primera columna donde se incorporó toda la información correspondiente a ambos símbolos. Para visualizar la mejora en la legibilidad al utilizar los homónimos, se describen en la segunda columna, los dos símbolos involucrados.

Sin Homónimos	Con Homónimos
<p>Insumo</p> <p>Noción</p> <ul style="list-style-type: none"> • Es todo aquello que ingresa al laboratorio : las materia prima, los granel, los producto semielaborado (1), los producto terminado, el material de acondicionamiento , los envases y los productos importados • Tiene un número de recepción • Tiene un número de análisis <p>Impacto</p> <ul style="list-style-type: none"> • Se le toma una muestra • Puede tener etiqueta amarilla, verde o roja • Se le realiza el control de calidad • Puede ser enviado al depósito de cuarentena, al depósito de aprobados, al depósito de rechazados, al depósito de devoluciones según corresponda • Si es producto o producto importado, se le puede tomar una contramuestra 	<p>Insumo(1)</p> <p>Noción</p> <ul style="list-style-type: none"> • Definición del Sector Control de Calidad • Es todo aquello que ingresa a la empresa : las materia prima, los granel, los producto semielaborado (1), los producto terminado, el material de acondicionamiento , los envases y los productos importados • Tiene un número de recepción • Puede tener el estado amarillo, verde o rojo <p>Impacto</p> <ul style="list-style-type: none"> • Se le toma una muestra • Se le realiza el control de calidad • Se lo envía a elaborar • Puede ser enviado al depósito de cuarentena, al depósito de aprobados, al depósito de rechazados, al depósito de devoluciones según corresponda • Si es producto o producto importado, se le puede tomar una contramuestra
	<p>Insumo(2)</p> <p>Noción</p> <ul style="list-style-type: none"> • Definición del Sector Producción • Son todos los elementos requeridos para producir: las materia prima, el material de acondicionamiento y los envases • Tiene un número de recepción • Tiene un número de análisis • Debe estar aprobado <p>Impacto</p> <ul style="list-style-type: none"> • Se utiliza para producir

Figura 87 – Ejemplos de uso de homónimos (Caso SCC)

Tratamientos de estados

Como se mencionó en el **Capítulo 10**, puede ser dificultoso percibir los estados, haciendo que el ingeniero/a de requisitos deba esforzarse para detectar la presencia de algún indicio que permita identificarlos. Esta búsqueda se debe realizar prestando una especial atención a las expresiones utilizadas por los clientes-usuarios, en las transcripciones de las entrevistas, en la literatura del macrosistema, etc. En ocasiones, los estados están explícitos, en otras, deben ser interpretados y confirmados.

A continuación, se analizan algunos problemas relacionados con los estados y como se debe actuar en cada caso.

- Determinar si se está en presencia de un estado.

“El encargado de planta de cada turno envía al final de su horario de trabajo todas las órdenes de producción terminadas, así como un listado de órdenes de producción parcialmente completadas durante su turno. Es decir, las órdenes incompletas quedan en poder del encargado de planta quién se las entrega al encargado de planta del turno siguiente.”

(Texto del Caso de Cajas de Cartón Corrugado)

En el texto aparece “órdenes de producción terminadas”, “órdenes de producción parcialmente completadas” y “órdenes incompletas”. Lo primero que se debe analizar es si se trata de tres símbolos objetos o es un símbolo objeto (“ordenes de producción”) con sus estados (“Urgente”, “Terminada” o “Parcialmente Completadas/Incompletas”). Justamente esta decisión acerca de si es un único símbolo o diferentes, es una de las dudas más recurrentes cuando se habla de estados en el LEL. En este caso, las “órdenes de producción” se generan antes de que sean ejecutadas, pudiendo resultar en que se terminen o no, en este caso quedan parcialmente completas.

En este caso se trata del símbolo Órdenes de Producción que tiene al menos tres estados Terminadas, Completadas e Incompletas y un tercero menos perceptible que aparenta ser anónimo. También en este caso, es importante destacar que el símbolo *Orden de Producción Urgente* es una especialización de *Orden de Producción*.

Para distinguir entre las especializaciones y los estados hay que responder la pregunta si hay cambios de estados involucrados o no. En este caso las Órdenes de Producción *Incompletas* se convertirán en algún momento en *Terminadas*, sin embargo, las *Órdenes de Producción Urgentes*, nunca fueron *Órdenes de Producción* normales, son generadas normales o urgentes dependiendo fundamentalmente del tiempo de entrega.

- Diferenciar los verbos de los estados.

*“Tan pronto como la orden de compra es **aprobada** en la oficina comercial de Norpack, una copia de la misma es remitida a la oficina de planificación de la producción.”*

(Texto del Caso de Cajas de Cartón Corrugado)

Puede observarse que en este texto existe una ambigüedad ya que no se comprende si *aprobada* se refiere a la acción de aprobar o un estado que aplica a la orden de compra. En este caso, la duda no puede ser resuelta en el mismo texto, obligando a retornar al UdeD en busca de más información.

- Estados que aplican a varios símbolos base.

Otro problema que puede aparecer es que un estado se repita para diferentes símbolos base. Por ejemplo, el estado “Terminada” aparece dos veces en el texto, primero aplicado a “Orden de Producción”, como se describe al inicio de esta sección, y luego, a “Cajas”, como se puede observar en el siguiente texto:

*“Las cajas **terminadas** pueden ser retiradas de 9 a 13 o de 14 a 17 en las instalaciones de la fábrica, pueden ser entregadas en el domicilio del cliente o remitirse por un transporte a otra ciudad.”*

(Texto del Caso de Cajas de Cartón Corrugado)

En este caso, los dos estados “Terminada” deben ser tratados como **homónimos**. O sea, deben ser descriptos como dos símbolos por separado haciendo referencia en la noción a qué símbolo base aplica cada uno de ellos (Terminada(1) y Terminada(2)).

- Detección de nuevos estados asociados

Debe tenerse siempre presente que por su propia naturaleza los estados no pueden existir aislados. Es así que, al enfrentarse a un estado aparentemente aislado o singular se debe desencadenar una revisión del contexto, para determinar cuáles son los otros estados que ratifican la condición de estado presupuesta. Para ello, se debe analizar tanto el símbolo base como los

símbolos relacionados, prestando especial atención a los hipervínculos, en busca de información que ayuden a detectar otros estados.

- Estados dominantes

Como se mencionó en el **Capítulo 10**, puede existir un estado que se acople naturalmente al símbolo base. O sea, un estado que es el más característico del símbolo base y en el cual se lo puede observar por defecto en el contexto. En general, no es inmediato que se detecte ya que se toma dicho estado como natural. Por ejemplo, un cliente que se da de alta es activo por defecto. Esta percepción se modifica en cuanto el estado es modificado a moroso o de baja. Existe el riesgo que cuando se describe al cliente, seguramente se registre información que no es propia, sino referida a su estado activo. En este caso, asumiendo que el estado existe como tal en el macrosistema, se lo debe describir como símbolo. Luego, se debe determinar cuál es la información que efectivamente pertenece al símbolo base y cual al estado.

- Diferenciar los estados del contexto de los estados de Sistema de Software

Los estados del contexto son aquellos que utilizan los clientes-usuarios en sus discursos. Los estados del software son aquellos necesarios para mantener la consistencia de la información. Existirán estados del contexto que se mantendrán activos en el UdeD futuro y se transformarán en estados del software, pero los estados del software no necesariamente serán los del contexto, ya que algunos de ellos sólo afectarán las funciones o procesos del Sistema de Software. Durante la Ingeniería de Requisitos los estados del contexto son observables durante la Comprensión del UdeD actual, mientras que los estados del software aparecen al describir los Escenarios Futuros.

- Estados de los verbos

Los estados de los verbos pueden confundirse fácilmente con jerarquías o viceversa. Cuando se está en presencia de una jerarquía de un verbo, la misma describe *diferentes formas de hacer algo*, mientras que los estados están relacionados con pasos intermedios en la realización de la actividad indicada por el verbo. Una potencial complicación adicional surge del hecho que es

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

posible que un componente de una jerarquía de un verbo tenga a su vez estados.

Afortunadamente no se han observado ocurrencias de esta combinación.

La sintaxis para describir un estado se encuentra en el patrón sintáctico correspondiente, donde se sugiere que en la noción se deje claro **qué** significa. En este punto es donde se debe incorporar la relación con el símbolo base. En el ejemplos de la Figura 88 se puede observar el símbolo estado Terminada y la sintaxis utilizada para indicar el símbolo base asociado.

URGENTE

Noción

- Son [órdenes de producción](#) atrasadas con un plazo de entrega menor a 10 días.
- Son [órdenes de fabricación](#) recibida de Papelera del Sudeste

Impacto

- Son incluidas en los [programas de fabricación](#) más próximos posibles.

Figura 88 – Ejemplos de definición de Estado (Caso SNor)

Para hacer más explícita la relación entre el símbolo base y sus estados, se propone incorporar en la noción del símbolo base una oración que indique explícitamente los estados que puede tomar (ver Figura 89), utilizando una frase del tipo:

“Sus estados son ...”, “Puede estar en el estado ...” o una expresión semejante.

ORDEN DE PRODUCCIÓN / ORDEN DE FABRICACIÓN / OP

Noción

- Es un documento digital que contiene el [despliegue](#) de cada [caja](#) o [accesorios](#) a fabricar
- Contiene un solo [tipo](#) de [caja](#) o [accesorio](#)
- Tiene una [fecha de entrega](#)
- Sus estados pueden ser [urgente, terminada y parcialmente completada](#)

Impacto

- Son utilizadas para [planificar la producción](#)
- Puede ser reprogramadas en la [Oficina de Planificación de la Producción](#)
- Pueden ser incluidas en el [Informe de Dificultades de Fabricación](#)
- Son [priorizadas](#) según el menor [desperdicio de materias primas](#).
- Cuando no puede ser resuelta en tiempo y forma puede ser entregada a la [Papelera del Sudeste](#) para su fabricación.

Figura 89 – Ejemplo de un símbolo base y sus Estados (Caso SNor)

Esta noción permite visibilizar fácil y rápidamente la relación entre un símbolo base y sus estados. Además, no presenta complicaciones cuando el LEL se debe actualizar.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

A continuación, se presentan un conjunto de observaciones acerca de la existencia de los estados en el contexto:

- 1) El estado debe aparecer como tal en el UdeD. O sea, debe ser usado como estado por los involucrados.
- 2) Suelen estar acompañados por verbos que muestran transición (“Se envía al final del día”, “queda en poder del encargado”, “se entrega en el domicilio del cliente”) o lugar (“es aprobada en la oficina”).
- 3) Puede ser algo o alguien que es evaluado o considerado de una determinada forma en un momento específico (“la maestra está enojada”)
- 4) Puede ser una caracterización de un elemento existente en el contexto, de un valor relevante del contexto, de una condición que influye sobre un comportamiento (“en el sorteo se incluyen todos los solicitantes menos los morosos”).
- 5) Cuando un estado aparece en el contexto junto al símbolo base, analizar si se trata de un único símbolo o si se está en presencia de dos símbolos donde uno de ellos es un estado (ejemplo, en un contexto puede ser “factura anulada” como objeto y en otro “factura” como objeto y “anulada” como estado).

Algunas recomendaciones para describir la relación entre el símbolo base y el estado:

- 1) El nombre de un estado **no** debe incluir el nombre del símbolo base (“anulada”, “vencida”, “pendiente”).
- 2) En cada estado, describir en la noción a qué símbolo base aplica con una oración del tipo “Es cuando el *símbolo base* está/se encuentra/etc ...”.
- 3) En el símbolo base, incorporar una noción que diga “Los estados son ...” o “Puede estar ...”.
- 4) Descubrir cuál es el estado dominante y analizar el símbolo base para ver si es necesario incorporar algún otro impacto relacionado con los otros estados o si se han incluido impactos que realmente corresponden al estado dominante.
- 5) Analizar la lista de estados descriptos en la noción del símbolo base. Pueden aparecer elementos no identificados como símbolos, asegurarse que realmente no lo sean o incorporarlos.
- 6) Analizar los estados ya descriptos, para detectar estados omitidos. Pueden aparecer en la noción al describir que sucedió antes o en los impactos al

determinar que acciones u otros estados desencadenan.

- 7) Cuando existe el mismo estado para distintos símbolos base, registrarlos como homónimos.

Se ha podido observar que estas recomendaciones permiten mejorar la heurística de descripción del LEL. Además, permiten detectar nuevos estados, siendo esto una mejora muy significativa para la completitud del LEL. Finalmente, se muestra en la Figura 95 como se puede aprovechar la información del LEL durante la descripción de un símbolo activando la Vista Clasificación. El símbolo “Orden de Producción” se relaciona con otros símbolos objetos, con estados y con jerarquías conceptuales como Caja y Accesorios. Esto permite que al utilizar el símbolo base o uno de los estados, el ingeniero/a de requisitos sepa que existen otros estados que se deben tomar en cuenta, o de lo contrario, que lo alerten que se han omitido otros estados.

Jerarquías Conceptuales

Las jerarquías conceptuales del contexto deben ser percibidas por el lector del LEL de una manera simple y natural. Esto es de fundamental importancia para una mejor comprensión del vocabulario utilizado en el macrosistema. Como ya se mencionó en el **Capítulo 10**, las jerarquías conceptuales existen en el macrosistema, pero no siempre son fáciles de detectar, principalmente en una entrevista. El ingeniero/a de requisitos debe estar atento a la aparición de sinónimos parciales, de indicaciones de la existencia de genéricos esenciales como “puede ser”, “en ocasiones...”, “cuando se...” o de componentes en una relación mereológica del tipo “es parte de”, “está formado por...”, “contiene a...” o equivalente. Si bien las jerarquías del LEL tienen cierta similitud con las jerarquías de la Programación Orientada a objetos (POO), existen diferencias importantes. Por un lado, en el LEL puede haber jerarquías en los verbos y eventualmente de los estados (ver Figura 90).

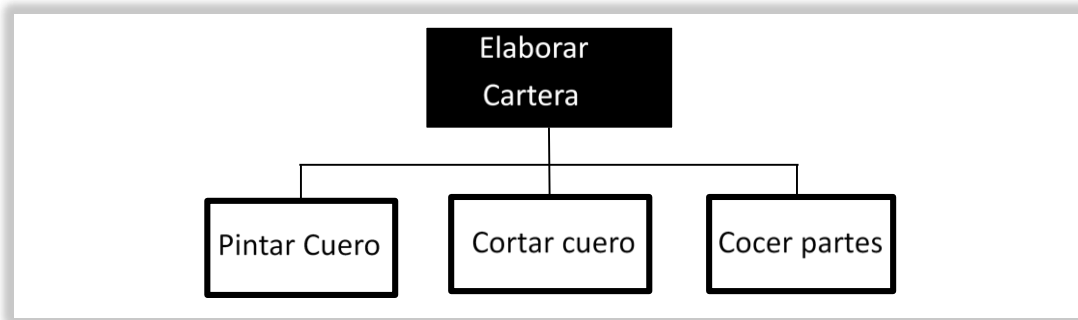


Figura 90 – Ejemplo de una Jerarquía Conceptual de Verbos (mereología)

Por el otro lado, las jerarquías de los objetos y los sujetos del LEL, que sí son cercanas a la POO, se diferencian sustancialmente en que esas jerarquías corresponden al UdeD actual, donde aún no existe el sistema y no se sabe si esas jerarquías permanecerán y se trasladarán a éste. Puede observarse en la Figura 91 que el término “Productos Elaborados” es un genérico y sus especializaciones son “Cajas” y “Accesorios”. A su vez, el símbolo “Cajas” es genérico de otra jerarquía, cuyas especializaciones son “Cajas con Solapa” y “Cajas Tapa y Fondo” como sucede con “Accesorios” cuyas especializaciones son “Separador” y “Divisor”.

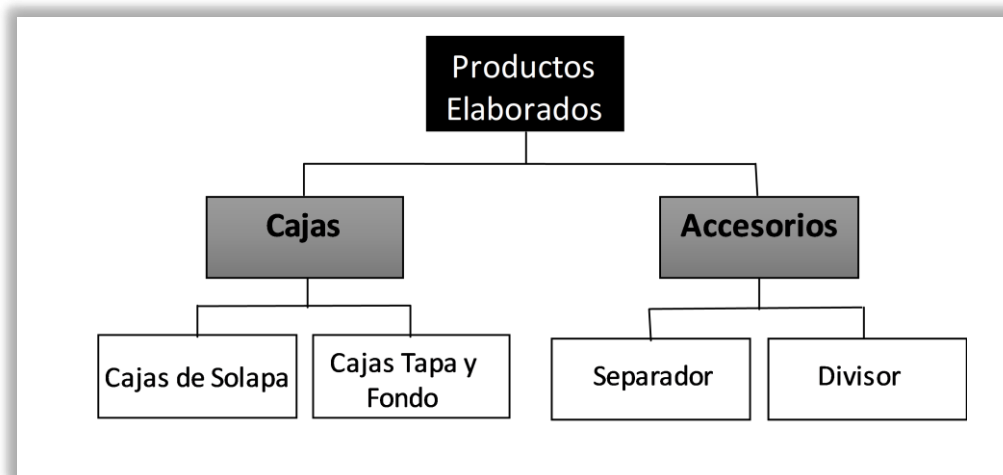


Figura 91 – Ejemplo de una Jerarquía Conceptual de Objetos (taxonomía) (Caso SNor)

Para permitirle al lector visibilizar las jerarquías en el LEL con el menor esfuerzo de comprensión posible, es necesario mantener una sintaxis homogénea en todo el glosario. La sintaxis propuesta es la siguiente.

- En la noción del símbolo genérico o composición enumerar todas sus especializaciones o componentes.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

- En cada especialización incluir una noción que indique la relación con su genérico o combinado:
 - a. Si es una taxonomía escribir una oración del tipo “Es un ...” o equivalente.
 - b. Si es una mereología escribir una oración del tipo “Es parte de ...” o equivalente.

Esta sintaxis se puede observar en la Figura 92 donde se ha representado en el LEL la jerarquía conceptual de la Figura 91. Puede observarse en los símbolos “Caja con Solapa” y “Caja Tapa y Fondo” se ha utilizado la expresión “Es un” para indicar que son especializaciones de “Caja”. Como ya se mencionó, en el caso de los sinónimos “Caja” y “Caja de cartón corrugado” puede ser utilizados indistintamente.

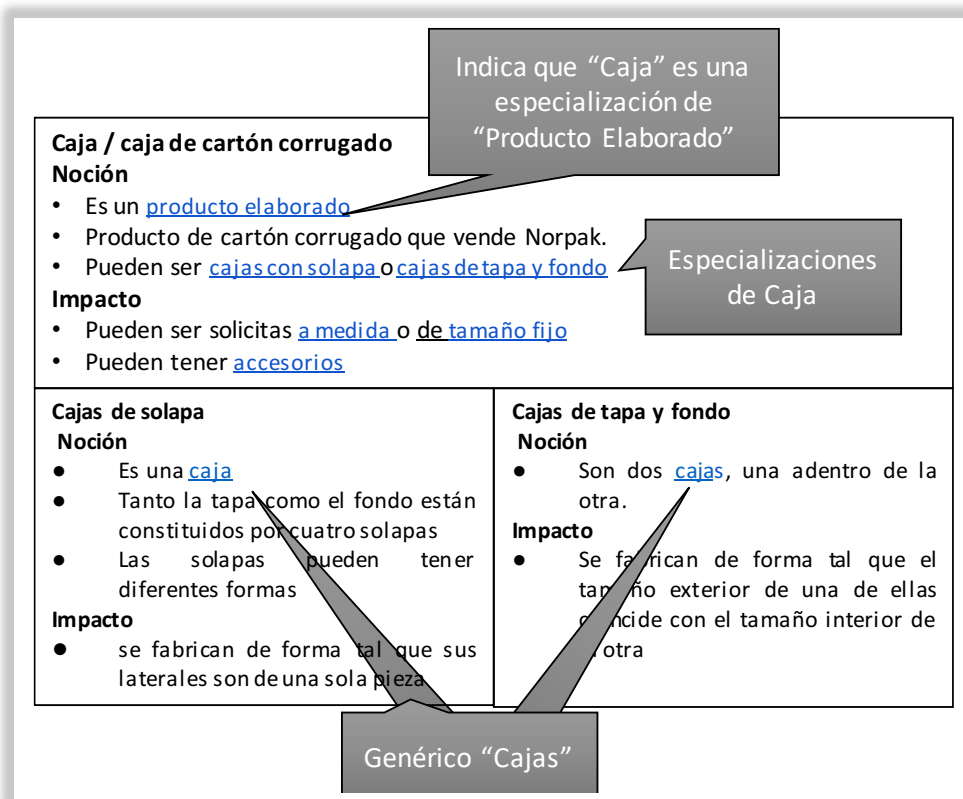


Figura 92 – Ejemplo de uso de Jerarquías en el LEL (Caso SNor)

Una jerarquía puede ser leída de diferentes maneras. La Figura 93 muestra todas las posibilidades. La lectura del genérico en forma aislada (ver Figura 93 -1) permite una mirada abstracta de la jerarquía. De esta manera no se presta atención a los detalles y sólo se atienden los aspectos comunes. La lectura parcial de la jerarquía (ver Figura 93-2), que comprende el genérico o agregado y una especialización o componente, permite entender

claramente esa parte de la jerarquía, dejando oculto el resto. La lectura aislada de una especialización o un componente permite una comprensión parcializada (ver Figura 93-3) ya que se omiten las características comunes que se encuentran en el genérico o el rol del componente en el agregado, pudiéndose interpretar erróneamente dicha información. La Figura 93-4, es la lectura más completa, donde se comprende toda la jerarquía.

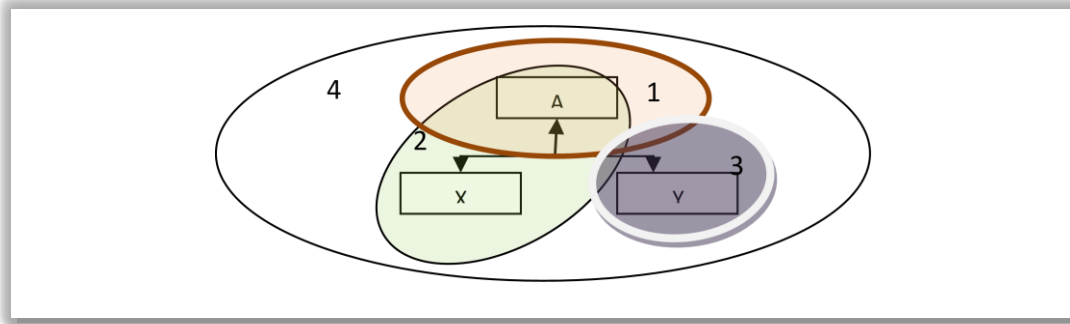


Figura 93 - Posibles lecturas de una Jerarquía Conceptual en el LEL

A continuación, se describe cómo detectar diferentes componentes de una jerarquía:

- Estar alerta a las expresiones que cambian el nivel de detalle en el discurso del cliente-usuario, o sea por momentos describe con mucho detalle y en otros generaliza.
- Estar alerta a las relaciones semánticas en el discurso del cliente-usuario entre diferentes palabras o frases, sean o no símbolos del LEL. Tener en cuenta que en el caso de las jerarquías se está en presencia de una relación muy estrecha de un subgrupo de términos del léxico.
- Estar alerta a sinónimos parciales. Por ejemplo, en determinados momentos se habla de un Gerente y luego se especifica que hace el Gerente de Sistemas.

Se debe tener en cuenta que los usuarios utilizan algunas expresiones que a simple vista son iguales pero que en realidad no lo son, pero ellos no desean marcar estas diferencias. Por ejemplo, todas las áreas generan un presupuesto. Por lo tanto, existirá un símbolo “Generar Presupuesto” para cada una. En este caso, se deberán utilizar tantos homónimos como áreas existan. Puede suceder que aun conociendo estas diferencias el cliente-usuario sólo mencione que se realizan los cálculos pertinentes en cada caso y que se entrega un informe a la gerencia. En este caso, a partir de lo que dicen los clientes-usuarios no se pueden definir las especializaciones y sólo aparece un único símbolo. Obviamente esto no es aceptable para el LEL, pero sucede que el ingeniero/a de requisitos no lo puede descubrir. Cabe destacar que cuando se construyan los Escenarios, estas diferencias se

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

harán visibles, por lo que será necesario modificar el LEL para registrar los homónimos correspondientes y desambiguar el símbolo creado oportunamente.

Como última observación, una vez finalizado el LEL, es necesario recorrerlo nuevamente buscando jerarquías que no fueron identificadas oportunamente.

Para facilitar el uso correcto de todos los elementos, se propone utilizar la Vista Clasificación que permite visibilizar la pertenencia a una jerarquía conceptual del contexto. Para explicar su sintaxis se irá describiendo paso a paso.

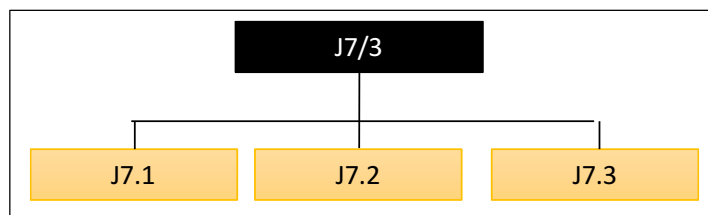
Como puede observarse en el ejemplo, en primer lugar, va el “tipo” del símbolo (S, O, V o E), luego, para determinar la presencia de una jerarquía se incluye la letra “J”. Cuando la letra “J” es superíndice indica que es un genérico o agregado y cuando es subíndice indica una especialización o componente.

$[\text{símbolo genérico}]_{\text{tipo}}^{\text{Jnro/cant}}$ y $[\text{símbolo especializado}]_{\text{tipo}}^{\text{Jnro.ele1.... ele}n[\text{cant}]}$

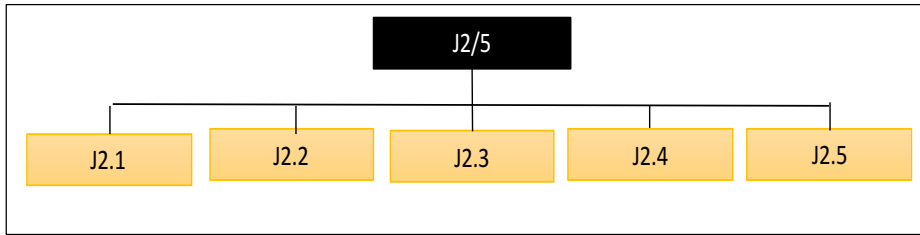
Puede observarse que a la letra “J” la sigue un “nro”, el cual agrupa a todos los elementos de una jerarquía, el menor valor que puede adquirir es 1 y el mayor es la cantidad total de jerarquías existentes en el contexto. Por lo tanto, si en un LEL existen 5 jerarquías, existirá J1, J2, J3, J4 y J5.

En los ejemplos que siguen, puede observarse que, en el caso de los genéricos, aparece la “J”, luego el “nro” que agrupa toda la jerarquía y le sigue una barra y con la cantidad total de especializaciones o componentes del nivel inmediatamente siguiente. Por ejemplo:

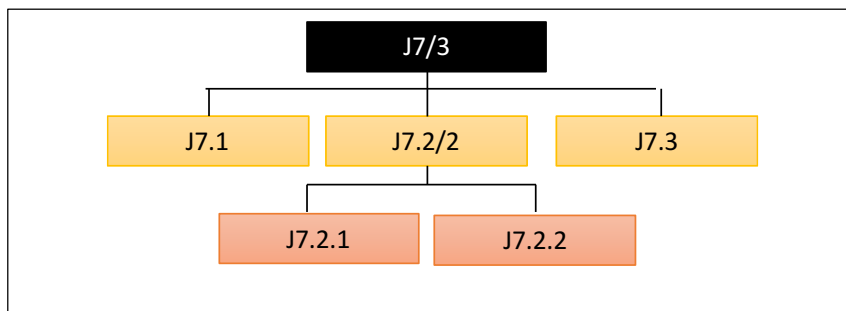
- 1) “J7/3” indica que es la jerarquía 7 con 3 especializaciones o componentes.



- 2) “J2/5” indica que es la jerarquía 2 con 5 especializaciones o componentes.



Esta nomenclatura permite aumentar el número de niveles de la jerarquía sin límites. Cabe aclarar que, en todos los ejemplos analizados, las jerarquías conceptuales se presentan en dos o tres niveles, pero no es un valor absoluto. En el caso de existir más de dos niveles, la representación es la siguiente:



En este caso se puede observar que el elemento “J7.2” se convirtió en el “J7.2/2” indicando con “/2” que existe otro nivel que depende de él con dos especializaciones o componentes. Por lo tanto, la barra indica la presencia de otro nivel y aquellos elementos que no llevan barra se pueden reconocer como hojas o terminales (“J7.1”, “J7.3”, “J7.2.1” y “J7.2.2”).

En el caso de la Figura 91 la representación con la Vista Clasificación es como se muestra en la Figura 94 y Figura 95.

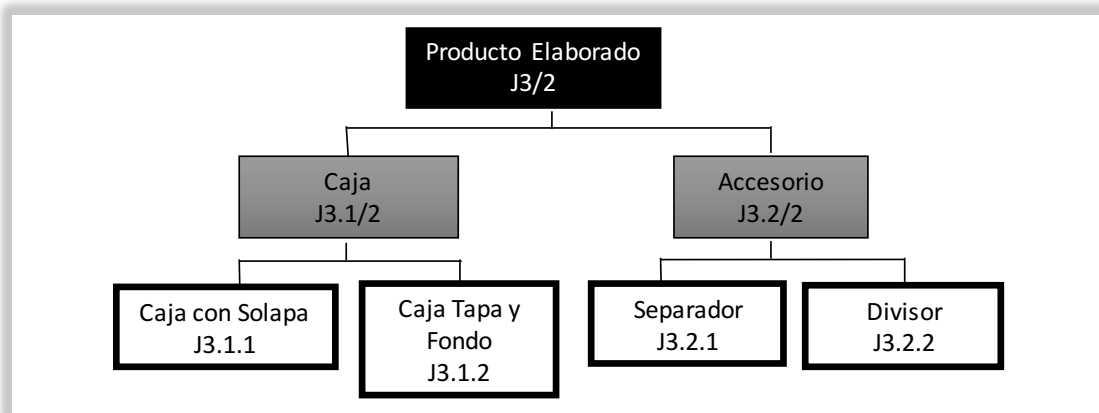


Figura 94- Identificación de la Jerarquía Conceptual (Caso SNor)

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

Cabe aclarar que toda esta información que utiliza la Vista Clasificación, a partir de la presente tesis, está disponible analizando el LEL, por lo que se trata de sólo una cuestión de visualización.

Orden de Producción

Noción

- Es un documento que describe los **[productos elaborados]_o^{J3/2}** que se deben fabricar
- Tiene una **[fecha de entrega]_o**
- Sus estados pueden ser **[urgente]_{E4}**, **[pendiente]_{E4}**, **[incompleta]_{E4}** o **[terminada]_{E4}**

Impacto

- Describe el **[despliegue]_o** de la **[caja]_o^{J3.1/2}** y los **[accesorios]_o^{J3.2/2}**
- Son elaboradas por la **[oficina de planificación de la producción]_s**
- Puede ser reprogramada **[oficina de planificación de la producción]_s**

Figura 95 – Ejemplo de la Vista Clasificación para Jerarquías Conceptuales (Caso SNor)

De esta manera se incorpora un mecanismo muy simple para identificar la presencia de jerarquías de dos niveles. Su importancia radica en comprender rápidamente, la estructura de cada jerarquía y mejorar la precisión de las descripciones y detectar alguna omisión relacionada con otra parte de la jerarquía

Evolución de los Patrones Sintácticos

Todas los cambios y agregados que se han presentado en este Capítulo, para que sean efectivos, han sido incorporadas a los patrones sintácticos. De esta forma se ayuda al ingeniero/a de requisitos a recordar todos los aspectos que son importantes para una mayor calidad del LEL. A tal efecto, en la Tabla 26, se presentan los patrones sintácticos actualizados. Las modificaciones incorporadas se han marcado con el formato de letra “negrita” para hacerlas más visibles. Un cambio para resaltar es la incorporación de la expresión “Actividad/Tarea”¹⁵ que permite representar todas las acciones que se realizan en el contexto. Este cambio se sustenta en mantener la mirada declarativa por sobre la procedural, ya que, aumentar el nivel de abstracción disminuye la dispersión de los procesos haciendo más comprensible el glosario.

¹⁵ RAE define **Actividad** como “Conjunto de operaciones o tareas propias de una persona o entidad” y **Tarea** como el “trabajo que debe hacerse en un tiempo limitado”

TIPO	NOCIÓN	IMPACTO
SUJETO	Definir claro quién es el sujeto. Identificar si es parte de una jerarquía. Identificar si tiene estados.	Registrar las actividades/tareas que realiza. Registrar qué actividades realiza sobre el sujeto. Identificar “es/deber ser” o “será”. Identificar cambios de estados
OBJETO	Definir el objeto y sus particularidades. Identificar otros objetos con el cual tiene relación. Identificar si es parte de una jerarquía. Identificar si tiene estados.	Describir las actividades/tareas que se realizan sobre el objeto. Registrar en qué actividad es utilizado el objeto. Identificar “es/deber ser” o “será”. Identificar cambios de estados.
VERBO	Describir concisamente la actividad que representa identificando quién lo ejecuta, cuándo (cronológicamente o relativo a otras actividades) y dónde se realiza. Identificar si es parte de una jerarquía. Identificar si tiene estados.	Actividades/tareas involucradas. Identificar las situaciones que impiden que la misma se realice. Qué otra acción desencadena en el ambiente y qué situaciones son causadas por la actividad. Indicar los cambios de estado que provoca la actividad. Identificar “es/deber ser” o “será”.
ESTADO	Dejar claro qué significa y a qué aplica. Qué actividades, estados y acciones llevaron a este estado. Identificar si compone una jerarquía. Identificar si tiene estados.	Identificar otros estados y actividades/tareas que pueden ocurrir a partir de este estado. Identificar “es/deber ser” o “será”.

Tabla 26 – Patrones sintácticos del LEL actualizados

Algunos comentarios finales que se incorporan a todo lo expresado en el **Capítulo 7**:

- En general, los impactos de los sujetos contienen un símbolo verbo que también es símbolo del LEL. Esto se debe a que las actividades que realizan los sujetos son de importancia en el contexto y, por lo tanto, es alta la probabilidad que sean verbos del LEL.
- En correspondencia al punto anterior, en la noción de los verbos deben estar mencionados uno o varios sujetos del LEL.
- Se espera que los impactos que no se relacionan con un símbolo verbo del LEL, sean en su mayoría, acciones atómicas, o sea, verbos de una sola acción (ejemplo, firmar). Estas acciones sólo deben ser impactos si son necesarias para comprender el comportamiento del sujeto.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

- Tanto en la noción como en el impacto, se debe prestar atención a las oraciones donde no aparece ningún símbolo del LEL relacionado. Si bien es posible, se debe evaluar la posibilidad de omisión de información, que el símbolo no pertenezca al contexto o que no se haya respetado el principio de circularidad.
- Para mostrar como aplican algunos cambios, se presentan a continuación algunos ejemplos del Caso SNorp realizado por alumnos/as de grado en 2006 con los patrones sintácticos existentes en ese momento. En 2019 se repitió la experiencia, utilizando los patrones sintácticos que se describen en la presente tesis. A continuación, se describe un ejemplo de cada tipo. Para facilitar la lectura, se han incorporado en cada ejemplo, los patrones sintácticos correspondientes a cada momento. Con el objetivo de hacer visible las diferencias, se han incorporado los ejemplos textuales.

OBJETO	
2006	2019
<p>Noción: Definir el objeto e identificar otros. objetos con el cual tiene relación.</p> <p>Impacto: Describir las actividades / acciones que pueden ser aplicadas al objeto.</p>	<p>Noción: Definir el objeto y sus particularidades dentro del macrosistema. Identificar otros objetos con el cual tiene relación. Identificar si es parte de una jerarquía. Identificar si tiene estados. Identificar aspecto temporal.</p> <p>Impacto: Describir las actividades/tareas que se realizan sobre el objeto. Registrar en qué actividad es utilizado el objeto. Identificar “es/deber ser” o “será”. Identificar cambios de estados.</p>
Informe de dificultades de fabricación	
<p>Noción</p> <ul style="list-style-type: none"> • Formulario con datos pertenecientes a una porción del programa de fabricación que no pudo ser ejecutado por el encargado de planta de un turno. <p>Impacto</p> <ul style="list-style-type: none"> • Confeccionado por el encargado de planta del turno. • Enviado por el encargado de planta a la oficina de planificación. • Utilizado para confeccionar el programa de fabricación extraordinario. 	<p>Noción</p> <ul style="list-style-type: none"> • Planilla de papel que contempla dificultades de fabricación encontradas en un programa de fabricación con una orden de producción urgente. <p>Impacto</p> <ul style="list-style-type: none"> • Es completado por el encargado de turno. • Se utiliza para informar dificultades de producción a la oficina de planificación de la producción.

Figura 96 – Ejemplo de un Símbolo Objeto con diferentes patrones sintácticos (Caso SNor)

Puede observarse en la Figura 96 que no existe mucha diferencia entre ambos años, esto se debe a que la percepción de los objetos no trae mayor dificultad. Las diferencias observadas corresponden a la percepción del contexto de cada grupo.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

SUJETO	
2006	2019
<p>Noción: Dejar claro quién es el sujeto.</p> <p>Impacto: Registrar las responsabilidades y actividades que realiza o recibe</p>	<p>Noción: Dejar claro quién es el sujeto. Identificar si es parte de una jerarquía. Identificar si tiene estados. Identificar aspecto temporal.</p> <p>Impacto: Registrar las actividades/tareas que realiza. Registrar qué actividad se realiza sobre el sujeto. Identificar “es/deber ser” o “será”. Identificar cambios de estados.</p>
Encargado de planta	
<p>Noción</p> <ul style="list-style-type: none"> Persona de turno a cargo de la línea de producción. Pueden estar a cargo del primer turno del segundo o del tercero. <p>Impacto</p> <ul style="list-style-type: none"> Recibe las órdenes de producción parcialmente completadas del encargado de planta del turno anterior. Entrega al encargado de planta del turno siguiente las órdenes de producción parcialmente completadas. Elabora listado de órdenes de producción parcialmente completadas. Envía al oficial planificador2 las órdenes de producción terminadas. Envía al oficial planificador2 el listado de órdenes de producción parcialmente completadas una vez elaborado. Ejecuta programa de fabricación perteneciente a su turno. Puede Ejecutar programa de fabricación perteneciente a los turnos siguientes. Puede Ejecutar programa de fabricación perteneciente a los turnos anteriores. Informar dificultad de fabricación que no le permite ejecutar programa de fabricación. Confecciona Informe de dificultades de fabricación Envía a la oficina de planificación el informe de dificultades de fabricación confeccionado. Recibe programa de fabricación extraordinario de la oficina de planificación Recibe telefónicamente un programa de fabricación extraordinario. Exhibe el programa de fabricación extraordinario al jefe de producción. Ejecuta programa de fabricación extraordinario. 	<p>Noción</p> <ul style="list-style-type: none"> Persona a cargo de la línea de producción Tiene el programa de fabricación para su turno y para algunos turnos siguientes. <p>Impacto</p> <ul style="list-style-type: none"> Envía información Ejecuta programa de fabricación Entrega las PO incompletas al encargado de planta. Recibe las PO incompletas del encargado de planta del turno anterior. Confecciona programa de producción extraordinario. Debería realizarlo el Oficial Planificador. Completa informe de dificultades de fabricación.

Figura 97 - Ejemplo de un Símbolo Sujeto con diferentes patrones sintácticos (Caso SNor)

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

En la Figura 97 nuevamente aparecen las diferencias en la percepción del contexto. En el 2006 determinaron que “turno” era un símbolo del LEL, mientras que en 2019 no lo consideraron con particularidades diferentes a las que tiene un turno en cualquier dominio. Otra diferencia consiste en que en 2019 se puede apreciar una relación mucho más estrecha entre los impactos del sujeto y los verbos del LEL. Nuevamente en 2006 se puede ver una mayor tendencia a describir procesos. También en este caso, se considera que todas las aclaraciones de la presente tesis han ayudado a mejorar la forma de describir el LEL. En el ejemplo de 2019 se puede observar la inclusión del punto de vista “Deber ser” en el impacto:

[Confeciona programa de producción extraordinario.](#) **Debería** realizarlo el Oficial Planificador.

VERBO	
2006	2019
<p>Noción: indicar quien la ejecuta, cuando sucede y la información necesaria para la ejecución de la acción</p> <p>Impacto: registrar las operaciones / procedimientos envueltos en la acción e identificar las situaciones que impiden que la misma se realice, qué otras acciones desencadenan en el ambiente y qué situaciones son causadas por la acción. Indicar que estados se desencadenan a partir de esta acción.</p>	<p>Noción: Describir concisamente la actividad que representa identificando quién lo ejecuta, cuándo (cronológicamente o relativo a otras actividades) y dónde se realiza. Identificar si es parte de una jerarquía. Identificar si tiene estados.</p> <p>Impacto: Actividades/tareas involucradas. Identificar las situaciones que impiden que la misma se realice. Qué otras acciones desencadenan en el ambiente y qué situaciones son causadas por la actividad. Indicar los cambios de estado que provoca la actividad. Identificar “es/deber ser” o “será”.</p>
Ejecutar programa de fabricación	
<p>Noción</p> <ul style="list-style-type: none"> El encargado de planta inicia la línea de producción para fabricar las cajas o accesorios solicitados en una orden de producción que corresponden a un programa de fabricación de un turno dado. <p>Impacto</p> <ul style="list-style-type: none"> Si no se puede cumplir con una porción del programa de fabricación que contiene una orden de producción urgente, se debe informar dificultad de fabricación. Si no se completa una orden de producción en el turno establecido por el programa de fabricación se la entrega al encargado del turno siguiente. 	<p>Noción</p> <ul style="list-style-type: none"> Es ejecutado por el encargado de planta durante su turno, luego de recibir el programa de fabricación. Lo realiza en la fábrica. <p>Impacto</p> <ul style="list-style-type: none"> El encargado de planta ejecuta las órdenes de producción contenidas en su programa de fabricación. Si la orden de producción fue completada en el turno asignado, el encargado de planta le asigna el estado terminada. Si la orden de producción no fue completada en el turno asignado, el encargado de planta le asigna el estado parcialmente completada. Si por algún inconveniente imprevisto alguna parte de su programa de fabricación no puede cumplirse, el encargado de planta

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 12 – Cambios y Agregados al LEL

<ul style="list-style-type: none"> Si se completa una orden de producción se la envía al oficial planificador2 al terminar el turno. 	está autorizado a ejecutar parte del programa de producción del o de los turnos siguientes.
---	---

Figura 98 – Ejemplo de un Símbolo Verbo con diferentes patrones sintácticos (Caso SNor)

En la Figura 98 se puede observar con mayor claridad, en la columna 2019, un uso más intenso de los estados (terminada y parcialmente completada).

ESTADO	
2006	2019
<p>Noción: dejar claro qué significa y qué actividades y acciones llevaron a este estado</p> <p>Impacto: Identificar otros estados y actividades/acciones que pueden ocurrir a partir de este estado.</p>	<p>Noción: Dejar claro qué significa y a qué aplica. Qué actividades, estados y acciones llevaron a este estado. Identificar si compone una jerarquía. Identificar si tiene estados.</p> <p>Impacto: Identificar otros estados y actividades/tareas que pueden ocurrir a partir de este estado. Identificar “es/deber ser” o “será”.</p>
Alianza y Urgente	
<p>Alianza</p> <p>Noción</p> <ul style="list-style-type: none"> Convenio estratégico entre la fábrica y la Papelera del Sudeste. <p>Impacto</p> <ul style="list-style-type: none"> Permite recibir órdenes de producción urgente de la Papelera del Sudeste. Permite enviar a la Papelera del Sudeste una orden de producción que ocasiona dificultades de producción. 	<p>Urgente</p> <p>Noción</p> <ul style="list-style-type: none"> Es cuando la orden de producción tiene una entrega menor a 10 días. Es cuando la orden de producción es recibida desde Papelera Sudeste. <p>Impacto</p> <ul style="list-style-type: none"> Se aumenta la prioridad de la orden de producción La orden de producción es incorporada en el programa de fabricación más cercano <p>Otros Estados detectados para la OP: Completa, Incompleta y Terminada. Estados para el Programa de fabricación: extraordinaria.</p>

Figura 99 - Ejemplos de Símbolos Estados con diferentes patrones sintácticos (Caso SNor)

En la Figura 99 se describen distintos estados porque no ha existido coincidencia. Se puede observar que la cantidad de estados detectados en 2019 es mayor. Se considera que una mejor descripción de su identificación, como se propone en la presente tesis, ayuda a detectarlo muy tempranamente. En el ejemplo de 2006, los estados se encuentran como sustantivos (“orden de producción urgente”).

Con respecto a las jerarquías conceptuales del contexto, se puede observar en la Figura 100 las diferencias en la percepción léxica entre ambos trabajos.

Jerarquías Conceptuales del Contexto	
2006	2019
<p>Accesorio Noción</p> <ul style="list-style-type: none"> • Elemento de cartón corrugado. • Puede acompañar a una caja a medida. • Puede ser de dos tipos, divisor o separador. <p>Impacto</p> <ul style="list-style-type: none"> • Solicitado a través de una orden de compra. • Pedido por el cliente. • Puede ser fabricado al ejecutar un programa de fabricación. • Puede ser fabricado al ejecutar un programa de fabricación extraordinario. • Entregado al cliente. <p>Divisor Noción</p> <ul style="list-style-type: none"> • Pieza de Cartón Corrugado usada para fragmentar el espacio interior de la caja a medida en forma horizontal. <p>Impacto</p> <ul style="list-style-type: none"> • Pedido por el cliente. <p>Separador Noción</p> <ul style="list-style-type: none"> • Pieza de Cartón Corrugado usada para fragmentar el espacio interior de la caja a medida en forma vertical. <p>Impacto</p> <ul style="list-style-type: none"> • Pedido por el cliente. 	<p>Accesorio Noción</p> <ul style="list-style-type: none"> • Es un producto elaborado. • Es un elemento que acompaña a la caja de cartón corrugado. • Puede ser un divisor o un separador. <p>Impacto</p> <ul style="list-style-type: none"> • Es solicitado por el cliente • Se lo fabrica <p>Divisor Noción</p> <ul style="list-style-type: none"> • Es un accesorio. • Separa el contenido de la caja en capas. <p>Impacto</p> <ul style="list-style-type: none"> • Es usada para fragmentar el espacio interior de la caja de cartón corrugado de forma horizontal. <p>Separador Noción</p> <ul style="list-style-type: none"> • Es un accesorio. • Separa el contenido en porciones iguales (ej. para botellas, vasos, frascos) <p>Impacto</p> <ul style="list-style-type: none"> • Es usada para fragmentar la caja de cartón corrugado de manera vertical, formando casilleros.

Figura 100 – Ejemplo de Jerarquías Conceptuales del contexto (Caso SNor)

Es necesario aclarar que el ejemplo de la Figura 100 es el único caso, entre casi un centenar de LEL analizados, donde se encontró la presencia de jerarquías. La identificación y representación de jerarquías se incrementó considerablemente una vez que se comenzó a hablar de ellas. Obviamente, como en todas las profesiones, existen ingenieros/as de requisitos con una mayor capacidad de observación, pero las calidades de los requisitos del software no pueden descansar en eso. La IR debe proveer mecanismos de construcción lo más seguro posible, siendo esto una de las motivaciones de la presente tesis.

Puede observarse que en esta comparación persiste el problema de la completitud del LEL, ya que en algunos casos aparecen impactos en 2006 no registrados en 2019 y viceversa.

Otro aspecto a considerar es que en este caso es una estructura jerárquica de elementos de producción tangibles que se ven, obviamente, reflejados en el léxico, por lo que su detección es más simple que cuando las jerarquías son intangibles. Finalmente, debe considerarse que en el ejemplo se han seleccionado del LEL sólo los símbolos necesarios para mostrar la jerarquía, pero cuando estos símbolos están dispersos en el glosario la dificultad para relacionarlos es mucho mayor. De esta manera, se puede observar la importancia del ordenamiento que le otorgan las jerarquías a un léxico.

Capítulo 13

Actualización de la Inspección del LEL

Resumen

Obtener un LEL de calidad depende en gran medida de un proceso de verificación sólido. Debido a la extensión y la gran cantidad de detalles que tiene esta actividad no es posible incorporar en el presente documento todos los cambios y agregados; es así que en este Capítulo se presenta un resumen de los formularios afectados y un par de formularios nuevos sólo a modo de ejemplo.

La Inspección del LEL se presenta en forma completa en el **Anexo A – Verificación de consistencias en el LEL y de los Escenarios**. Cabe aclarar que en dicho anexo se encuentra la inspección del LEL [Kaplan00] y la de Escenarios [Doorn98] [Leite05]. A continuación, se describe la actualización del LEL, mientras que la de los Escenarios se describe en el **Capítulo 17**.

Cambios en la Inspección del LEL

En el **Anexo A** se encuentra la “Tabla LEL” que corresponde a los formularios de inspección del LEL (de la página 2 a la 21). Esta inspección fue actualizada para incorporar los cambios de la presente tesis. A tal efecto, en la Tabla 27, se describen los formularios existentes que se ven afectados y los que deberán ser incorporados en el futuro.

Títulos de Formularios Existentes	Modificaciones realizadas
ANÁLISIS DE LAS NOCIONES E IMPACTOS	<ul style="list-style-type: none"> Incorporar Homónimos
VERIFICACION DE SÍMBOLOS TIPO OBJETO	<ul style="list-style-type: none"> Cumplimiento de las nuevas sugerencias del Patrón Sintáctico objeto
VERIFICACION DE SÍMBOLOS TIPO SUJETO	<ul style="list-style-type: none"> Cumplimiento de las nuevas sugerencias del Patrón Sintáctico sujeto
VERIFICACION DE SÍMBOLOS TIPO VERBO	<ul style="list-style-type: none"> Cumplimiento de las nuevas sugerencias del Patrón Sintáctico verbo
VERIFICACION DE SÍMBOLOS TIPO ESTADO	<ul style="list-style-type: none"> Cumplimiento de las nuevas sugerencias del Patrón Sintáctico estado
Títulos de Formularios Nuevos	Modificaciones realizadas
IDENTIFICACIÓN DE SÍMBOLOS DEL MACROSISTEMA	<ul style="list-style-type: none"> Identificar símbolos sin impactos
DETECCIÓN DE JERARQUÍAS INCOMPLETAS – IDENTIFICAR GENÉRICOS PRÓXIMOS (TIPO B Y E)	<ul style="list-style-type: none"> Identificar jerarquías incompletas Identificar jerarquías no identificadas
DETECCIÓN DE JERARQUÍAS INCOMPLETAS - IDENTIFICAR ESPECIALIZACIONES (TIPO C y D)	<ul style="list-style-type: none"> Identificar jerarquías incompletas Identificar jerarquías no identificadas
VERIFICACIÓN DEL PRINCIPIO DE CIRCULARIDAD EN LAS JERARQUÍAS	<ul style="list-style-type: none"> Falta de identificación dentro del jerarquía Omisión de símbolos hacia fuera de la jerarquía y hacia la propia
VERIFICACIÓN DEL PUNTO DE VISTA “ES”, “DEBER SER” y “PRÓXIMO”	<ul style="list-style-type: none"> Falta de identificación del punto de vista

Tabla 27 – Actualización a la Inspección del LEL

Actualización de la Inspección del LEL

Como se mencionó en la introducción, este Capítulo tiene por finalidad mostrar los cambios realizados para actualizar la inspección. Se presentan a continuación dos

formularios nuevos, los cuales facilitan la verificación de la existencia de homónimos y el uso correcto de algunas jerarquías conceptuales del contexto.

Detección de Homónimos

Al revisar la inspección existente se encontró que el mismo formulario que detecta sinónimos, también puede detectar **homónimos**. La búsqueda de sinónimos consiste en detectar símbolos con distinto nombre, pero semánticamente equivalentes. Por el contrario, los homónimos son aquellos símbolos que tienen el mismo nombre, pero no significan lo mismo. El formulario que detecta sinónimos es el Form IV, el cual ayuda a realizar una comparación semántica entre todos los símbolos existentes, con el objetivo de detectar símbolos iguales. Al revisar este formulario, se observó que la comparación también puede detectar homónimos que no fueron detectados oportunamente. Este ejemplo es importante porque muestra la posibilidad de reutilizar los formularios existentes, ampliando su capacidad de detección. A continuación, se presenta un ejemplo donde “Asignatura” y “Materia” son sinónimos. Cuando la palabra “Lengua” se refiere a un curso y también se refiere a una forma de comunicación (lengua de señas, lengua gauchesca, etc.) son homónimos.

SÍMBOLO	CANT NOCIÓN	CANT IMPACTO	SÍMB.	CANT NOCIÓN	CANT IMPACTO	NOCION IGUALES %	IMPACTO IGUALES ¹	OBS*
Asignatura	2	2	Materia	3	2	2	2	SINÓNIMO
			Libro	2	2	0	0	-----
			...					
Lengua	3	2	Lengua	2	2	0	0	HOMÓNIMO
			Cursada	1	1	0	0	-----
			...					

Form. IV – Análisis de las nociones e impactos

Puede observarse que los ejemplos de los formularios de inspección algunos tienen una numeración (ejemplo, *Form. IV*) mientras que otros no. Esto se debe a que existen modificaciones a formularios preexistentes (ver Anexo A), a los cuales se les ha respetado su enumeración original. Los formularios que se presentan nuevos no se han enumerado debido a que no se ha definido aún cómo serán insertados en el conjunto de formularios.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 13 – Actualización de la Inspección del LEL

Es necesario aclarar que antes de tomar la decisión de incluir los homónimos es necesario analizar el UdeD para asegurarse que efectivamente sean homónimos y que no se confunda con una jerarquía o estados.

Otro caso de homónimos y quizás el más complejo, es cuando un símbolo aglutina dos o más símbolos diferentes. En este caso, se debe analizar la correlación entre la noción y el impacto, ya que se deben buscar impactos que no se relacionan con alguna parte de la noción. Para detectar esta situación se creó un **nuevo formulario** denominado “*Símbolos superpuestos*”. Este formulario detecta homónimos, pero también otros casos de superposición, por errores de descripción o de interpretación del contexto.

SÍMBOLO	NOCIÓN	IMPACTO	¿ESTÁN RELACIONADOS?	OBS
Libro	A	1	Si	
		2	Si	
		3	No	*
	B	1	No	*
		2	No	*
		3	Si	

Nuevo Form.1 – Símbolos superpuestos

Cada símbolo del LEL tiene una relación muy estrecha entre sus componentes, o sea entre la noción y el impacto. Cuando esta relación se rompe es una alerta para el ingeniero/a de requisitos. En el ejemplo del *Nuevo Form.1*, existe el símbolo “Libro” que tiene 2 nociones y 3 impactos. Para comprobar esa relación, se compara cada Noción con todos los Impactos. O sea, la Noción A con los tres impactos y la Noción B con los tres impactos. Obviamente, esto se repite para cada símbolo. Puede observarse que el Impacto 3 no tiene relación con la Noción A, pero sí la tiene con la Noción B. En cambio, los Impactos 1 y 2 se relacionan con la Noción A, pero no con la B. En estos casos, primero se debe asegurar que no se trata de información de otro símbolo. De ser así, se debe trasladar la noción e impacto que corresponda al símbolo en cuestión. En segundo lugar, si este símbolo no existe, se debe crear el nuevo símbolo.

Detección de Jerarquías Incompletas

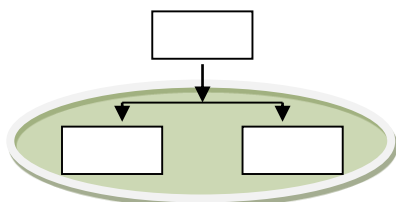
El formulario nuevo *Detección de Jerarquías incompletas – Identificar genéricos o agregados próximos (Tipo B y E)*. El tipo de jerarquías incompletas que detecta este

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

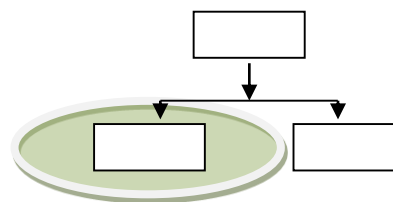
Capítulo 13 – Actualización de la Inspección del LEL

formulario, son aquellas donde las especializaciones o las partes mencionan un genérico o un agregado que no es símbolo del LEL. Se reiteran aquí las jerarquías incompletas b) y e) de la Figura 76 relacionadas con este formulario de inspección:

b) ausencia del genérico o agregado más próximo



e) ausencia del genérico o agregado más próximo y de la especialización o parte



SÍMBOLO	Noción e Impacto	Frases para jerarquías	Genérico próximo	¿Es símbolo?	OBS
CAJA CON SOLAPA	1	“es una caja con ...”	CAJA	SI	La jerarquía ya existe
	2	---	---	---	
	3	---	---	---	
	1	---	---	---	
	2	---	---	---	
ADMINISTRATIVO	1	“es una persona ...”	PERSONA	NO	Posible jerarquía omitida
	2	---	---	---	
	1	---	---	---	
JEFE ADMINISTRATIVO	1	---	---	---	
	1	“El empleado le avisa al jefe ...”	JEFE	NO	Posible jerarquía omitida

Nuevo Form.2 Detección de Jerarquías Incompletas – Identificar genéricos próximos

Para completar el *Nuevo Form.2* se han enumerado las nociones e impactos sólo para una mayor facilidad de llenado. En ambos casos, se comienza en 1, esto permite diferenciar cuando es noción y cuando impacto. Se debe determinar para cada Noción, si aparece alguna oración del tipo “es un ...” o “es parte ...” o semejante, donde se puede estar mencionado un genérico o un agregado. Si aparece se debe completar la columna Genérico o Agregado Próximo y determinar si está marcado como símbolo del LEL (negrita, subrayado, etc.) y luego, verificar que efectivamente exista en el LEL. De ser afirmativo, la jerarquía ya fue incluida y de no serlo, se puede estar en presencia de una posible omisión. En el caso del genérico “persona”, es correcto que no sea un símbolo del LEL excepto que tenga alguna característica particular en el contexto. Estos pasos se

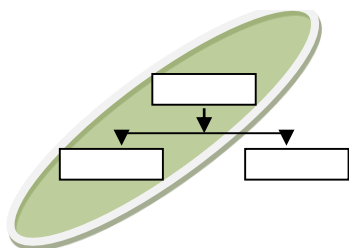
Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 13 – Actualización de la Inspección del LEL

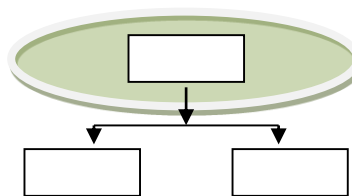
repiten para los impactos, pero ahora se busca la presencia de algún posible genérico en el texto, como es el caso de “jefe”. Como en todos los casos, la decisión de registrar la jerarquía en el LEL queda a criterio del autor del LEL, quien debe cerciorarse en el UdeD que la misma realmente existe.

Para completar la detección de jerarquías incompletas, se presenta otro formulario para la *Detección de Jerarquías Incompletas – Identificar especializaciones o componentes (Tipo C y D)*, el cual permite identificar los símbolos de la especialización o componente que están en la noción, pero no son símbolos del LEL. Nuevamente, para mejorar la lectura, se recuerdan las jerarquías incompletas de la Figura 76 involucradas en este formulario de inspección:

c) Jerarquía Incompleta: omisión de otras especializaciones o componentes



d) Jerarquía Incompleta: ausencia de todas las especializaciones o componentes



SÍMBOLO	NOCIÓN	LISTA DE ITEMS ¹	ELEMENTOS DE LA LISTA	CANTIDAD DE SÍMBOLOS	OBS
CAJA CON SOLAPA	Está compuesta por ...	SI	5	5	Jerarquía completa
	Es un artículo que se vende	No	--	--	Va en el impacto
ADMINISTRATIVO	Son aquellas personas que trabajan en el Stock, en Atención al Cliente y en Soporte	SI	3	2	Es posible que el que falta sea un símbolo omitido
	Puede realizar jornada completa o media	NO	--	--	

¹ lista de especialización que debe estar en la noción del genérico.

Nuevo Form.3 Para la Detección de Jerarquías Incompletas – Identificar especializaciones o componentes

Se puede observar en el *Nuevo Form.3*, que en el símbolo “Administrativo” se reconoce una lista de ítems, pero sólo existen 2 de las 3 especializaciones mencionadas como símbolos del LEL. En este caso, se puede estar en presencia de una jerarquía incompleta, o sea, de una especialización anónima. Se utiliza el potencial porque puede suceder que

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 13 – Actualización de la Inspección del LEL

no todos los elementos sean símbolo del LEL. Esto sucede, por ejemplo, en jerarquías conceptuales que exceden el contexto. La inspección pone en alerta al ingeniero/a de requisitos, para que se asegure que no se omitió el símbolo.

Trabajo Futuro

En el presente Capítulo se realizó una revisión preliminar de la Inspección del LEL, la cual debe ser continuada para asegurar que la inspección realmente mejore la consistencia y completitud del LEL. Queda pendiente para trabajos futuros:

- Analizar los formularios existentes para incluir otros aspectos de la verificación, tal es caso de los homónimos incorporados al “Form IV”. Al estudiar los formularios existentes se previene de construir formularios innecesarios.
- Crear los nuevos formularios restantes, con los cambios que no pueden ser incluidas en el punto 1.
- Crear nuevos formularios con aspectos importantes del LEL que no fueron incluidos en la inspección original. Como ser: la relación entre sujetos y verbos, el nivel de abstracción de los impactos de los sujetos, la presencia de cadenas de estados y su debida representación en el LEL, etc.

Se debe revisar la inspección del LEL para determinar qué se debe modificar o agregar para el LEL_R (ver **Capítulo 15**).

Capítulo 14

Construcción de LEL por Proximidad

Resumen

Uno de los resultados del estudio empírico realizado en el **Capítulo 10**, determinó la necesidad de robustecer el proceso de Construcción del LEL. Siendo que los cambios necesarios para alcanzar este objetivo eran de tal relevancia se decidió crear una nueva heurística de construcción.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 14 – Construcción del LEL por proximidad

Durante el desarrollo de la nueva heurística se consideraron dos aspectos relevantes, la dispersión de la información en las diferentes fuentes de información y que la atención del ingeniero/a de requisitos se desperdigue lo menos posible. Pero una estrategia de construcción del LEL que proponga concentrarse en pocas cosas, conlleva a una contradicción. Si la actividad se concentra en unos pocos símbolos obliga a recorrer todas las fuentes de información y esto es más disperso que ver todos los símbolos simultáneamente. Entonces, es preferible concentrarse en una fuente de información hasta agotarla de manera de reducir al máximo posible la necesidad de volverla a consultar.

En el estudio empírico del **Capítulo 10** fueron analizados diferentes factores, internos y externos, que afectan la calidad del LEL. Entre las desventajas más significativas que se analizaron oportunamente, se encuentra la generación de una lista inicial de símbolos, la cual actúa como columna vertebral de todo el proceso de construcción del glosario, pero congela tempranamente la composición del léxico que se desea representar. Esto obligó a buscar un mecanismo más eficiente que incluya, además, todos los cambios y agregados presentados en el **Capítulo 12**, que le dan al glosario mayor consistencia y completitud.

Es importante destacar que la nueva heurística no elimina la lista de símbolos candidatos, sino que le da un uso muy diferente. Esta lista se crea a partir de un grupo de unos pocos símbolos iniciales y luego, se amplía y revisa a lo largo de todo el proceso de construcción con el objetivo de guiar al ingeniero/a de requisitos acerca de los símbolos que se deben describir.

Construcción del LEL por Proximidad

Como se puede observar en la Figura 104 la propuesta es una estrategia IID (iterativa e incremental) que expande el LEL mediante el agregado de símbolos por etapas. Durante el modelado de cada símbolo aparecen nuevas palabras o frases relevantes del macrosistema, las cuales deben ser consideradas como potenciales nuevos símbolos. Este mecanismo tiene la característica de ir avanzando mediante la detección de símbolos cercanos. O sea, que a partir de un símbolo se detectan otros y durante la descripción de estos aparecen otros y así hasta completar el léxico del macrosistema observable. Este proceso de búsqueda y elicitación se realiza sobre la literatura del macrosistema o documentación organizacional, siempre y cuando exista, y sobre todo aquello elicitado de los involucrados. Cabe recordar que el Proceso de Requisitos en cuestión ha sido pensado para organizaciones medianas y grandes donde es habitual contar con documentos,

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 14 – Construcción del LEL por proximidad

manuales, estándares, etc. Utilizar la documentación organizacional permite ordenar la información e identificar las diferencias que pueden existir entre lo operativo y la mirada de la alta esfera organizacional. Este tema es de suma importancia ya que de no contar con ambos puntos de vista es posible que se construya un software que no cumpla con las necesidades reales, sean estas las operativas como las directivas. Las posibles diferencias entre el punto de vista “Es” y el “Deber ser” se obtienen al analizar la literatura del macrosistema y contrastarla con la información obtenida de las personas que realizan las tareas o entre diferentes puntos de vista de las personas que responden a diferentes niveles jerárquicos en la pirámide organizacional. Cabe destacar que una misma persona no siempre tiene un único punto de vista, sino que el mismo puede fluctuar dependiendo del rol que toma en la organización en cada momento.

En la nueva heurística, la elicitación se concentra tanto en la literatura del macrosistema como en las personas. Uno de los documentos más importantes en este proceso son las transcripciones de las entrevistas. Para contar con ellas se sugiere grabar las entrevistas y luego, transcribirlas.

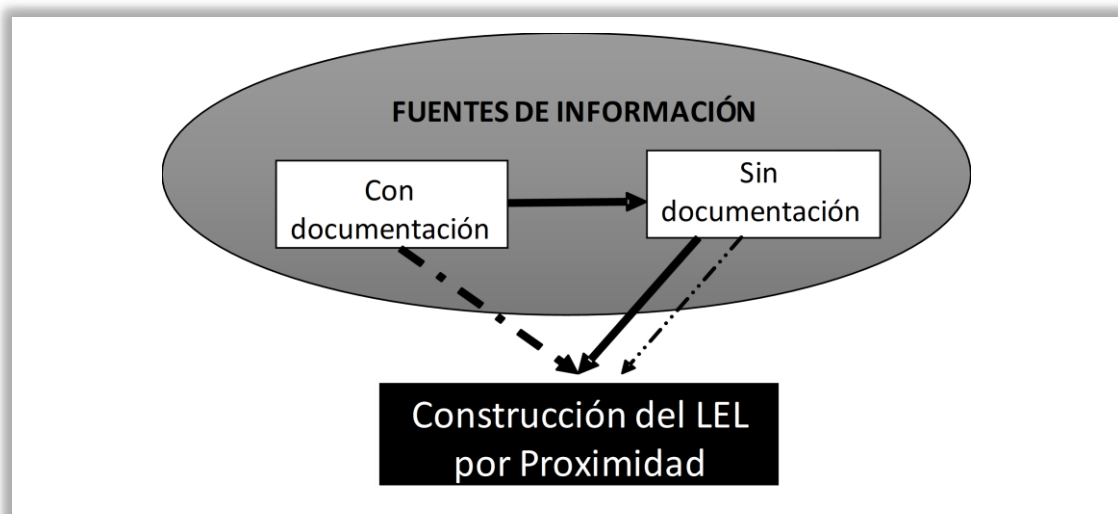


Figura 101 – Fuentes de información en la construcción del LEL por Proximidad

Puede observarse en la Figura 101 que existen tres caminos posibles para obtener la información necesaria para construir el LEL por proximidad. Cada camino se representa con diferente tipo de línea. Cada camino indica una forma de trabajar diferente según las fuentes de información disponibles. Los caminos con líneas punteadas indican dos casos extremos donde existe un tipo de fuente de información que prevalece, “Con Documentación” (predomina la literatura del macrosistema) y “Sin Documentación” (predomina la información de las personas). Mientras que el camino con línea sólida, es

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 14 – Construcción del LEL por proximidad

el esperado, donde existen diferentes tipos de fuente de información. A continuación, se describen los tres caminos:

- 1) El camino con *líneas punteadas gruesas* se establece cuando se puede acceder a la documentación de la organización o la documentación del dominio y el acceso a las personas es muy limitado o nulo. Este caso, si bien extremo, sucede en los contextos donde se debe acatar estrictamente alguna reglamentación, normativa, protocolo, etc. Son contextos donde predomina el punto de vista “Deber ser” presuponiéndose que no existen apartamientos importantes del mismo. Por ejemplo, en Argentina existe un organismo descentralizado de la Administración Pública Nacional denominado Administración Nacional de Medicamentos, Alimentos y Tecnología Médica (ANMAT) que colabora en la protección de la salud humana garantizando los medicamentos, los alimentos y los dispositivos médicos. Para lograr estos objetivos reglamenta los procedimientos y crea estrictos protocolos que las organizaciones deben cumplir. De ser necesario construir un nuevo Sistema de Software en el marco de estas disposiciones, la fuente de información que prevalece son las disposiciones publicadas por la ANMAT que afecten el contexto en estudio.
- 2) El camino con *líneas punteadas finas*, sucede cuando no existe documentación del macrosistema o la misma no resulta confiable, no siendo prudente su uso. En este caso predomina el punto de vista “Es” o del contexto observable. A diferencia del caso anterior, detectar la existencia del otro punto de vista, requiere de comparaciones estrictas entre lo elicitado con el personal de diferentes niveles organizacionales (operativos, tácticos o estratégicos). Por ejemplo, en contextos donde la documentación no es actualizada sistemáticamente.
- 3) Finalmente, el caso de líneas sólidas permite acceder tanto a la literatura del macrosistema o del dominio como a las personas involucradas. De esta manera se pueden identificar y contrastar los diferentes puntos de vista existentes. En este caso, como se muestra en la Figura 102, es posible relacionar la documentación con la información proveniente de las personas, como también las posibles variantes internas de cada fuente de información. Por ejemplo, en a) se puede contrastar la información de un manual de calidad con un manual de procedimientos; en b) la información de la dirección con la del jefe o la de este último con un operario y c)

la información de un manual de procedimientos con la información proveniente de un operario.

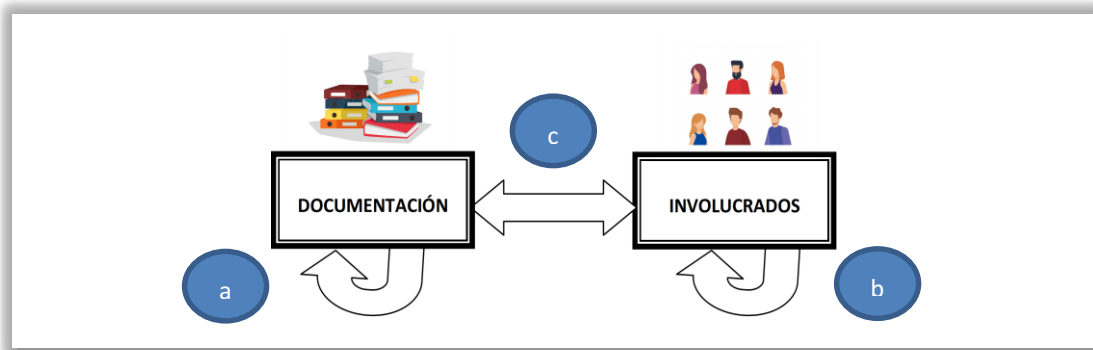


Figura 102 – Relación entre las diferentes fuentes de información

Cabe aclarar que es muy poco usual la ausencia total de personas que colaboren con el Proceso de Requisitos. Pueden existir contextos donde la participación de los involucrados sea débil, pero rara vez es nula.

Cada FI disponible en el contexto tiene una perspectiva propia del macrosistema, la cual se relaciona mejor con determinados tipos de símbolo. Por ejemplo:

1) Documentos:

- a. Cuando describen funciones/actividades/responsabilidades (manual de organización y funciones, manual de calidad, etc.) se recomienda comenzar por símbolos *sujetos*.
- b. Cuando describen procesos (manual de procedimientos, protocolos, estatutos, etc.) se recomienda comenzar por símbolos *verbos*.
- c. Cuando describen productos (manual de Producto, especificaciones técnicas, etc.) se recomienda comenzar por símbolos *objetos*.

2) Involucrados: se deben realizar las entrevistas de manera descendente, o sea comenzar por los clientes-usuarios de mayor responsabilidad e ir bajando.

Heurística por Proximidad

La presente heurística es un cambio de estrategia en la forma de elicitar información y usar las fuentes de información disponibles. Al construir el LEL por proximidad se debe prestar mucha atención al Principio de Circularidad ya que algunos símbolos que deben referenciarse aún no han sido descriptos. Es necesario que el ingeniero/a de requisitos esté

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 14 – Construcción del LEL por proximidad

atento a cumplir dicho principio, aunque algunas omisiones se pueden corregir al verificar el LEL (ver Anexo A) con un esfuerzo mayor.

Cabe recordar que, al llegar a la presente heurística, el ingeniero/a de requisitos ya realizó previamente algunas entrevistas donde detectó o definió con el cliente el Objetivo General del Sistema. Este objetivo es la primera definición concreta sobre el contexto en estudio que existe en un proceso de construcción del software y se considera disponible a partir de ese momento.

Para describir la heurística se presenta en la Figura 103 una síntesis de la misma utilizando la simbología de Diagramas de Nassi-Shneiderman [Nassi73] [Aguilar88]. Cabe aclarar que en la presente tesis se han tomado algunas libertades de representación para mejorar la lectura.

La Figura 103 presenta dos caminos posibles. La propuesta es, siempre que sea posible, comenzar por las FI que representen el “Deber ser”, o sea la literatura del macrosistema o las personas con roles jerárquicos dentro de una organización. Cabe destacar que siempre que exista un documento confiable del macrosistema, se debe comenzar por él. Luego, se va accediendo a las otras FI, según proponga la heurística. La propuesta de comenzar por el punto de vista del “deber ser” es tener la posibilidad de compararlo con el punto de vista “es” que se obtiene de los niveles medios (jefes) y de las personas que realizan las tareas. La primera actividad de la heurística, “**Identificar símbolos relevantes relacionados con el Objetivo General del Sistema**”, permite obtener los *símbolos semilla* que actuarán como núcleo de cristalización alrededor de los cuales se construirá el LEL. Para identificar los *símbolos semilla* se debe analizar el Objetivo General del Sistema. De no estar claro el Objetivo General del Sistema, es el momento ideal para precisarlo junto al cliente. Es conveniente que estos símbolos sean los más representativos, pero esto no es una cuestión esencial, ya que el procedimiento iterativo siempre convergerá rápidamente hacia los símbolos más importantes, por lo que la mencionada conveniencia sólo está relacionada con la cantidad de iteraciones necesarias.

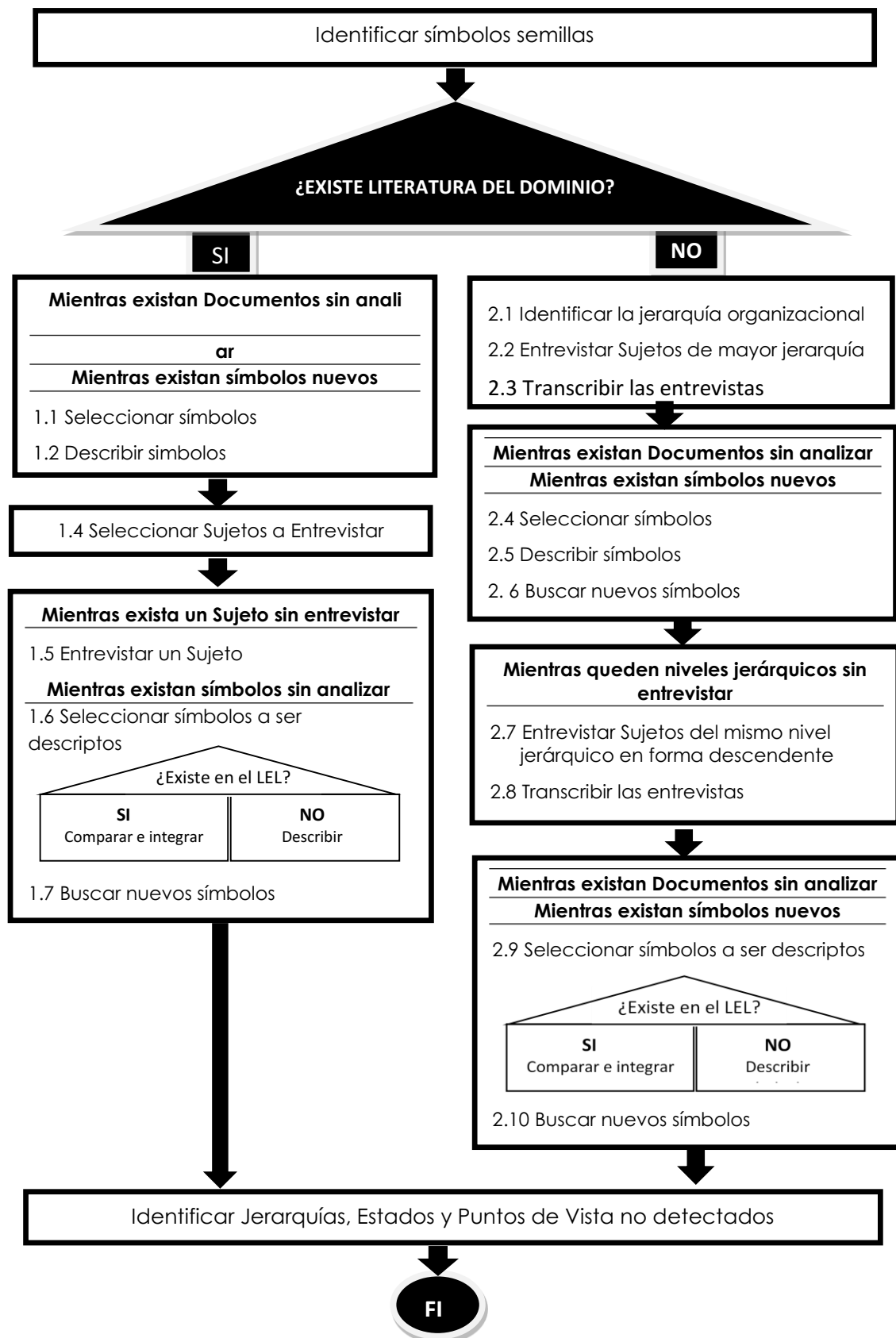


Figura 103 –Heurística del Construcción del LEL por Proximidad

Con los símbolos semillas, se puede observar en la Figura 103 que se abren dos caminos, los cuales responden al siguiente interrogante:

¿Existe literatura confiable del macrosistema?

SI

Paso 1.1:

A partir de los símbolos semillas que se han incorporado a la lista de símbolos y sabiendo que existen documentos para analizar, se deben “*seleccionar los símbolos a describir*” para determinar el mejor documento por el cual comenzar, siempre que sea posible. En este caso se sugiere elegir aquel que tenga la misma perspectiva que los símbolos seleccionados. Por ejemplo, cuando el símbolo es tipo verbo y el documento es un manual de procedimientos, la información estará menos dispersa y es probable que se pueda completar bastante. Por otro lado, con el mismo manual de procedimientos y símbolos objetos, la información estará dispersa y es alta la probabilidad de que el símbolo quede muy incompleto. Como ya se mencionó, esta selección mejora el proceso de construcción del LEL, pero no es indispensable ya que en las próximas iteraciones se irá compensando.

Cada vez que se comienza con un documento, este se utiliza para *elicitar toda la información* posible. Esto significa incorporar el máximo posible de información en todos los símbolos pendientes de la lista de símbolos en diferentes iteraciones. Cuando este documento se agota recién se pasa a otro. De esta manera se incorpora en el LEL toda la información posible desde la literatura del macrosistema.

Paso 1.2:

Con los símbolos seleccionados y el mejor documento para elicitar información se deben “*Describir*”. Para ello se debe buscar información en todo el documento y completar lo que sea posible. Cabe recordar que el punto de vista de la literatura del macrosistema es el “deber ser” y por lo tanto no es necesario analizar los puntos de vista.

Durante la descripción se debe actualizar la lista de símbolos que irá creciendo a medida que se avanza en el proceso. Para comprender cómo se utiliza la Lista de Símbolos se presenta una figura descriptiva de la evolución de la misma a lo largo del proceso (ver Figura 104). Como ya se mencionó, se utiliza una tilde para indicar los

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 14 – Construcción del LEL por proximidad

símbolos que están en proceso de descripción y una doble tilde para indicar que está completo.

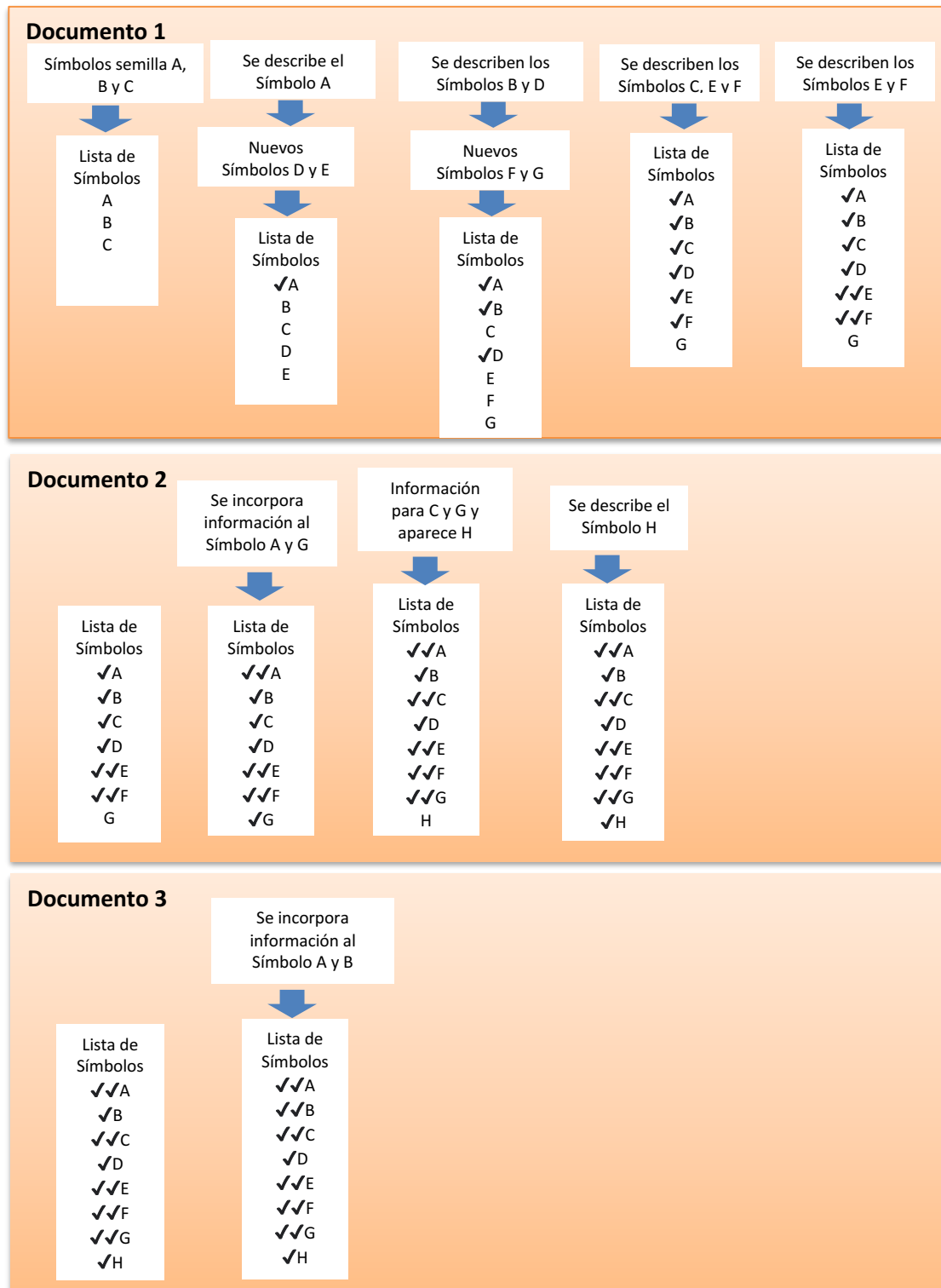


Figura 104 – Ejemplo de la lista de símbolos en la Construcción del LEL por Proximidad

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 14 – Construcción del LEL por proximidad

La lista comienza con los símbolos semilla. Luego, durante la elicitación sobre el Documento 1 se completa toda la información posible en los símbolos existentes y se buscan nuevos símbolos por proximidad, como es el caso de “D” y “E”. Estos símbolos se describen lo más posible. Luego, aparece “F” y “G”. Para el ingeniero/a de requisitos, los símbolos “E” y “F” están completos, por lo tanto, lo cierra con una doble tilde. Puede suceder que aparezca más información para estos símbolos cerrados en otra FI, en ese caso se la debe agregar. Luego, en el Documento 2, se cierra “A”, “C” y “G” y aparece “H”. Finalmente, en el documento 3, se completa la información de “B” pero se puede observar que “D” y “H” quedan con una sola tilde, por lo tanto, no se los consideran completos. Para cerrarlos, se deberá elicitación más información en el UdeD o durante la validación del LEL.

En resumen, se espera que a medida que se avanza en la descripción del LEL, cada nuevo documento que se consulte aporte detalles a los símbolos existentes y permita detectar nuevos símbolos. Es un proceso iterativo e incremental.

Paso 1.3:

Para “*buscar nuevos símbolos*” se debe analizar cada símbolo descrito buscando palabras o frases relevantes del contexto y que se encuentren dentro del marco del Objetivo General del Sistema. También se pueden encontrar nuevos símbolos analizando los empotramientos (ver **Capítulo 12**). Cada vez que aparece un símbolo nuevo se debe corroborar que no exista en la lista de símbolos. Si existe y corresponde su definición, se debe analizar la posibilidad de incorporarle información. Si existe, pero su definición no corresponde con lo esperado, incorporar un homónimo. Finalmente, si no existe, incorporar el símbolo a la lista. Una vez agotado el documento repetir desde el Paso 1.1 para todos los documentos disponibles.

Una vez que se han analizados todos los documentos organizacionales se puede contar con un LEL muy pobre o con uno completo, dependiendo de la calidad de los documentos consultados.

Paso 1.4:

El próximo paso es comenzar a trabajar con las personas del macrosistema. Esta actividad tendrá un esfuerzo inversamente proporcional a la cantidad y calidad de los documentos consultados, yendo de completar las descripciones a una temprana validación.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 14 – Construcción del LEL por proximidad

Para “*Identificar los sujetos a entrevistar*” se puede generar un mapa (ver Figura 105) que relacione los símbolos tipo sujeto con las personas del macrosistema. En contextos donde no existe una forma obvia de determinar a quién se debe entrevistar o existe un riesgo de dejar afuera personas importantes, se sugiere completar este mapa y de ser necesario incorporar más información de cada persona para decidir junto al cliente-usuario quienes son las personas más adecuadas. Seleccionar las personas con mayor conocimiento del macrosistema mejora la calidad de la Ingeniería de Requisitos. Por ejemplo, en una organización con sucursales, puede suceder que las personas con el mayor conocimiento del contexto estén dispersas y no ser vistas por el ingeniero/a de requisitos.

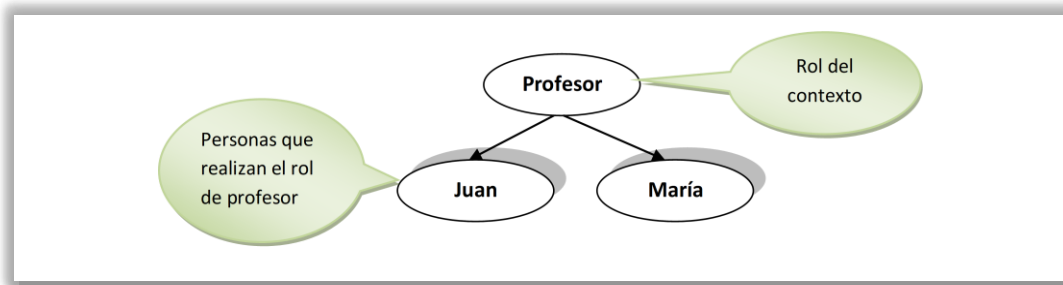


Figura 105 – Ejemplo de mapa de relación entre los roles y las personas

Paso 1.5:

Al “*entrevistar un sujeto*” se debe prestar una debida atención a cómo se realiza dicha entrevista, o sea, a la postura que toma el ingeniero/a de requisitos, teniendo en cuenta que se va a elicitarse específicamente terminología. Por lo tanto, se le debe preguntar al usuario-cliente “¿Ud. qué hace?” “¿Qué insumos necesita?” “¿Qué cosas se producen?” “¿Qué herramientas utiliza?” y dejarlo que se exprese libremente. Esto permite que aparezca el punto de vista del “es”, si existe. Siempre que sea posible y pertinente, preguntar si existe alguna otra propuesta para realizar las actividades o tareas mencionadas. Esta pregunta intenta detectar diferentes puntos de vista. Por cada sujeto entrevistado, se debe tratar de elicitarse toda la información posible, esto permite optimizar el tiempo de las personas. De ser posible, se sugiere grabar las entrevistas. De lo contrario, se debe tomar nota de todo lo que dice el cliente-usuario. Estas grabaciones permiten generar transcripciones, las que serán consideradas a partir de este momento como documentación del macrosistema.

Paso 1.6:

Para incorporar la información de los entrevistados en el LEL se debe “*Seleccionar Símbolos a describir*”. O sea, buscar cada símbolo de la lista de símbolos en el documento con el objetivo de incorporar nueva información. En caso que la lista no contenga símbolos incompletos, leer el documento en busca de nuevos símbolos.

Cada vez aparece información para un símbolo se debe “*comparar*” con la información existente en el símbolo. Puede ser más detalle para el símbolo, en este caso incorporarla. Si al analizar aparecer información con otro punto de vista, se debe “*integrar*” correctamente al símbolo utilizado la sintaxis descrita en el **Capítulo 12**. Si se está en presencia de un homónimo se debe incorporar un nuevo símbolo utilizando la sintaxis del **Capítulo 12** y agregarlo a la lista de símbolos.

Paso 1.7:

Nuevamente, por cada símbolo descripto, se deben “*buscar nuevos símbolos*” y Retornar al Paso 1.5.

NO

Paso 2.1:

En este camino de la heurística el punto de vista del “Deber ser” se debe obtener de las personas, ya que la literatura del macrosistema no es adecuada.

Incorporar los símbolos semillas a la lista de símbolos.

Para realizar las entrevistas en orden jerárquico se debe “*identificar la jerarquía organizacional*”. Para ello es necesario solicitar un organigrama o consultar acerca de los roles existentes para conocer dicho orden jerárquico. Tener en cuenta que existen organigramas informales que deben ser consultados, en ocasiones son informados y en otros casos es necesario identificarlos.

Paso 2.2:

Comenzar a “*entrevistar sujetos de mayor jerarquía*”. En este caso puede ser una única persona o varias. En estas entrevistas, cuando son directivos o gerentes, el punto de vista que se obtiene es casi siempre el “Deber ser”. En el caso de cargos jerárquicos intermedios (Jefes), se debe prestar atención ya que pueden fluctuar entre el “Deber ser” y el “Es”, pero el punto de vista predominante sigue siendo el “deber ser”. El

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 14 – Construcción del LEL por proximidad

ingeniero/a de requisitos debe tratar de intervenir lo menos posible, preguntar al usuario-cliente “*Cuénteme, ¿Ud. qué hace?*” y dejarlo explayarse.

Paso 2.3:

Grabar las entrevistas para luego *transcribirlas*.

Paso 2.4:

En este momento se cuenta con la lista de símbolos y las transcripciones del nivel jerárquico, las cuales son desde ahora, documentos del macrosistema.

Para comenzar, tomar un documento y completar los símbolos semilla. Para esto se deben *Seleccionar Símbolos*, solo algunos pocos.

Paso 2.5:

Recorrer todo el documento y *Describir*, incorporando el máximo de información posible.

Paso 2.6:

En cada símbolo descripto, *Buscar nuevos símbolos*, prestando atención a palabras o frases relevantes del macrosistema en el marco del Objetivo General del Sistema. Cada nuevo símbolo detectado debe ser incorporado a la lista de símbolos y repetir estos pasos hasta agotar el documento. Al pasar a otro documento se debe ir tomando la lista de símbolos para saber que símbolos completar y si aparecen nuevos, incorporarlos a la lista. Repetir todo el proceso en todos los documentos.

Estos símbolos descriptos no necesitan ser comparados, o sea no es necesario analizar si existen diferentes puntos de vista, ya que todos tienen el punto del “deber ser”. A pesar de que se espera que solo aparezca este punto de vista, no se debe descartar la aparición de algún punto de vista “es”, principalmente si se trabaja con personal jerárquico cercano a lo operativo (por ejemplo, jefes). En este caso, se debe incorporar la información al LEL como se describió en el **Capítulo 12**.

Si se detecta alguna incongruencia o inconsistencia, se debe retornar al UdeD.

Paso 2.7:

Una vez completo el nivel jerárquico se comienza a trabajar con el nivel inferior siguiente. Se deben “*entrevistar a todos los sujetos de un mismo nivel*”.

Paso 2.8:

Una vez realizadas las entrevistas se deben *transcribir*.

Paso 2.9:

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 14 – Construcción del LEL por proximidad

En este momento se debe trabajar como en el punto 1.6 del otro camino de la heurística.

En estos casos, es posible que aparezca con mayor fuerza diferencias entre los puntos de vista, por lo tanto se debe *Seleccionar los símbolos, Comparar e integrar o Describir*, según corresponda.

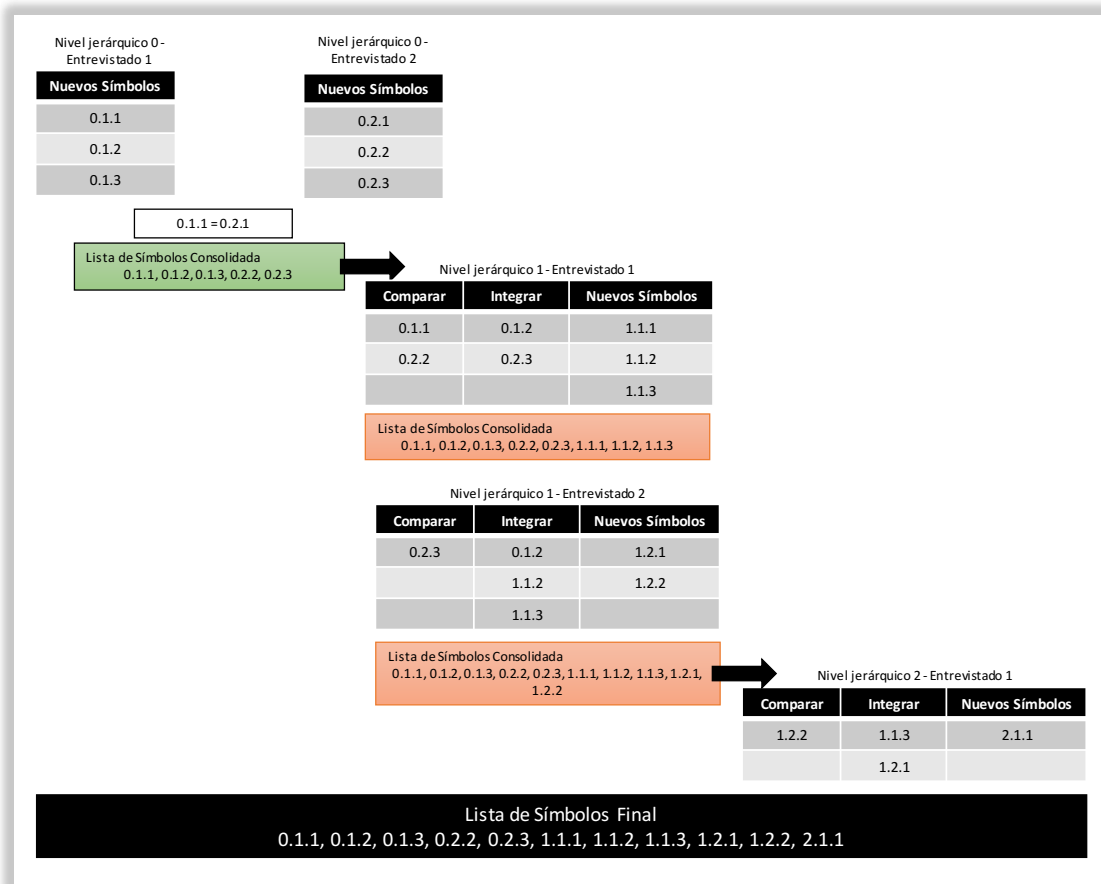


Figura 106 – Ejemplo de lista de símbolos del LEL por jerarquías organizacionales

En la Figura 106 se puede observar cómo se forma la lista de símbolos, incorporando símbolos luego de cada entrevista. También se van completando los símbolos existentes. Es de esperar que a medida que se completa el LEL, la cantidad de símbolos nuevos sea menor. En cada cambio de nivel se pueden encontrar puntos de vista, por tal motivo se debe *comparar* un nivel con el otro. En el ejemplo, aparecen puntos de vista para los símbolos 0.1.1, 0.2.2, 0.2.3 y 1.2.2. También se puede observar que se incorporó información durante las entrevistas en los diferentes niveles jerárquicos, específicamente en los símbolos 1.1.1, 1.1.2, 1.1.3, 1.2.1 y 1.2.2. En cada paso se consolida la lista de símbolos, incorporando los símbolos nuevos. Por cuestiones de legibilidad del gráfico, solo se muestra cómo trabajar con los símbolos ya que la forma de llevar la lista de símbolos se describió en la Figura 104.

Paso 2.10:

Nuevamente, por cada símbolo descripto, se deben “*buscar nuevos símbolos*” y Retornar al Paso 2.7.

En este momento, por ambos caminos, se tiene un LEL completo. Para consolidar las descripciones de diferentes ingeniero/as de requisitos, donde pueden quedar ocultas algunas jerarquías o puntos de vista, se debe realizar la última actividad que se refiere a “*Identificar Jerarquías Conceptuales, Estados y Puntos de Vista*”. Para ello, es necesario recorrer todo el LEL buscando algunas particularidades como sinónimos parciales para jerarquías, estados omitidos pero presentes en otro símbolo, etc. Este procedimiento actúa como una pre-verificación, donde el objetivo es identificar alguna de estas posibilidades y acceder rápidamente a los documentos o a las personas para rectificarlo en el LEL. En caso de detectar una jerarquía, modificar los símbolos correspondientes dejando claro qué tipo de jerarquía es y cuáles son sus elementos. Para incorporar jerarquías, estados y puntos de vistas utilizar la representación descrita en el **Capítulo 12**.

Es altamente probable que esta manera de construir el LEL aumente la cantidad de IE (ver **Capítulo 19**), específicamente información adelantada (IA), debido a que la incorporación de puntos de vista en el LEL es un disparador para pensar en ella.

Capítulo 15

LEL de Requisitos

Resumen

Se ha detectado que el uso del LEL en los EF, paradójicamente, en vez de mejorarlos les agrega ambigüedad. Esto se debe a que el LEL cubre las descripciones del UdeD actual, pero cuando es necesario describir un nuevo servicio del software o cuando se desea expresar una necesidad, esos términos no son suficientes. Así como los Escenarios evolucionan de Actuales a Futuros, también lo hace el léxico. En el presente Capítulo se describe el LEL de Requisitos (LEL_R) que contiene el léxico del UdeD futuro para asegurar la comprensión de los EF y de la ERS.

Evolución del vocabulario

Para describir los servicios del nuevo Sistema de Software suele ser necesario un léxico diferente que el utilizado hasta ese momento en el macrosistema [Kaplan10] [Kaplan13b]. Este vocabulario representa los cambios en el UdeD y están asociados exclusivamente al nuevo Sistema de Software; muestran la evolución natural que sufre el léxico durante la construcción del software. Curiosamente, esta evolución del vocabulario lleva más tiempo que la de los cambios en los procesos, los cuales evolucionan rápidamente en la etapa de la puesta en servicio del Sistema de Software. Se puede observar en la Figura 107 que los cambios en el léxico comienzan antes de la modificación de los procesos del negocio y finalizan después de haber puesto en producción el Sistema de Software. Este inicio anticipado se debe a que los clientes y usuarios comenzaron a pensar en el nuevo Sistema de Software mucho antes de decidir construirlo. Y termina después de haberlo puesto en producción ya que es cuando se conoce qué parte de este léxico fue realmente aceptado por los usuarios.



Figura 107 – Evolución del léxico del dominio durante la construcción del software

El LEL, creado tempranamente en el Proceso de Requisitos y utilizado para describir las situaciones observables del UdeD, no es apto para describir todas las situaciones que ocurrirán cuando se utilice el nuevo Sistema de Software, ya que la forma de abordar estas situaciones requiere en muchos casos, modificar los términos existentes o definir nuevos términos. Por lo tanto, los documentos posteriores a los EA no estarán totalmente cubiertos por el LEL. Es paradójico que el glosario que fue creado para reducir la ambigüedad de los modelos construidos se convierta en una fuente de ambigüedad. Para que esto no suceda, es necesario que el LEL sea adaptado al UdeD futuro para asegurar la consistencia de todos los documentos, principalmente los que contienen los requisitos del software. Se debe remarcar que el UdeD tampoco preserva en forma inmutable el

vocabulario registrado al inicio del Proceso de Requisitos (ver Figura 108). Los clientes y usuarios también modifican su vocabulario como consecuencia de participar en la planificación del contexto en el cual se desarrollará el futuro Sistema de Software. A estos cambios en el vocabulario se los denomina en la presente tesis, evolución (ver **Capítulo 3**).

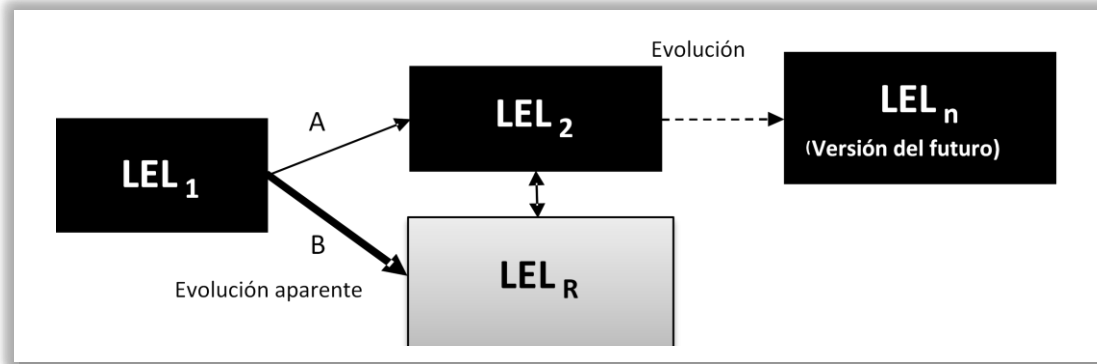


Figura 108 – Evolución del LEL

La Figura 108 muestra en el camino A, que existe un LEL relacionado con la evolución del UdeD. Este glosario registra las modificaciones del vocabulario muy lentamente y genera un documento con una vigencia mucho más prolongada. Registrar esta evolución, no ofrece ninguna utilidad al Proceso de Requisitos. Por su parte, el camino B es aquel relacionado con el Proceso de Requisitos en sí mismo y que se denomina *evolución aparente* porque solo queda registrada en los documentos de desarrollo. El léxico que permite representar la evolución aparente se lo denomina en la presente tesis LEL de Requisitos (LEL_R) [Kaplan13a]. Este glosario es significativamente diferente del LEL inicial ya que incorpora todos los términos necesarios para desambiguar los EF y la ERS.

Estrategias de Construcción del LEL_R

Para construir la heurística del LEL_R se analizaron dos aspectos fundamentales. Por un lado, la *estructura* que debe tener el documento que contenga el nuevo léxico y por el otro, el *enfoque de construcción*.

Con respecto a la *estructura* del documento se analizaron dos opciones. La primera opción es tener un solo glosario para todo el Proceso de Requisitos, mientras que la segunda opción, es generar un documento independiente para el nuevo glosario. En el caso de tener un solo documento, la primera opción, implica incorporarle al LEL los términos del UdeD futuro. Esta opción fue descartada debido a que se pierde el objetivo fundamental del LEL que es representar los términos que efectivamente se usan en el

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 15 – LEL de Requisitos

UdeD. Además, la inclusión de los nuevos términos agrega complejidad, dificultando la lectura del glosario. Este último problema se debe a la necesidad de utilizar una gran cantidad de homónimos para registrar aquellos términos del UdeD actual que se siguen utilizar en el UdeD futuro, pero tienen un significado propio en el UdeD futuro. En el caso de la segunda opción, crear un nuevo glosario en un documento independiente, tiene la desventaja relacionada con tener dos glosarios dentro del mismo proceso, pudiendo generar algún tipo de confusión. En la presente tesis se optó por la segunda opción, ya que el riesgo de confusión es minimizado al tener claramente identificados los documentos del UdeD actual y los del UdeD futuro. Algunas características que justifican la separación de ambos glosarios:

- El LEL describe el vocabulario del UdeD efectivamente utilizado. El LEL_R describe el vocabulario de los documentos que se han generado.
- El LEL representa un vocabulario vigente. El LEL_R es parcialmente artificial.
- El LEL se genera muy tempranamente en el Proceso de Requisitos. El LEL_R se crea durante la planificación del UdeD futuro.
- El LEL es un glosario descriptivo mientras que el LEL_R es prescriptivo. En el primer caso el glosario describe el léxico del macrosistema. En el caso del LEL_R dice cómo se deben llamar a las cosas nuevas.

Es esperable que, con el transcurso del tiempo, gran parte de las novedades del LEL_R pierdan su condición de ser artificiales y se incorporen al uso habitual en el UdeD.

Una vez analizada la estructura del glosario, es necesario considerar cuál es el mejor *enfoque de construcción*. A diferencia del LEL que es la primera actividad del proceso, la construcción del LEL_R puede ser realizada simultáneamente con la construcción de los EF o posteriormente, cuando los EF están completos. Parte de esta decisión consiste en lograr que la concentración del ingeniero/a de requisitos sea realmente efectiva.

La **construcción en paralelo** consiste en ir identificando y registrando en el LEL_R los términos nuevos o cuya definición se debe modificar, en el mismo momento de describir los EF. La ventaja de este enfoque es que en el momento de construir el glosario se cuenta con la información directa del UdeD. Pero este enfoque genera una sobrecarga de atención y concentración para el ingeniero/a de requisitos ya que se deben construir dos modelos en forma simultánea, que, a pesar de ser complementarios, tienen objetivos diferentes. Esta dispersión de la atención puede disminuir la calidad de los dos modelos.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 15 – LEL de Requisitos

Además, construirlos en paralelo dificulta cumplir con el principio de circularidad, donde se maximiza el uso de otros símbolos en las descripciones, ya que los EF que contendrán esos otros símbolos aún no se han descrito.

La **construcción posterior o tardía** se refiere a construir el LEL_R una vez finalizado el proceso de construcción de los EF. En esta forma de trabajar, toda la atención del ingeniero/a de requisitos se concentra en el vocabulario necesario para minimizar la ambigüedad de los EF y en construir un glosario de buena calidad. Este enfoque permite una visión en dos etapas con mecanismos cognitivos distintos. En la primera etapa, durante la construcción de los Escenarios, la atención está puesta en describir y definir el nuevo Sistema de Software y su contexto. Mientras que, en la segunda etapa, en la construcción del LEL_R, se está pensando sólo en el vocabulario y no en los procesos.

Tomando en cuenta todo lo antes dicho, la presente tesis adhiere a construir un documento independiente para el LEL_R utilizando un enfoque mixto, primero en paralelo y luego, en batch. Este enfoque mixto intenta aprovechar las ventajas de cada enfoque minimizando sus inconvenientes. En ese sentido, si durante la construcción de los EF sólo se registran los términos que deben ser incorporados al LEL_R o modificados, pero se posterga su descripción detallada, el grado de perturbación de la construcción de los EF es mínimo. En la Figura 109 se puede observar una síntesis de ambas estrategias.

Como puede observarse en la Figura 109, la construcción tardía se realiza entre la actividad Verificar y Validar de la construcción de los EF. Esta decisión cumple con el objetivo de reducir la ambigüedad de los EF y asegurar la validación de los mismos, ya que ésta no sería igual sin un glosario. Es posible que en la validación sea necesario retocar el glosario, pero en los casos estudiados estos cambios resultaron muy escasos.

Dos menciones aclaratorias. La primera, para analizar y definir la heurística que sigue se han tenido en cuenta todos los cambios y agregados al LEL presentados en el **Capítulo 12** y la segunda, todos los hipervínculos existentes en los EF y en la ERS son, a partir de este momento, exclusivas referencias al LEL_R.

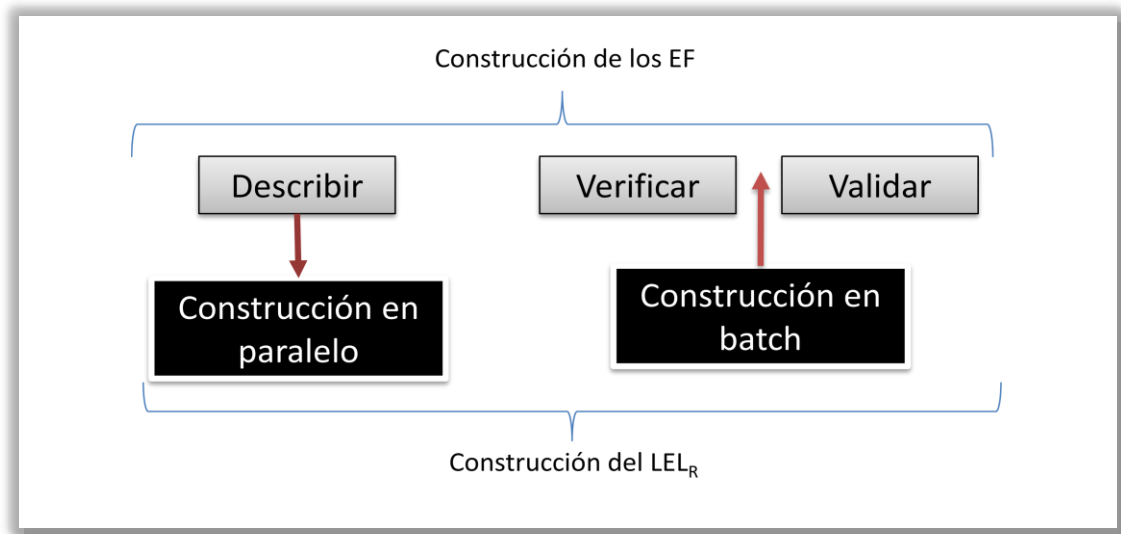


Figura 109 – Enfoques de construcción del LEL_R

Heurística de Construcción del LEL_R

Antes de comenzar con la heurística, es importante mencionar que para un mayor aprovechamiento del proceso y, con el objetivo de retomar la información léxica obtenida del UdeD futuro, es necesario, siempre que sea posible, grabar todas las entrevistas, filmar las observaciones, etc. que se realizaron durante la negociación de los EF y luego, transcribirlas.

A continuación, se describen los pasos que conforman la heurística de construcción del LEL_R.

CONSTRUCCION EN PARALELO

Esta parte de la heurística se realiza en paralelo a la descripción de los EF.

Paso 1:

Copiar LEL al LEL_R: copiar toda la información del LEL al LEL_R. Desde ahora en adelante solo se utiliza el LEL_R.

Paso 2:

Marcar hipervínculos. Al describir los EF aparecen palabras y frases relevantes del UdeD futuro. Cada una de ellas se deben marcar como un hipervínculo al LEL_R. Se debe utilizar la misma forma de marcación del LEL.

Este léxico relevante del contexto futuro puede aparecer en cualquier componente del Escenario, pero se espera que una mayor asiduidad en el componente Episodios.

Paso 3:

Incorporar nuevos símbolos. Por cada hipervínculo se controla su existencia en el glosario. Si no existe, se agrega un símbolo al LEL_R indicando sus componentes (Nombre, Noción e Impacto). El Nombre es igual al texto del hipervínculo y los componentes Noción e Impacto deben estar vacíos.

Antes de dar un alta asegurarse de eliminar los duplicados. O sea, en un Escenario puede aparecer varias veces mencionado el mismo símbolo, en este caso poner solo uno.

Paso 4 (optativo):

Incorporar información a un símbolo nuevo. Esta es una decisión del ingeniero/a de requisitos, pero se sugiere que, si desea incorporar información al símbolo, solo sea en la Noción. Esto responde a distraer lo menos posible su atención del objetivo principal, construir los EF.

CONSTRUCCION EN BATCH

Esta parte de la heurística se realiza cuando todos los EF están completos y verificados.

Paso 5:

Recorrer cada EF. Cada vez que se encuentre un hipervínculo en el escenario se debe buscar el símbolo en el LEL_R para analizar su definición. La búsqueda debe realizarse teniendo en cuenta todos los nombres posibles del símbolo (sinónimos).

Paso 6:

Este paso se realiza cuando se encuentra el símbolo.

Agregar información. Si el símbolo está completo y la noción del símbolo aplica al contexto del EF, solo se debe analizar si es necesario agregar alguna información particular.

Completar símbolo. Si el símbolo está vacío, completar el símbolo.

Si las nociones son distintas pero el nombre es el mismo, se trata de otro símbolo.

Crear homónimo. Si el nombre del símbolo es correcto pero la definición del símbolo no es consistente con lo indicado en el EF, se debe crear un homónimo.

Paso 7:

Este paso se realiza cuando no se encuentra el símbolo.

Incorporar sinónimo. Este caso es más difícil de identificar, pero se debe prestar atención a la existencia de dos o más símbolos nuevos cuya noción sea coincidente o muy similar.

Paso 8:

Detectar nuevos símbolos: Puede suceder que al revisar un EF se detecten otros símbolos que no fueron vistos con anterioridad. En este caso se agrega el hipervínculo y se repiten los pasos anteriores.

Paso 9:

Eliminar símbolos no referenciados: Todo símbolo que no fue referenciado desde algún EF debe ser eliminado.

Paso 10:

Verificar LEL_R: esta actividad es para analizar que se cumpla la circularidad cuando sea posible, que se unifiquen sinónimos, etc. (ver Anexo A)

Esta heurística, específicamente por su construcción en batch, funciona como un mecanismo de verificación de la completitud de los EF.

A continuación, se presentan algunas recomendaciones para describir el LEL_R:

- Utilizar los *patrones sintácticos* para completar la noción y el impacto (ver Evolución de los Patrones Sintácticos en el **Capítulo 12**).
- Se debe estar siempre atentos a la posibilidad de la aparición de *sinónimos*. Si bien es difícil de detectar, principalmente cuando trabajan varios ingeniero/as de requisitos, es importante saber que puede suceder.
- Para detectar nuevos símbolos o Escenarios faltantes analizar los empotramientos de los términos propios del UdeD futuro. Para visibilizarlos activar la Vista Empotramiento (ver Vista Empotramiento en el **Capítulo 12**).
- El LEL_R debe cumplir con el *Principio de Vocabulario Mínimo* y el *Principio de Circularidad*. Este último, como ya se mencionó, cuando sea posible.
- Aplicar la identificación de *jerarquías* post construcción. Se debe prestar una especial atención a no crear jerarquías duplicadas. Si no existe se debe crear la jerarquía y analizar la completitud de la misma. También puede suceder que una jerarquía existente en el LEL se disperse en este nuevo contexto y obligue a revisar los componentes noción de todos sus elementos para eliminar rastros de la misma (ejemplo, si se encuentra un “es un” o “es parte de” se debe reemplazar por otra

expresión para no confundir al lector) (ver Jerarquías Conceptuales en el **Capítulo 12**).

- Cuando un símbolo del LEL (ver Figura 110) tiene *puntos de vista*, sólo deben quedar el o los puntos de vista que fueron aceptados. Se debe recordar que esta decisión fue tomada en el pasaje de los EA a EF.

Empleado de Farmacia (S)
Noción:
• Trabaja dentro del sector de farmacia.
• Responde a la autoridad del [farmacéutico](#).
Impacto:
• Ordena el [almacén](#).
• [Dispensa medicamentos](#).
• Puede [recibir pedidos](#).
• Realiza [control de stock mínimo](#) Lo debe realizar el [farmacéutico](#).

Figura 110– Símbolo del LEL con Punto de Vista

Este paso permite verificar si no se tomó una decisión oportunamente, si no se hizo pudo ser por un olvido o porque no había suficiente información para hacerlo. Este paso en la construcción del LEL_R permite detectar una omisión de gran importancia en los Escenarios Futuros.

- Cuando se analiza un símbolo *Estado* se debe considerar también el símbolo al que aplica para asegurar la pertinencia de ambos en el UdeD futuro. Esto permite detectar omisiones en los EF. Se deben recorrer los EF donde participa el símbolo al que aplica alguno de los estados.
- Existe una diferencia conceptual muy importante entre lo que sucede con los verbos en el LEL y en el LEL_R. El LEL es el primer modelo que se construye y su principal objetivo es comprender el vocabulario del macrosistema. En este contexto, es necesario incluir todos los verbos que aparecen, de lo contrario no se alcanzan esos objetivos. En el caso del LEL_R no existen esos propósitos. El objetivo del LEL_R es asegurar la comprensión de los EF y de la ERS por parte de los clientes y usuarios. No se espera que este léxico sea aprendido porque no se sabe cuál es la persistencia que tendrá una vez que el Sistema de Software esté en ejecución. Por lo tanto, si el Principio de Circularidad se ve afectado, no es relevante en este contexto. Cabe destacar que no incluirlos no afecta la comprensión del UdeD futuro, ya que la mayoría de los verbos del LEL_R serán representados directamente en Escenarios. Por lo tanto, los verbos del LEL_R serán

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 15 – LEL de Requisitos

sólo aquellos que no son Escenarios. Los verbos que están en los títulos de los Escenarios y sub-Escenarios no deben ir al LEL_R.

Cabe recordar, que sólo pensar en la construcción de un sistema informático genera un cambio en la forma de observar el contexto, transformando los elementos existentes o incorporando nuevos. Por ejemplo:

En el UdeD actual donde no existe un software preexistente, el Cliente corresponde a una persona. En este caso el tipo de símbolo es sujeto. En el UdeD futuro, cuando exista un sistema informático, aparecerán dos símbolos Cliente: uno para representar a la persona (sujeto) y otro para los datos (objeto).

Como ya se mencionó, el LEL_R es un glosario para la etapa de Proyección del UdeD futuro que puede ser adjuntado a cualquier ERS. Si para especificar se decide utilizar el estándar IEEE 830-1998 [IEEE830], este glosario cumple con todos los requisitos establecidos por la norma.

Capítulo 16

Cambios y Agregados a los Escenarios

Resumen

En el presente Capítulo se presentan los cambios y agregados a los Escenarios, los cuales tienen dos orígenes. Por un lado, existe una estrecha relación con el LEL, lo que ha generado que los cambios y agregados al glosario deban ser incorporados en los Escenarios Actuales. Por el otro lado, se atienden las particularidades propias de los Escenarios, las cuales han sido descritas en el **Capítulo 11**.

Síntesis de los cambios y agregados

Lo que se presenta a continuación tiene su origen en el estudio empírico presentado en el **Capítulo 11**. Se puede observar en el centro de la Figura 111 los temas centrales a tratar, mientras que las flechas indican a qué modelo o actividad aplican. A excepción de la Actividad Derivar que es exclusiva de los EA, el resto aplica a ambos tipos de Escenarios, Actuales y Futuros.

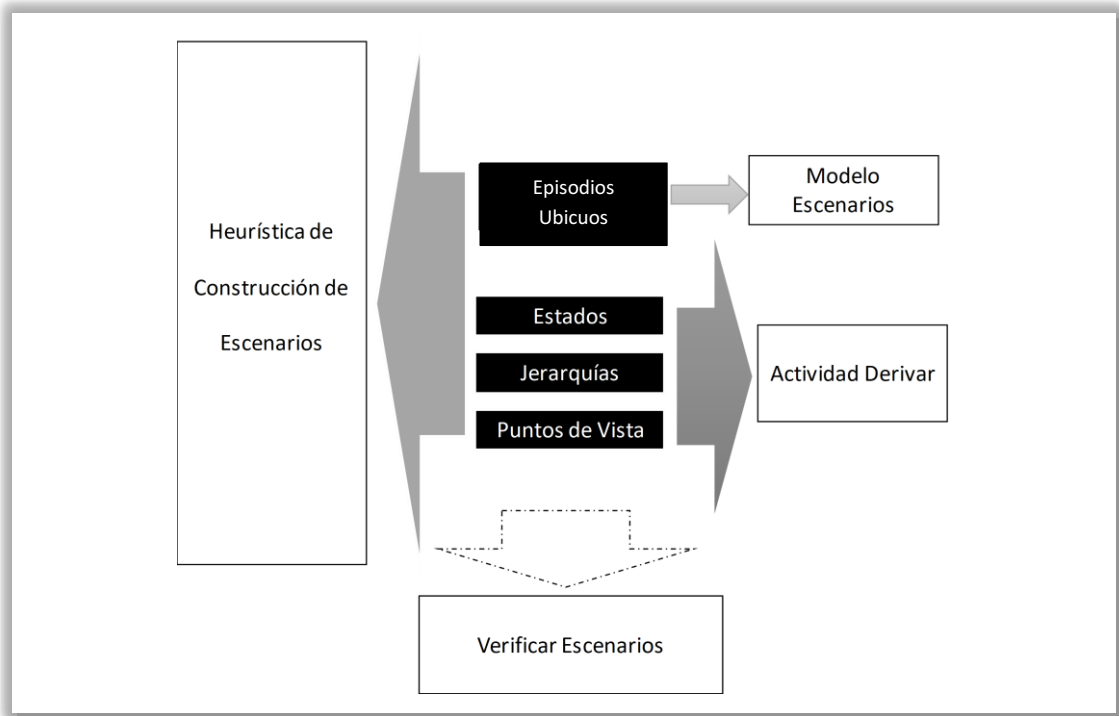


Figura 111 – Cambios y agregados a los Escenarios

En la Figura 111 se puede observar que la flecha hacia la actividad Verificar Escenarios es punteada, indicando que todas las modificaciones a los formularios de inspección no serán tratadas en su totalidad, en la presente tesis, sólo se describe en el **Capítulo 17** algunos ejemplos de formularios con el objetivo de mostrar la naturaleza de los cambios.

Nueva Actividad Derivar

Como se describió en el **Capítulo 11**, la decisión de construir una nueva heurística de Derivación fue el resultado del trabajo de análisis sobre el LEL y los Escenarios. Cabe mencionar, que en un primer momento se consideró la adaptación de la heurística existente, pero debido al carácter de los problemas encontrados se decidió reemplazarla

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 16 – Cambios y Agregados a los Escenarios

para incorporar todos los cambios necesarios. Por razones de espacio y para una mejor comprensión del tema, la nueva heurística de derivación se describe en el **Capítulo 18**.

Vinculación con el LEL

Como se ha mencionado reiteradas veces en la presente tesis, los Escenarios Actuales tienen un vínculo muy estrecho con el LEL. Parte de la fuerza de este vínculo se sustenta en la necesidad de mantener la facilidad de lectura de los Escenarios, haciendo al LEL hospedador de gran parte del detalle relacionado con cada símbolo. Los Escenarios utilizan el LEL para reducir la ambigüedad y mejorar la comprensión de las situaciones descritas. La relación entre ambos modelos se efectiviza con hipervínculos que van desde los Escenarios al LEL. A través de estos se puede acceder a cada definición del LEL y corroborar su significado en el contexto. Realizar o evitar revisar las definiciones del LEL es una decisión de quien describe o lee un Escenario. Cuando no se verifica cada referencia al LEL, existe el riesgo de dejar inconsistentes los Escenarios. Por ejemplo, si se tienen dos símbolos, uno es un sustantivo (“Devolución”) clasificado en el LEL como objeto y otro es un verbo (“Devolver”), es muy común encontrar expresiones, en cualquier componente del Escenario, que se los utilice indistintamente, vinculando al símbolo objeto cuando lo debe hacer al verbo y viceversa. Este problema, que se corrige cambiando el hipervínculo al símbolo correcto, no suele ser tan fácil de identificar.

Se debe recordar que cuando se trata de los EF la referencia al glosario es al LEL_R (ver **Capítulo 15**).

Para garantizar la consistencia entre ambos modelos, se propone utilizar la **Vista Clasificación** del LEL descrita en el **Capítulo 12**, que ayuda al ingeniero/a de requisitos a visibilizar algunos datos del LEL y de esta manera, facilitar la detección de algunos errores. Así como la Vista Clasificación mejora la inclusión de los símbolos del LEL, también mejora la inclusión de las jerarquías conceptuales y de los estados. Las jerarquías conceptuales del contexto tienen la característica de pertenecer a un mismo conjunto de símbolos y por lo tanto, introducen un riesgo importante de ser utilizadas incorrectamente. La Vista Clasificación las visibiliza facilitando el análisis preliminar de cada situación. Por ejemplo, permite detectar el uso incorrecto de un genérico o de una especialización, mejorando la precisión de la descripción. Este último punto que se relaciona con el nivel de detalle de los Escenarios, es de suma relevancia principalmente en los EF, los cuales contienen los requisitos del software y donde a mayor abstracción en la descripción de

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 16 – Cambios y Agregados a los Escenarios

los episodios, mayor esfuerzo para definir los requisitos asociados. Al utilizar un genérico en lugar de una especialización, se pueden asignar tareas a personas equivocadas, autorizar accesos a quienes no corresponde, llevar a cabo procesos con recursos incorrectos, etc. Al utilizar una especialización en vez de un genérico, se deja afuera al resto de las especializaciones, no permitiendo el acceso a quienes deben tenerlo, dejando tareas sin responsables, indicando que los procesos se realizan con menos recursos de los realmente necesarios, etc.

Para ejemplificar los beneficios de la Vista Clasificación se utiliza la taxonomía de la Figura 112, ya utilizada previamente en otro ejemplo.

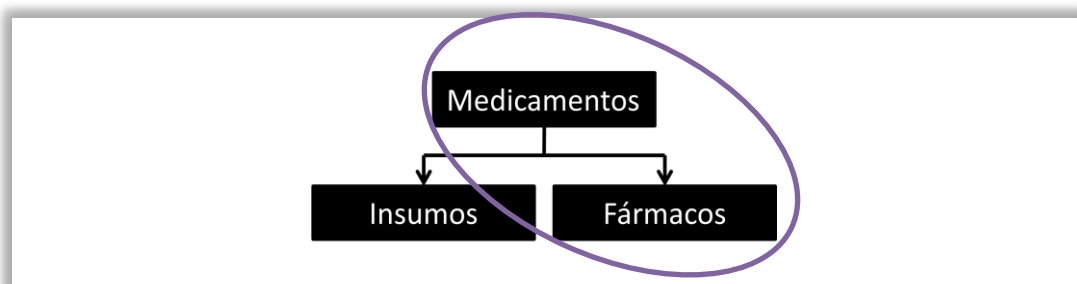


Figura 112 – Ejemplo de Jerarquía del Contexto

En el Escenario de la Figura 113 se puede observar un uso inapropiado del genérico en el componente Título, Objetivo y en la Precondición, donde se hace referencia a “medicamentos” como sinónimo parcial de “fármaco”, al igual que en el último episodio.

Título: DISPENSAR MEDICAMENTOS

Objetivo: entregar a un [servicio]_s personalmente los [medicamentos]_o ^{J1/2} solicitados

Contexto:

Ubicación Geográfica: [Sector Farmacia]_o

Ubicación Temporal:

Precondición: El [medicamentos]_o ^{J1/2} fue pedido por el [servicio]_s en cuestión.

Recursos: [fármaco]_o ^{J1.2} [historia clínica]_o [pedido de medicamentos]_o [solicitud de compra de medicamentos]_o

Actores: [empleado de farmacia]_s [servicio]_s

Episodios:

- El [empleado de farmacia]_s revisa el [pedido de medicamentos]_o solicitado por un [servicio]_s
- Busca en los estantes los [fármaco]_o ^{J1.2} solicitados
- Si hay stock del [fármaco]_o ^{J1.2} entonces retira del estante
- Si hay stock del [fármaco]_o ^{J1.2} entonces deja registro en la [historia clínica]_o lo que entregó al [servicio]_s
- Si no hay stock del [fármaco]_o ^{J1.2} entonces genera una [solicitud de compra de medicamentos]_o
- Si no hay stock entonces marca en la [historia clínica]_o que queda [pendiente]_{E,2} de entrega
- Entrega los [medicamentos]_o ^{J1/2} al [servicio]_s

Excepciones:

Figura 113 – Ejemplo Vista Clasificación para detectar errores de vinculación en el LEL

Las definiciones que utilizan el genérico dan a entender que la farmacia puede entregar a un servicio insumos y fármacos, algo que podría generar complicaciones muy serias en

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 16 – Cambios y Agregados a los Escenarios

el contexto, llegando a sumarios administrativos para aquellas personas que lo permitan, ya que la farmacia no está autorizada a entregar insumos. Puede observarse que la solución es simple, sólo se debe cambiar “medicamentos” por “fármacos”, pero no es tan fácil de identificar el error. La precondition debería quedar de la siguiente manera:

El [fármaco]_o¹¹ fue pedido por el [servicio]_s en cuestión.

Las jerarquías conceptuales también permiten verificar la completitud de los Escenarios ya que, la detección de un símbolo que pertenece a una jerarquía invita a un análisis del resto de la jerarquía, pudiendo detectar episodios omitidos e inclusive, detectar situaciones faltantes, mejorando la definición de los límites del contexto que se está modelando.

Otro aspecto a considerar, referido al vínculo entre el LEL y los Escenarios, es el tratamiento de los **estados**. Como se describió en el **Capítulo 11**, existen diferentes instancias en el Proceso de Requisitos en los cuales se los debe analizar. La primera es en la Derivación desde el LEL a los EA. Pero, además, se les debe prestar una atención particular durante toda la construcción de los Escenarios, tanto del UdeD actual como del futuro.

En el **Capítulo 11** se han realizado una serie de observaciones acerca del uso de los estados en los Escenarios. En la Tabla 28 se recuerdan dichas observaciones en la primera columna y en la segunda columna, se menciona como resolverlas.

Observaciones del estudio empírico de los Escenarios	Cambios y Agregados
Se debe cuidar que la atención del ingeniero/a de requisitos se concentre en la situación que describe el Escenario, procurando que se distraiga lo menos posible.	Se propone enfatizar en las heurísticas esta situación.
Cada vez que aparece un estado en un Escenario, se debe analizar todo el símbolo estado para saber con qué otros estados se relaciona. De esta manera se pueden detectar episodios y restricciones omitidas.	Se propone la Vista Clasificación e incluir en la heurística de construcción como tratar esta situación.
Un recurso o un actor puede comenzar en un Escenario en un estado particular o dominante.	Se propone incorporar restricciones también a los actores. Este dato debería incluirse en las restricciones de los actores y de los recursos.

Tabla 28 – Relación entre el estudio empírico de los Escenarios y las soluciones

Se puede observar en la Figura 114 como la Vista Clasificación permite identificar rápidamente la presencia de los estados para un análisis inmediato.

ALMACENAR INSUMOS

Objetivo: almacenar los [\[insumos\(1\)\]](#) según su estado

Contexto:

Ubicación Geográfica: depósito que corresponda

Ubicación Temporal:

Precondición: Deben existir [\[insumos\(1\)\]](#) sin almacenar

Recursos: [\[insumos\(1\)\]](#)_o, [\[depósito de cuarentena\]](#)_o, [\[depósito de aprobados\]](#)_o, [\[depósito de rechazados\]](#)_o, [\[depósito de devoluciones\]](#)_o

Actores: [\[Persona de producción\]](#)_s, [\[técnico\]](#)_s

Episodios:

- Si el [\[insumo\(1\)\]](#)_o está [\[pendiente\]](#)_{E3} entonces la [\[persona de producción\]](#)_s determina un sector del [\[depósito de cuarentena\]](#)_o para almacenarlos
- Si el [\[insumo\(1\)\]](#)_o está [\[aprobado\]](#)_{E3} entonces el [\[técnico\]](#)_s determina el sector del [\[depósito de aprobados\]](#)_o para almacenarlos
- Si el [\[insumo\(1\)\]](#)_o está [\[rechazado\]](#)_{E3} entonces el [\[técnico\]](#)_s determina el sector del [\[depósito de rechazados\]](#)_o para almacenarlos
- Si es una [\[devolución\]](#)_o entonces el [\[técnico\]](#)_s determina el sector del [\[depósito de devolución\]](#)_o para almacenarlos
- [Solicita ayuda para ubicar los [\[insumos\(1\)\]](#)_o en su lugar]

Excepciones:

Figura 114 – Ejemplo de la Vista Clasificación para identificar Estados (Caso SCC)

El uso de la Vista Clasificación puede provocar una **retroalimentación al LEL**. Esto sucede porque al analizar los símbolos utilizados en los Escenarios se pueden detectar problemas como errores, omisiones o discrepancias existentes en el LEL. También, es más fácil detectar **homónimos** que no fueron vistos oportunamente. Por ejemplo, en un macrosistema donde existe un símbolo “área de planeamiento” definido como objeto y el ingeniero/a de requisitos lo utiliza de la siguiente manera:

El [\[director\]](#) entrega la [\[Long list\]](#) con las [\[iniciativas de inversión\]](#) al [\[área de planeamiento\]](#)

Ahora bien, si al mismo ejemplo se le activa la Vista Clasificación queda expuesto el error ya que “Área de planeamiento” es un objeto y está utilizado como un sujeto.

El [\[director\]](#)_s entrega la [\[Long list\]](#)_o con las [\[iniciativas de inversión\]](#)_o al [\[área de planeamiento\]](#)_o

En este caso, el ingeniero/a de requisitos tiene casi la certeza de haber omitido un homónimo, el cual debe ser corroborado en el UdeD. De confirmarse la omisión, debe ser incorporado al LEL y se deben controlar todos sus usos en los Escenarios.

Con la Vista Clasificación, es posible identificar errores involuntarios muy difíciles de detectar y que comprometen seriamente la comprensión del Escenario, sin salir de la

edición de los Escenarios y con una mínima distracción.

Puntos de Vista

Durante la descripción de cada EA puede aparecer nueva información sobre un punto de vista no considerado o tratado en forma incompleta durante la construcción del LEL. En este caso, se deben incorporar al LEL y luego, rastrear como aplican en los EA. Luego, durante la Proyección del UdeD futuro, estos puntos de vista del UdeD actual se deben analizar en función del Objetivo General del Sistema y del objetivo de cada Escenario Futuro. En este momento puede suceder que el punto de vista se resuelva o se mantenga en el UdeD futuro. Obviamente, si el punto de vista corresponde a los procesos del negocio involucrados con el nuevo Sistema de Software, se espera que en algún momento se resuelvan.

A efectos de simplificar la explicación, se utiliza el mismo ejemplo que para el LEL (ver Figura 85), donde primero se detallan diferentes formas de describir un mismo impacto con puntos de vista. Sólo a efectos de mejorar el ejemplo para los Escenarios se han eliminado aquellos impactos que deberían haberse corregido en la validación del LEL. Se puede observar en la Figura 115, en los ítems 2, 3 y 4, la presencia del punto de vista “Es”, pero también aparece el “Deber ser”. En los ítems 1 y 5 sólo se percibe un punto de vista y, por lo tanto, se asume que coincide el “Es” con el “Deber ser”. Pero en estos casos, se sugiere revisar esta percepción.

- 1) La materia prima es solicitada personalmente, por mail o telefónicamente.
- 2) La materia prima es solicitada telefónicamente. Debe ser solicitada personalmente o por mail
- 3) La materia prima es solicitada telefónicamente, pero debe ser solicitada personalmente o por mail.
- 4) La materia prima debe ser solicitada personalmente o por mail, pero se hace también telefónicamente.
- 5) La materia prima se solicita personalmente, por mail y telefónicamente. Esta última se realiza cuando hay una necesidad operativa.

Figura 115 – Ejemplos de impactos con Puntos de Vista

Ante la presencia de un punto de vista en el LEL se debe trasladar a los EA tomando en cuenta como está expresado el punto de vista. En relación a esto se determina si se traslada como:

- Episodio Opcional.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 16 – Cambios y Agregados a los Escenarios

- Episodio Condicional.
- Excepción.
- Restricción.

En el primer caso, trasladar como episodio opcional, se da cuando ambos puntos de vista están claramente definidos en el LEL, como es el caso de los puntos 2, 3 y 4 de la Figura 115. En estos casos siempre poner el episodio simple con el “Deber ser” y luego, para trasladar el “Es” utilizar un episodio opcional:

El jefe de producción solicita personalmente o por mail la MP → “Deber ser”
[El jefe de producción solicita telefónicamente la MP] → “Es”

Se traslada como episodio condicional cuando existen alternativas, por ejemplo, el punto 5) de la Figura 115, donde no es posible identificar el “Deber ser”. Aquí no hay conflicto sino alternativas que se resuelven con un condicional.

Si no existe una necesidad operativa entonces el jefe de producción solicita la materia prima personalmente o por mail.
Si existe una necesidad operativa entonces el jefe de producción solicita la materia prima telefónicamente.

Se traslada como excepción, cuando el “Deber ser” tiene un carácter de obligatoriedad y su ejecución es disruptiva.

• El jefe de producción analiza las necesidades de producción
• #[envía el mail a stock]
• [solicita personalmente]#
Excepción:
Cuando el jefe de producción no está en la planta, solicita la materia prima telefónicamente.

Por último, cuando solo se menciona el “Deber ser” y no existe ningún conflicto, el punto de vista puede ser tratado en el Escenario como una restricción:

El jefe de producción solicita materia prima
Restricción: se debe solicitar personalmente o por mail.

En el caso del punto de vista “Próximo”, aparece en el LEL como “será” o similar y se traslada a los EA incorporando condicionales.

Una vez completados los EA, se deben trasladar adecuadamente a los EF. En un primer momento, se deben trasladar tal como están en los EA, y luego de ser analizados y de ser

necesario, consultar con los clientes y usuarios durante las entrevistas para tomar las decisiones pertinentes en cada caso, las cuales pueden incluir:

1. Incorporadas ambas conductas.
2. Incorporar sólo el “Deber ser”.
3. Incorporar sólo el “Es”.

Cuando se decide dejar ambas conductas significa que el conflicto es aparente, por lo tanto, se mantiene el episodio condicional.

En el segundo y tercer caso, se genera un episodio simple con el punto de vista correspondiente. En todos los casos, cuando se trata de una excepción se la debe mantener como tal, pero se debe asegurar que sigue siendo una excepción en el UdeD futuro.

Cuando en el UdeD futuro resulta necesario eliminar una conducta irregular, se debe incorporar una restricción que indique dicha prohibición, por ejemplo, que se debe impedir que los pedidos al stock se realicen de manera telefónica.

Con respecto al punto de vista “*Próximo*” que aparece en los EA como “Será” tiene la particularidad de poder sobrevivir o no el período de uso del sistema, algo que debe ser evaluado antes de su incorporación a los EF. Se debe trasladar a los EA como una restricción que deberá ser tenida en cuenta al describir los EF, donde se decidirá junto al cliente-usuario si es incorporada como un episodio, un Escenario o si se rechaza.

Se debe tener en cuenta que en muchas ocasiones la incorporación de un sistema informático cumple el rol de forzar ciertas conductas y los conflictos que aparecen se deben resolver.

Importancia de las Jerarquías Conceptuales y los Puntos de Vista en los Requisitos del Software

El objetivo de esta sección es mostrar cómo influye el análisis de los puntos de vistas y de las jerarquías conceptuales en los requisitos del software. Puede observarse en lo que sigue, que la omisión en el proceso de estos elementos puede afectar considerablemente la calidad de los requisitos de software construidos. Cuando más temprana es su detección y análisis, mejor tratamiento de estos temas se puede lograr.

Se puede observar en la Figura 116 el recorrido que realiza un punto de vista desde su registro en el LEL, hasta que se incorpora en un EF. Se describen tres decisiones que se pueden tomar al construir el EF dependiendo de cómo se resolvió el punto de vista en el EF. En el camino 1) el punto de vista se trasladó del EA al EF sin cambios, arrastrando la

ambigüedad a los requisitos. En el camino 2) se decide en el EF un tratamiento claro de la obligatoriedad de los datos, manteniendo el “Deber ser” y “es” existentes. En el camino 3) se decide la obligatoriedad de todos los datos, convirtiendo el “Es” en “Deber ser”.

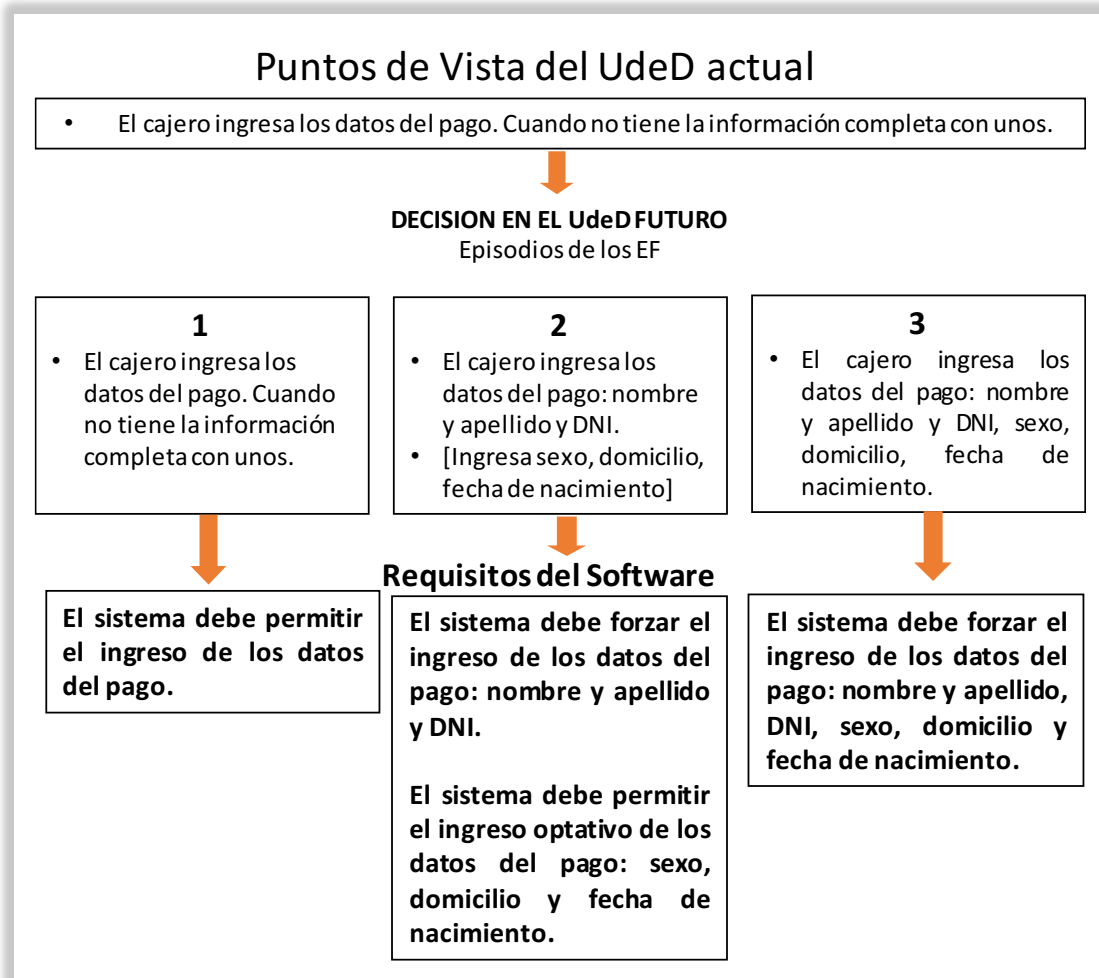


Figura 116 – Repercusión de los Puntos de Vista en los requisitos del software

El resultado más comprometido para los requisitos es el camino 1) ya que mantiene la ambigüedad y la falta de precisión resultando en requisitos de muy baja calidad. En este caso, el encargado del diseño no podrá continuar y deberá solicitar precisiones. Los casos 2) y 3) expresan una decisión acerca de cómo se espera trabajar según la operatividad deseada. En cualquiera de estos dos casos, se puede observar como un punto de vista detectado tempranamente, permitió realizar un análisis más profundo en el momento adecuado, o sea durante la construcción de los EF.

Otro aspecto que determina una mayor calidad en los requisitos, son las jerarquías conceptuales, en este caso, cuando no son visibles, se oculta información valiosa para los requisitos. La falta de identificación de las jerarquías puede provocar servicios

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 16 – Cambios y Agregados a los Escenarios

incorrectos, mal aplicados, falta de precisión, errores de asignación, etc. los cuales perjudicarán la calidad del Sistema de Software a construir. En la Figura 117 se puede observar un ejemplo del LEL, los EF y los requisitos en un caso donde se omitió una jerarquía, llegando a producir requisitos parcialmente correcto, interpretaciones errores, etc. Cabe destacar que es posible que estos defectos sean identificados por los usuarios al validar, pero existe el riesgo de que no lo noten o que lo noten y no lo mencionen debido a que el requisito es parcialmente correcto.

En la Figura 117 se puede observar que los requisitos se refieren a “equipos”. En la Figura 118 se puede observar que este símbolo es en realidad un genérico de una jerarquía y que no es sinónimo de “cajero”. Lo más riesgoso es que es posible que no sea percibido ni por los usuarios ni por los desarrolladores hasta poner el Sistema de Software en producción. Esto se debe a que los usuarios utilizan “Equipo” como sinónimo parcial de “Cajero”, por lo que estas expresiones le parecerán correctas. Este problema se presenta en muchos EF de los cuales se obtuvieron requisitos imprecisos. Fue durante una validación con un usuario cuando se alertó acerca de que los equipos o productos es todo lo que se vende, o sea son los servicios, partes y cajeros. Esto obligó a rever todo el proceso realizado para incorporar la jerarquía correctamente y volver a construir los requisitos.

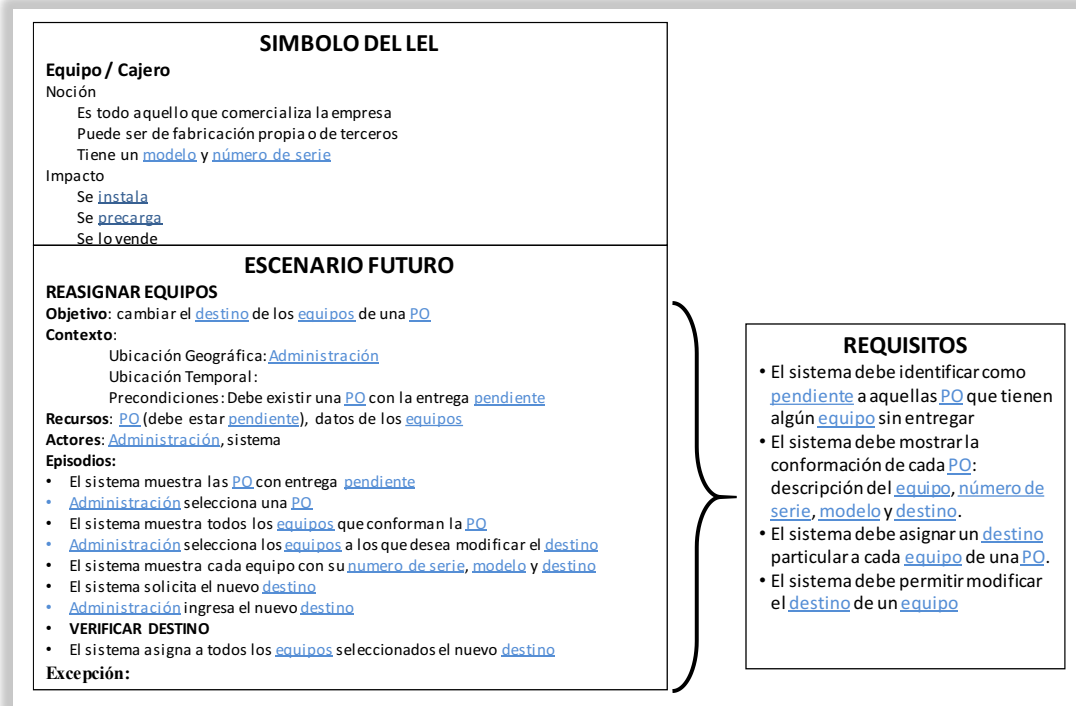


Figura 117 – Ejemplo de jerarquías omitidas y su repercusión en los requisitos (Casa SMAC)

A continuación, se muestra en la Figura 118 la jerarquía existente en el contexto y el símbolo del LEL correctamente descrito.

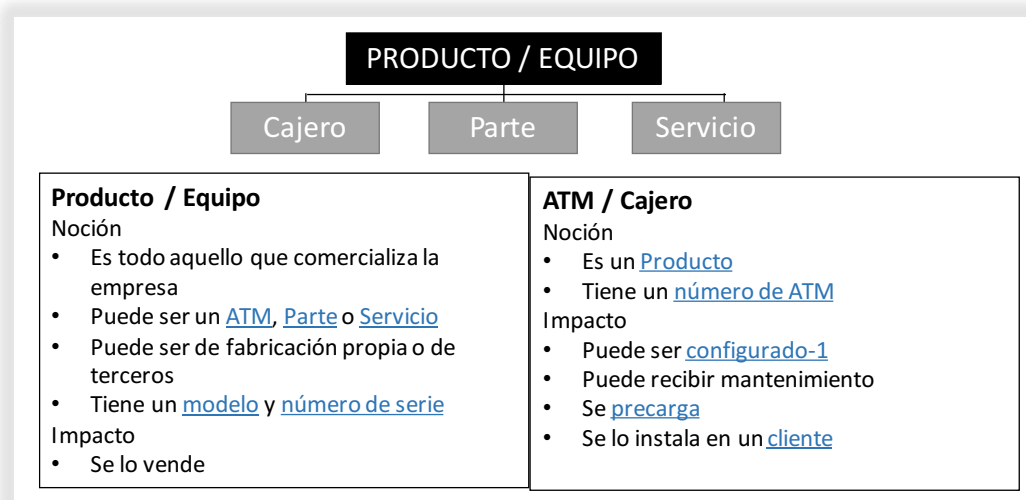


Figura 118 – Ejemplo de símbolo del LEL con Jerarquía no detectada (Caso SMAC)

De esta manera, es posible observar como un error en el LEL se propaga por todos los modelos y puede afectar significativamente la comprensión de los requisitos del software y definir los servicios incorrectamente. En este caso, solo los cajeros o ATM pueden ser entregados en diferentes destinos (por ejemplo, un banco compra cajeros y los envía a diferentes sucursales), las PO con servicios o partes se entregan completas en un mismo

destino. A continuación, se describen los requisitos de manera correcta, reemplazando “Equipo” por “Cajero”:

REQUISITOS

- El sistema debe identificar como pendiente a aquellas PO que tienen algún cajero sin entregar
- El sistema debe mostrar la conformación de cada PO: descripción del cajero, número de serie, modelo y destino.
- El sistema debe asignar un destino particular a cada cajero de una PO.
- El sistema debe permitir modificar el destino de un cajero

Cabe mencionar que las jerarquías conceptuales, los estados y los puntos de vista también permiten verificar la completitud de los Escenarios al analizar el cubrimiento de cada estructura léxica en el conjunto de Escenarios descriptos.

Episodios Ubicuos

Existe un tipo de episodio que no puede ser representado y es aquel que puede ser realizado en cualquier momento del Escenario. Este episodio no tiene secuencia con ningún otro episodio y a la vez con todos. Son aquellas acciones que se desencadenan en cualquier momento, se realizan y se continúa con el siguiente episodio. En la presente tesis se los denomina *episodios Ubicuos* y la propuesta es utilizar un adverbio para identificarlos:

“En cualquier momento” o “En algún momento” o “A cualquier hora”

Como indican las reglas del lenguaje español, después de un adverbio va una coma (“,”). Por lo tanto, la sintaxis es la de la Figura 119. Tratar siempre que sea posible de ubicar el episodio ubicuo como primer episodio del Escenario, esto permite que el lector se entere lo antes posible de su existencia.

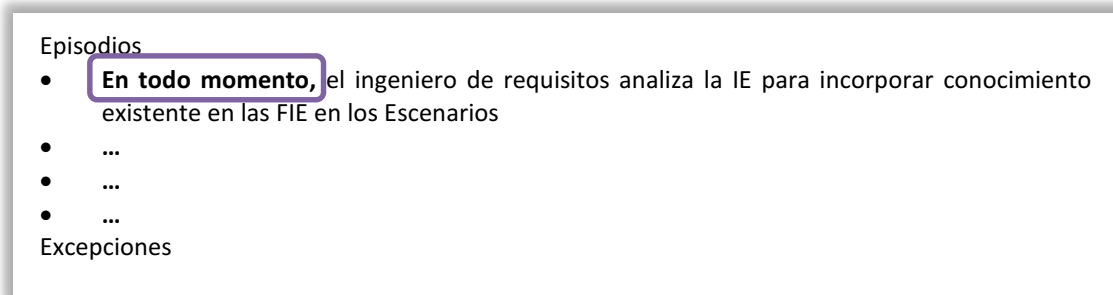


Figura 119 - Ejemplo de un episodio ubicuo

Restricciones a los Actores

La restricción de los actores completa la información de un Escenario. Por las características del componente, se utiliza la misma representación que la de los Recursos. Para el mismo ejemplo utilizado en el **Capítulo 11**, el componente actores debería ser representado de la siguiente manera:

Enfermero (debe tener la certificación para atender pacientes covid)

Otro tema que se mencionó y se relaciona con las restricciones de los actores está asociada a la posibilidad de reflejar cuando un actor comienza un Escenario en un estado específico. En este caso se puede utilizar la precondition del contexto o una restricción del actor, de la misma manera que sucede con los recursos.

Modificación del Modelo Escenarios

En la presente sección se propone una actualización al modelo de Escenario. Por un lado, se han identificados algunos errores en el modelo existente que deben ser atendidos. Por el otro lado, se incorporan los temas tratados en este Capítulo que impactan en el modelo. En la Figura 120 se muestra el modelo de Escenarios que se presentó en el **Capítulo 8**, pero con las observaciones recién mencionadas, las cuales son justificadas a continuación. Las justificaciones a las observaciones realizadas al modelo de Escenarios son las siguientes:

- 1) Modificar la sintaxis de los episodios sub-Escenarios.

Justificación:

La sintaxis es “([Actor | Recurso] + Verbo + Título)”, donde “Título” representa un sub-Escenario. Esta sintaxis puede generar errores, ya que los actores y los recursos de un sub-Escenario son locales y no se suman a los actores y recursos del Escenario que contiene el sub-Escenario. Al eliminar el actor y el recurso de la sintaxis del episodio, el verbo pierde significado. La propuesta es dejar sólo el “Título del sub-Escenario” con su respectivo hipervínculo. De esta manera, además, se simplifica la lectura.

- 2) Incorporar la palabra “restricción” para dejar claro cuando la misma aparece en un Escenario.

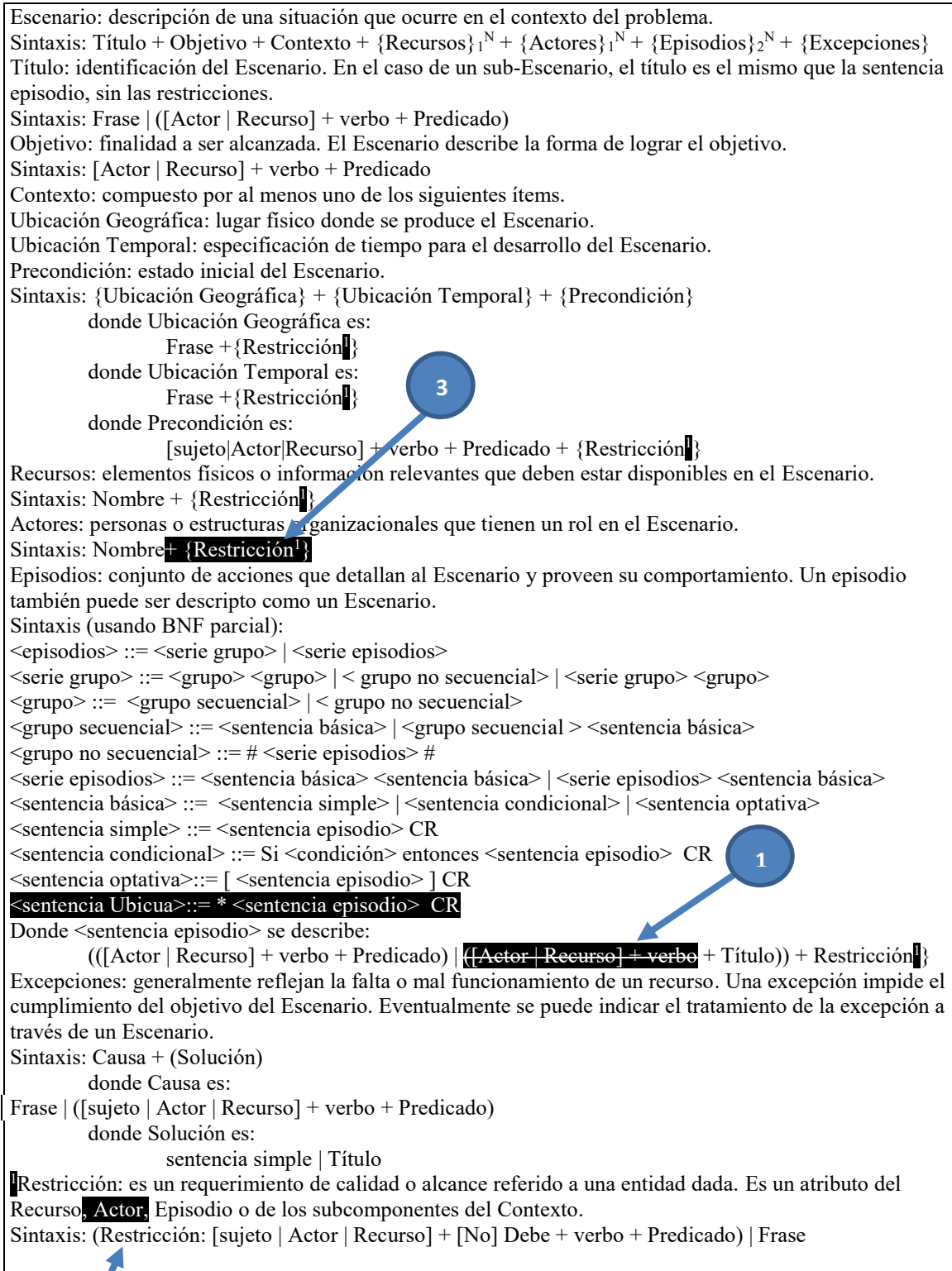
Justificación:

La inclusión de las restricciones puede ser en el contexto, episodios, recursos y desde la presente tesis, en los actores. En el modelo existen diferentes formas de hacerlo. Por ejemplo, en los recursos se utilizan los paréntesis, mientras que en los episodios se coloca una oración debajo del mismo. La utilización de una palabra puede ayudar a una mejor identificación de las mismas. Se propone la inclusión de la palabra “Restricción:” de manera homogénea en todos los casos y a continuación la descripción de la misma.

- 3) Incorporar las restricciones a los actores de un Escenario.

Justificación:

Ya fue descrita en este mismo Capítulo, en la sección Restricciones a los Actores. La propuesta es utilizar la misma sintaxis que para los recursos.



2

Figura 120 – Modelo Escenarios actualizado

Capítulo 17

Actualización de la Inspección de Escenarios

Resumen

La inspección de Escenarios, como sucede con el LEL, debe ser actualizada incorporando los cambios y agregados de la presente tesis. Como ya se mencionó, la Verificación es un tema de gran amplitud y excede el objetivo central de la presente tesis. En este Capítulo se describen los cambios necesarios para los formularios de inspección de Escenarios existentes y se proponen algunos nuevos, sólo como ejemplo.

Los formularios de inspección existentes para verificar los Escenarios se encuentran en el **Anexo A – Verificación de consistencias en el LEL y en los Escenarios**. En este anexo existen dos reportes referidos a los Escenarios: “Reporte de consistencia interna de Escenarios” y el “Reporte de consistencia entre Escenarios” [Leite05]. La primera inspección se ocupa de detectar las discrepancias, errores y omisiones en cada Escenario y la segunda, detecta discrepancias, errores y omisiones en el conjunto de Escenarios realizados por un mismo ingeniero/a de requisitos.

Cambios en la Inspección “Consistencias entre Escenarios”

Los cambios y agregados aplicados al LEL en el **Capítulo 12** y a los Escenarios en el **Capítulo 16**, no modifican los formularios de inspección existentes, pero obligan a incorporar nuevos formularios, los cuales se describen en la Tabla 29. Cabe destacar que esta actualización se concentra en las propuestas más importantes de la presente tesis, como es la incorporación de jerarquías conceptuales y puntos de vista.

Título de Formulario	¿Qué detectan?
CUBRIMIENTO DE LOS ESTADOS	<ul style="list-style-type: none">• Identificar omisiones y usos incorrectos de los estados del LEL
CUBRIMIENTO DE LAS JERARQUIAS	<ul style="list-style-type: none">• Identificar que no se han utilizado correctamente los elementos de una Jerarquía conceptual
CUBRIMIENTO DE LOS PUNTOS DE VISTA	<ul style="list-style-type: none">• Identificar si se han pasado correctamente los puntos de vista “Es, Deber ser y Será”
NUEVO EPISODIO	<ul style="list-style-type: none">• Asegurar el uso correcto de los episodios Ubicuos

Tabla 29 – Nuevos formularios para la Inspección de Escenarios

Actualización de la Inspección de Escenarios

A modo de ejemplo, se describe a continuación, un par de dichos formularios. En el primero se inspecciona la utilización en los Escenarios de las jerarquías definidas en el LEL y en el otro, el pasaje correcto de un punto de vista del LEL a los EA.

Uso de jerarquías contextuales del contexto

Esta inspección detecta el uso de las jerarquías conceptuales del LEL en los EA. Cabe aclarar que esta inspección pone en alerta posibles omisiones, dejando a criterio del ingeniero/a de requisitos si es correcta la inclusión de más información. Esto se debe a que una jerarquía puede ser definida en el LEL, pero no participar completa en los EA.

JERARQUÍA 1					
ESCENARIO	Cant. De ocurrencias del genérico	Cant. de ocurrencias de la especialización 1	Cant. de ocurrencias de la especialización 2	...	Cant. de ocurrencias de la especialización n
1	0	1	0	...	0
2	1	1	0	...	0
3	0	0	1	...	1
TOTAL	1	2	1	...	+ 1

Nuevo Form1. Para la Detección de Ocurrencia de Jerarquías

En el formulario *Nuevo Form1* se puede observar que todos los símbolos integrantes de la **JERARQUÍA 1** han sido tratados en algún Escenario ya que el total de cada columna de la misma es mayor a cero (“total>0”). Cabe aclarar que este formulario no informa si la inclusión de cada símbolo de la jerarquía en los diferentes Escenarios es correcta ni tampoco que no falte información referida a ella.

JERARQUÍA 2					
ESCENARIO	Cant. De ocurrencias del genérico	Cant. de ocurrencias de la especialización 1	Cant. de ocurrencias de la especialización 2	...	Cant. de ocurrencias de la especialización n
1	0	1	0	...	0
2	0	1	0	...	0
3	0	0	0	...	1
TOTAL	0	2	0	...	1

En el segundo ejemplo del *Nuevo Form1*, **JERARQUÍA 2**, se puede observar que se detecta la omisión del uso del genérico y de la especialización 2. Esto puede indicar que hay un mal uso de alguno de los símbolos de la jerarquía en alguno de los Escenarios o que parte del contexto no fue modelado.

En este caso, **JERARQUÍA 3**, se puede observar que sólo se ha utilizado el genérico. Esto puede indicar falta de detalle en los episodios u omisión de episodios, incluso la omisión de Escenarios que contengan estas especializaciones.

JERARQUÍA 3					
ESCENARIO	Cant. De ocurrencias del genérico	Cant. de ocurrencias de la especialización 1	Cant. de ocurrencias de la especialización 2	...	Cant. de ocurrencias de la especialización n
1	2	0	0	...	0
2	1	0	0	...	0
3	3	0	0	...	0
TOTAL	6	0	0	...	0

Puntos de vista en los Escenarios

Este formulario detecta si los puntos de vista incluidos en el LEL han sido utilizados correctamente en los EA. Una de las estructuras que se verifica es que el impacto del LEL que contiene el punto de vista del “Deber ser”, se transforme en un episodio con restricción. En el ejemplo que sigue se puede observar como un impacto con punto de vista se transforma en el episodio con restricción.

3. El jefe de Producción solicita telefónicamente la materia prima. Lo debe realizar personalmente o por mail.



2. El jefe de Producción solicita telefónicamente la materia prima.
Restricción: **se debe** realizar personalmente o por mail.

El formulario que sigue debe ser parte de un conjunto de formularios que verifiquen todos los puntos de vista del LEL.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 17 – Actualización de la Inspección de Escenarios

SÍMBOLOS CON PUNTO DE VISTA	Impacto con punto de vista	EA que representa el punto de vista	¿Existe la restricción?
1	3	2	Si
2	1	0	-
3	2	4	Si
3	2	11	No

Nuevo Form2. Para identificar si fueron incluidos los puntos de vista

En el *Nuevo Form2* los símbolos e impactos se han enumerado para simplificar la carga del formulario. La inspección identifica los impactos con puntos de vista. En el caso de la primera fila, el traslado fue correcto. No se sabe si fue incluido en todos los lugares que corresponden, sólo que fue trasladado a los EA. En el caso del símbolo 2, se debe buscar en qué EA se debe incluir e incorporarlo. Este es el ejemplo de un punto de vista que no fue trasladado a los Escenarios. En el caso del símbolo 3, se incorporó correctamente en el Escenario 4 pero no sucedió lo mismo en el Escenario 11. En este último caso, se sugiere analizar el Escenario para detectar si sólo se trata de un tema sintáctico o si se omitió información.

Relación entre la inspección de EA y de los EF

A continuación, en la Tabla 30, se presenta la lista de formularios existentes para los EA y se indica cuales requieren ser actualizados para realizar la inspección de EF.

Títulos de Formularios Existentes	UdeD futuro
CARÁTULA	Si
RESUMEN CUANTITATIVO DEL CONJUNTO DE ESCENARIOS	SI
VERIFICACIÓN SINTÁCTICA	SI
INTERRELACIÓN CON EL LEL	No aplica
INTERRELACIÓN DE COMPONENTES	SI
COMPOSICIÓN DE ESCENARIOS INTEGRADORES	SI
COMPROBACIÓN DE SUB-ESCENARIOS	SI
CONSISTENCIA DE PRECONDICIONES EN EL CONTEXTO	SI
RELACIÓN ENTRE ESCENARIOS Y SUB-ESCENARIOS	SI
CONTROL DE SUPERPOSICIÓN DE ESCENARIOS	SI
VERIFICACIÓN DE OCURRENCIA DE ACTORES	SI
VERIFICACIÓN DE OCURRENCIA DE RECURSOS	SI
SÍMBOLOS DEL LEL NO UTILIZADOS	No aplica
CONTROL DE IMPACTOS VERSUS ESCENARIOS/EPISODIOS	No aplica

Tabla 30 – Cubrimiento de la Inspección de EA en los EF

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 17 – Actualización de la Inspección de Escenarios

En la Tabla 31 se describen los formularios que se deben incluir a la inspección de EA y se menciona si aplican a los EF.

Títulos de Formularios Nuevos	¿Aplica al UdeD futuro?
CUBRIMIENTO DE LOS ESTADOS	SI
CUBRIMIENTO DE LAS JERARQUÍAS	SI
EPISODIOS UBICUOS	SI

Tabla 31 – Nuevos formularios de inspección para EF

La inspección de Escenarios del Anexo A fue construida para los EA. Puede observarse que las similitudes con el UdeD futuro son muchas, ya que utilizan el mismo modelo. Se debe tener en cuenta que todo lo referido a la relación “LEL – EA”, a partir de la presente tesis, se debe modificar en el UdeD futuro por la relación “LEL_R – EF”. Toda la inspección del LEL debe ser adaptada a las particularidades del LEL_R y lo mismo sucede con los EF.

Trabajo Futuro

En el presente Capítulo se realizó una revisión preliminar de la inspección de los Escenarios, la cual debe ser continuada para asegurar la consistencia de los EA y de los EF. Queda pendiente para trabajos futuros:

- Incluir los cambios del **Capítulo 16** en los formularios de inspección existentes.
- Crear nuevos formularios para los agregados del **Capítulo 16** que no hayan sido incluidos en el punto 1.
- Crear la inspección para los EF tomando en cuenta sus particularidades. Por ejemplo, que la utilización del nuevo actor *sistema* sea correcta en los episodios, que los episodios que realiza el *sistema* tenga el nivel de detalle adecuado, etc.

Capítulo 18

Derivación de Escenarios Actuales por Proximidad

Resumen

El análisis realizado sobre la heurística existente arrojó un problema similar al de la construcción del LEL. En este caso, la lista inicial de Escenarios Actuales candidatos cumple una función similar a la lista de símbolos del LEL. Por tal motivo, se generó una nueva *Heurística de Derivación de Escenarios Actuales por Proximidad* que reemplaza a la existente, la que además de generar la primera versión de los EA, otorga un conjunto de nuevas premisas para describir los Escenarios.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 18 – Derivación de Escenarios por Proximidad

Como ya se mencionó, el principal problema de la derivación existente radica en la diferencia de granularidad y de ordenamiento de la información entre el LEL y los Escenarios. Esto sucede porque el objetivo del LEL es describir términos, mientras que los Escenarios describen el proceso del negocio observable. De esta manera, la información de una situación puede estar dispersa en varios símbolos, generando Escenarios ficticios. En virtud de esto, el mecanismo que propone la heurística existente de pasar información desde el LEL a los Escenarios de manera textual, se debe descartar. Esto no significa eliminar la derivación, sino aprovechar la información disponible mientras sea beneficiosa. Otra desventaja que se detectó en la heurística existente es que no consulta toda la información disponible, desaprovechando información valiosa. Aun cuando no se disponga de documentación del macrosistema, para construir el LEL fue necesario realizar entrevistas y sus transcripciones son un documento de mucho significado para la IR en general y, para la derivación de Escenarios en particular. Utilizar la documentación existente ayuda a reestructurar el orden intrínseco del glosario, desde una perspectiva declarativa a una procedural. Este nuevo mecanismo de derivación, a diferencia de la derivación existente, incluye a todos los símbolos del LEL, sin excepciones, con un adecuado tratamiento de los estados, las jerarquías conceptuales y los puntos de vista del macrosistema. El resultado de la nueva heurística es un conjunto de Escenarios derivados más representativos y completos.

Nueva Actividad Derivar

La nueva Derivación incluye todos los cambios y agregados del LEL y de los Escenarios (Capítulos 12 y 16). Una característica singular del nuevo mecanismo es que se superpone con la actividad Describir, no siendo tan claro cuando termina una y comienza la otra. La nueva Derivación toma los verbos del LEL, los cuales actúan como desencadenantes para orientar la búsqueda de información en la documentación. Esta búsqueda abandona la mirada léxica para abocarse a una estricta mirada procedural. Durante la elaboración de la presente estrategia, como sucedió con la Heurística de Construcción del LEL por Proximidad en el Capítulo 15, se tomó en cuenta la dispersión de la información y la necesidad de concentración del ingeniero/a de requisitos. Tomando en cuenta estos dos aspectos, dispersión de la información y concentración del ingeniero/a, es posible trabajar de dos maneras diferentes:

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 18 – Derivación de Escenarios por Proximidad

- Describir un Escenario por vez recorriendo todas las FI disponibles.
- Analizar cada FI hasta agotarla, o sea completar todos los Escenarios posibles con cada documento.

Las pruebas preliminares, han mostrado que concentrarse en un documento favorece la concentración del ingeniero/a de requisitos. De esta manera, se fortalece la construcción del LEL, volviendo a elegir una construcción por proximidad, iterativa e incremental, donde a partir de unos pocos Escenarios iniciales se obtiene el resto. Los nuevos Escenarios aparecen durante la descripción de los Escenarios previamente detectados, lo que implica que debe haber un paso inicial que dé origen a un conjunto reducido de Escenarios que se constituirán en las semillas de toda la actividad de derivación. Tanto los *Escenarios semilla* como los que se vayan detectando por proximidad se irán completando a medida que se avanza en la actividad de derivación. Cuando un Escenario aparece muy tarde en el proceso, es muy poco probable que sea necesario volver a revisar los documentos ya analizados. Esto ocurre porque una aparición tardía está casi sin excepción asociada a un documento que aborda aspectos del proceso del negocio no detallados en los documentos ya utilizados. La idea es ir construyendo los Escenarios de a pequeños grupos fuertemente vinculados. Al finalizar la derivación de los Escenarios, se debe consultar el LEL para verificar el cubrimiento del mismo con el objetivo de detectar posibles omisiones.

Heurística de Derivación

Puede observarse que sólo se utilizan documentos escritos, ya que la actividad de derivación es un mecanismo que realiza el ingeniero/a de requisitos en soledad. El primer paso de la heurística es analizar el Objetivo General del Sistema. En este momento se identifican aquellas acciones que mejor lo definen. Estas acciones permitirán identificar símbolos verbos del LEL con los cuales es conveniente comenzar el proceso de derivación. La salida de la actividad es un conjunto de EA derivados, los cuales representan fidedignamente el proceso del negocio observable.

En la Figura 121 se puede observar el marco conceptual del proceso. La actividad Derivar tiene por entrada el Objetivo General del Sistema, el LEL y los documentos. Se deben seleccionar sólo aquellos documentos del macrosistema que estén vigentes.

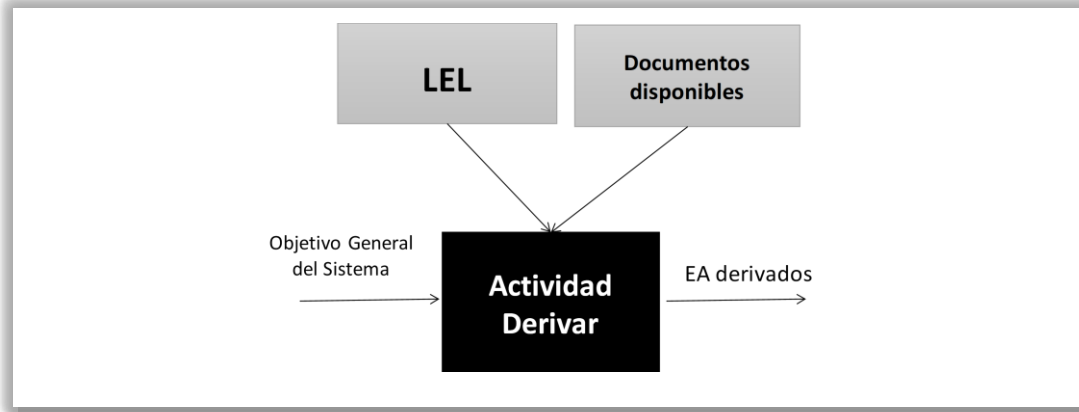


Figura 121 – Marco conceptual de la Actividad Derivar

La estrategia de derivación consiste, como puede observar en la Figura 122, en tomar información desde el LEL (ver flecha identificada con el “1”), pero también es utilizada como guía para saber “qué buscar” en el “*Doc.1*”. Este primer documento es la transcripción de la entrevista inicial del proyecto. Se construyen los EA a partir de los verbos y los impactos de los sujetos del LEL que contienen verbos que no son símbolos del LEL. Luego, en la flecha “2” se busca en el documento los símbolos seleccionados, analizando la información existente desde una perspectiva de procesos y hasta agotar todo el documento. En la flecha “3” se completan los Escenarios hasta donde sea posible con la información encontrada en el paso 2. En la flecha “4” se busca en los documentos restantes los nombres de los Escenarios construidos. Se completan los Escenarios con la información existente en estos documentos. Se debe tener en cuenta que se analiza un documento por vez hasta agotarlo, recién entonces se pasa al siguiente. Finalmente, en la flecha “5”, se verifica el cubrimiento del LEL para detectar omisiones.

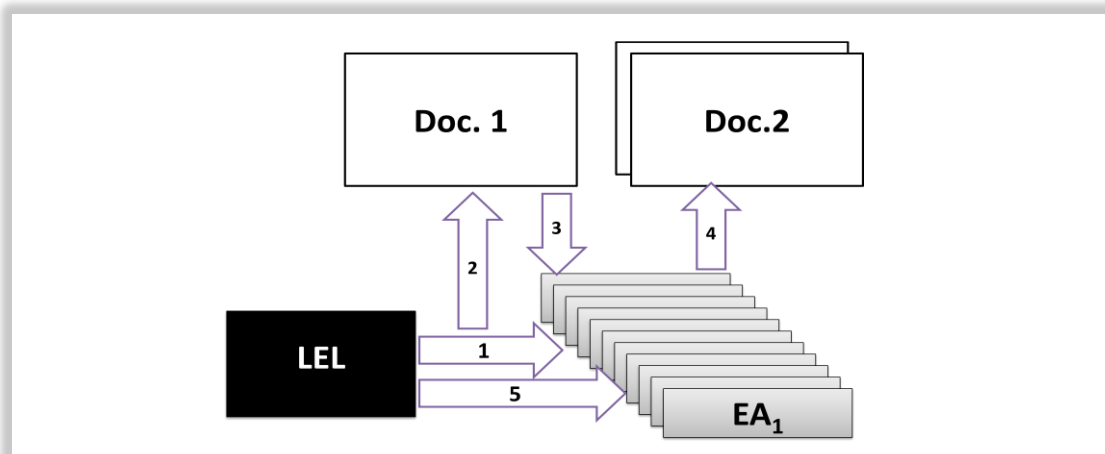


Figura 122 – Estrategia de Derivación de Escenarios por proximidad

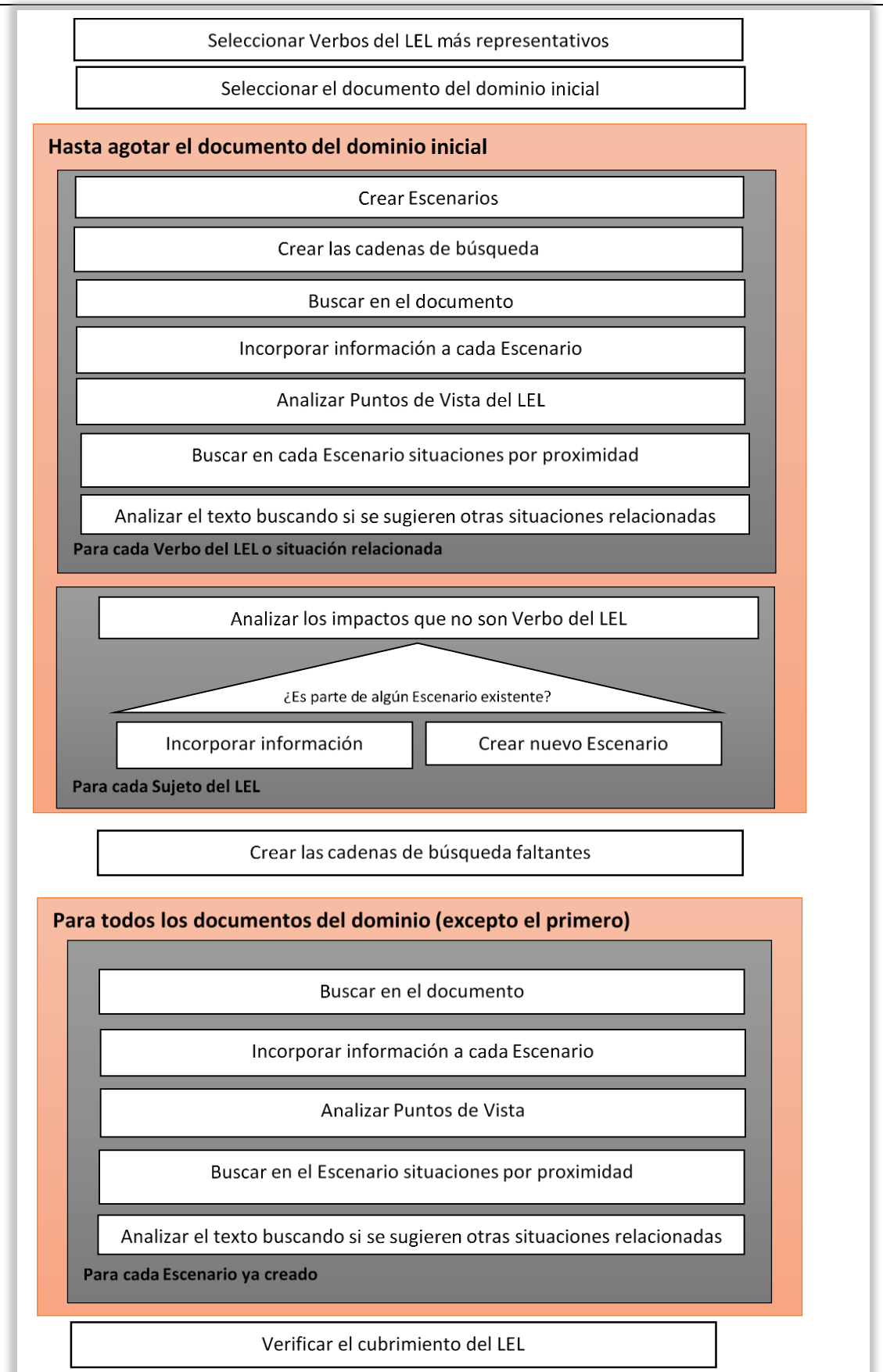


Figura 123 – Heurística de Derivación de Escenarios por Proximidad

A continuación, se describe cada paso de la heurística de Derivación (ver **Figura 123**):

Paso 1:

“*Seleccionar verbos del LEL más representativos*”, preferentemente, aquellos que mejor definan el Objetivo General del Sistema, pero es posible comenzar por cualquier verbo. Siempre trabajar con grupos reducidos o de a uno por vez.

Paso 2:

“*Seleccionar el documento del macrosistema inicial*”. Aquí sucede algo similar a lo descrito en el Paso 1, se puede comenzar por cualquier documento del macrosistema, pero para un mejor aprovechamiento de la información disponible, se sugiere comenzar por la transcripción de la entrevista donde se definió el proyecto, aquella donde el cliente-usuario describió el contexto en general, usualmente desde el punto de vista del Objetivo General del Sistema. De comenzar por otro documento, asegurarse que su contenido cubra todo el contexto en estudio.

Paso 3:

Se deben “*crear Escenarios*”, uno por cada símbolo verbo. El título de cada Escenario será igual al nombre del símbolo verbo que le dio origen. Cabe destacar que este título puede ser modificado para que sea más representativo. Incorporar los Escenarios a la lista de Escenarios.

Siempre que se analice un símbolo del LEL se deben considerar también los símbolos relacionados. Esto permite detectar:

- **Estados** asociados: buscar en la noción frases del tipo “Los Estados son” y tomar de lista los Estados asociados. Luego, analizar si esos Estados tienen alguna incidencia en el Escenario que se está describiendo.
- **Jerarquías**: si algún símbolo es parte de una jerarquía se debe asegurar su correcto uso del genérico o de la especialización, según corresponda.

Se puede activar la Vista Clasificación para acceder rápidamente a la información del símbolo.

Paso 4:

Para realizar la exploración en el documento, se deben “*crear las cadenas de búsqueda*” con los nombres de los símbolos verbos. Armar la expresión de cada cadena de búsqueda de la siguiente manera: Crear una cadena de caracteres con el

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 18 – Derivación de Escenarios por Proximidad

nombre del símbolo verbo. También hacerlo con partes del nombre del símbolo. La búsqueda puede ser parecida a la utilizada al realizar una búsqueda sistemática de la literatura o una revisión sistemática [Kitchenham07], se intenta asegurar que se han buscado todas las opciones posibles.

Las cadenas de búsqueda serán utilizadas en cada documento que se analice, por lo que se sugiere guardarlas para reutilizarlas. Se debe recordar que las cadenas de búsqueda se repiten para cada Escenario que se está describiendo y luego, para cada documento existente.

Paso 5:

“*Buscar en el Documento*” se refiere a ejecutar las cadenas de búsqueda en el documento. Se trata de identificar toda información relevante para el Escenario, esta puede estar dispersa en el documento. Esto último dependerá de la perspectiva del macrosistema que tenga el documento en cuestión.

Paso 6:

Cada vez que se encuentre información en el documento, se debe analizar su pertinencia con la situación que se está describiendo. Cuando la información corresponda a dicha situación “*incorporar la información a cada Escenario*”. Puede existir información para cualquier componente de un Escenario.

Paso 7:

“*Buscar en los Escenarios situaciones por proximidad*”. Se debe analizar todo el Escenario para detectar la presencia de nuevas situaciones próximas que no fueron vistas aún. En este caso se deben analizar todos los componentes.

Paso 8:

“*Analizar el texto buscando si se sugieren otras situaciones relacionadas*”. Cuando se está buscando información en el documento, es posible detectar más información acerca de la *jerarquía de los Escenarios*:

- Situación con sub-situaciones. En este caso, crear episodios del tipo sub-Escenario. Incorporar los episodios sub-Escenarios y anotar estas sub-situaciones para describirlas luego.
- La situación como parte de otra mayor. Identificar el Escenario de mayor nivel y comenzar a describirlo, poniendo el Escenario que se está describiendo como sub-Escenario.

Tener presente que siempre se debe agotar el documento en el cual se está buscando. Recién entonces pasar al siguiente.

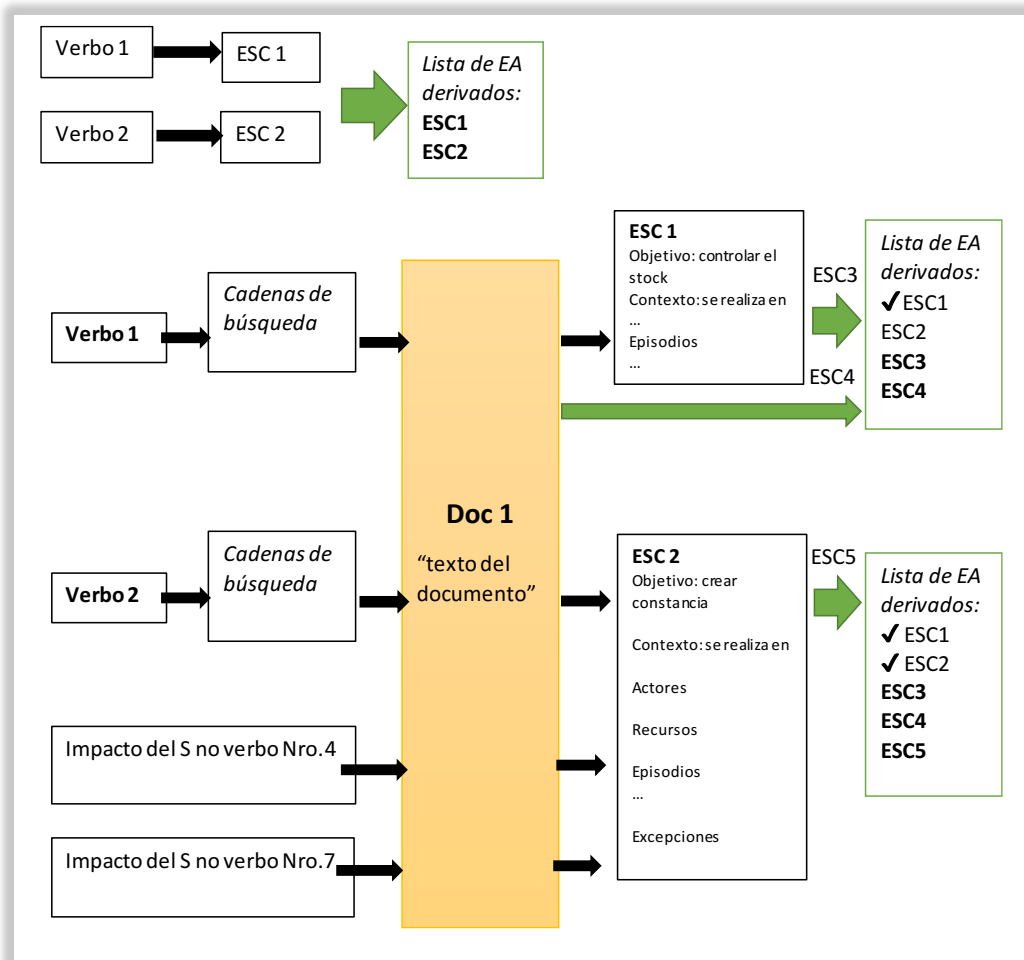


Figura 124 – Ejemplo de Derivación de EA por Proximidad desde el LEL

En la Figura 124 se puede observar que se parte de los verbos del LEL, en este caso Verbo1 y Verbo2. Se construyen los Escenarios derivados correspondientes y se incluyen en la lista de Escenarios. Luego, a partir de cada verbo, se crea la cadena de búsqueda y se busca en el documento seleccionado para tal fin. Si existe información, se completa el Escenario, en este caso el “ESC 1”. Con el Escenario parcialmente completo se buscan nuevas situaciones por proximidad, por un lado, analizando el Escenario en sí mismo donde se detectó el “ESC 3” y en el otro, buscando en el documento información acerca de la *jerarquía de Escenarios*, o sea, si la situación descripta es parte de una situación mayor o contiene situaciones incrustadas, “ESC 4”. En estos casos, las nuevas situaciones se incorporan a la lista de Escenarios. Luego, se toma otro verbo y se repite el mismo procedimiento, siempre con el mismo documento. De este último se encuentra por proximidad en el Escenario el “ESC 5”. Una vez

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 18 – Derivación de Escenarios por Proximidad

realizado el proceso para verbos del LEL se debe continuar con los impactos de los sujetos que no son verbos (Paso 9). Se sigue trabajando con el primer documento.

Paso 9:

Una vez analizados todos los verbos se deben *“analizar los impactos de los sujetos que no son verbos”*. Esto permite identificar información para completar los Escenarios existentes, pero en algunos casos, pueden aparecer nuevas situaciones:

- Si existe el Escenario, *“incorporar información”*.
- Si no existe el Escenario, *“crear Nuevo Escenario”* e incluir dicha información.

Se puede observar en la Figura 124 que las dos últimas cajas representan los impactos no verbo de los sujetos. Estos impactos se buscaron en el “Doc 1” y se detectó información para completar el “ESC 2”.

En este momento, se debe *“Analizar los Puntos de Vista del LEL”*, o sea identificar en los impactos de los símbolos del LEL alguna referencia a puntos de vista. Por ejemplo, impactos con dos oraciones, expresiones que permitan **“Identificar es/deber ser/será”** (ver **Capítulo 12**). Si se detecta un punto de vista debe ser trasladado adecuadamente al Escenario (ver **Capítulo 16**).

Paso 10:

“Crear las Cadenas de Búsqueda faltantes”. Se deben generar las cadenas de búsqueda de todos los Escenarios que se detectaron por proximidad (en la Figura 124 los “ESC3”, “ESC4” y “ESC5”). Se debe recordar que se habían creado las cadenas de búsqueda de los Escenarios derivados de verbos, “ESC1” y “ESC2”.

En este momento, se cuenta con un conjunto de Escenarios derivados que abarcan todo el contexto. Por este motivo es relevante la elección del primer documento. El nivel de completitud de estos Escenarios dependerá de la calidad de este documento.

Paso 11:

Ahora se deben analizar los documentos restantes, o sea todos aquellos que no fueron utilizados en la primera parte.

Para cada documento que aún no fue analizado, se deben buscar todas las cadenas de búsqueda guardadas previamente. Para ello, se selecciona un Escenario de la lista y se aplica las cadenas de búsqueda que le corresponden. Se *“busca en el documento”* toda

información relacionada al Escenario. Si existe información se “*incorpora la información al Escenario*”.

Paso 12:

“*Analizar Puntos de Vista*” consiste en analizar los episodios existentes en el Escenario con la nueva información del documento. Si aparecen discrepancias, deben ser descriptas utilizando la sintaxis detallada en el **Capítulo 16**.

Paso 13:

“*Buscar en los Escenarios situaciones por proximidad*”. En este momento es menos probable que aparezcan. Por cada Escenario que se completa con la información del documento, se deben buscar nuevas situaciones relacionadas.

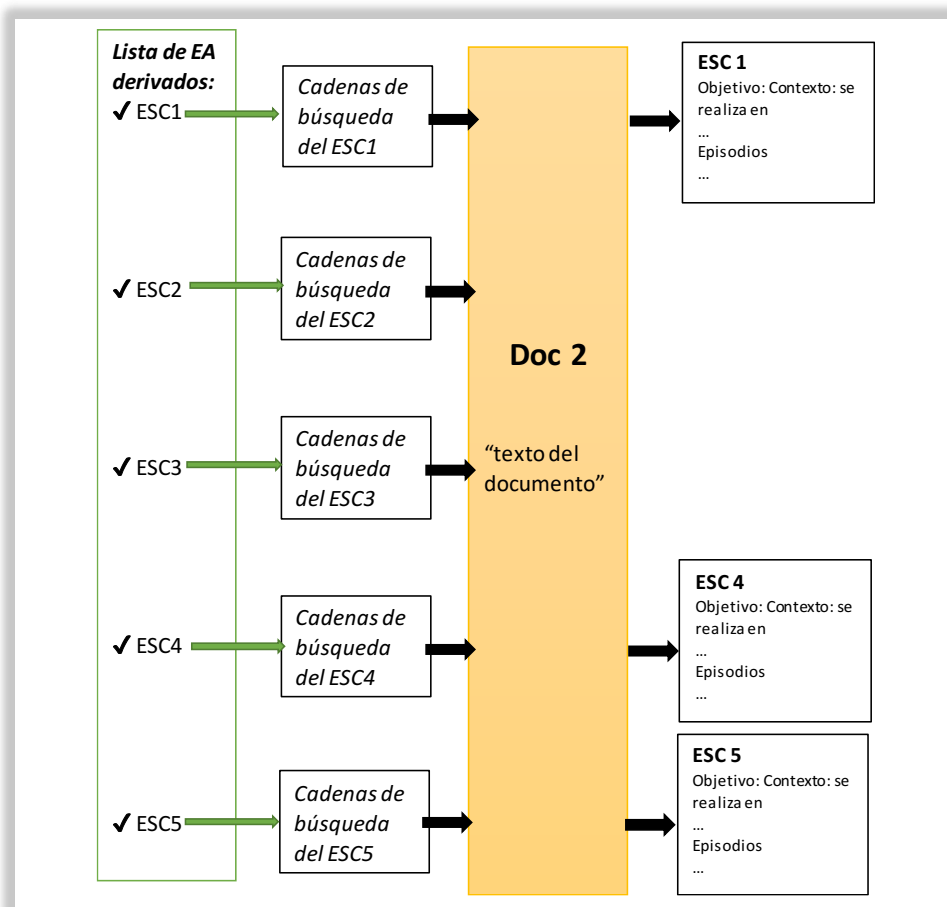


Figura 125 – Ejemplo de derivación de EA por proximidad desde documentos

En la Figura 125 se puede observar la elicitación de información desde el “Doc 2”. En este caso las cadenas de búsqueda ya fueron creadas y se aplican en el documento para detectar información particular para cada Escenario. Puede observarse que no se ha encontrado ninguna información para los “ESC2” y “ESC3”. Podría haber aparecido información por proximidad de nuevas situaciones. En este ejemplo no se representa,

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 18 – Derivación de Escenarios por Proximidad

pero de existir se crea un nuevo Escenario y se completa con la información disponible en el mismo documento.

Paso 14:

“Analizar el texto buscando si se sugieren situaciones relacionadas”. Nuevamente, esto permite detectar Escenarios por proximidad. Si se detectan nuevas situaciones, crear el Escenario y completarlo hasta donde se pueda. Si esto sucede, es probable que se deba consultar nuevamente algún documento ya analizado, pero la probabilidad de que esto suceda es muy baja.

En síntesis, para cada documento organizacional se deben ejecutar todas las cadenas de búsqueda creadas y reservadas durante el análisis del documento inicial. En este momento, se considera que todos los Escenarios ya han sido creados, por lo tanto, se debe concentrar en recorrer cada documento hasta agotarlo buscando nueva información. De encontrarla, se debe incorporar al Escenario correspondiente. También, se debe analizar cada Escenario para detectar nuevas situaciones por proximidad.

Paso 15:

“Verificar Cubrimiento del LEL”. Finalmente, se debe volver al LEL para asegurarse que no quedaron partes del contexto sin analizar. Ver que todos los verbos se hayan transformado en Escenarios y que todos los impactos de los sujetos estén incluidos en algún Escenario como episodios o sean en sí mismos un Escenario. En este último caso, se sugiere agregar el símbolo verbo de ese impacto en el LEL y modificar el impacto del sujeto. También se debe controlar que todos los objetos hayan sido incluidos en algún Escenario y que los estados estén asociados a sus símbolos base. Esta heurística ha sido probada en algunos pocos casos dando resultados promisorios. Se presenta a continuación la prueba realizada con el Caso SNor.

Planificar la producción

Noción

- Conjunto de acciones que permite organizar la [fabricación](#) de una semana
- Se realiza los martes y comienza su vigencia es desde el miércoles en el [turno](#) de 14 a 22 hasta el otro miércoles en el [turno](#) de 6 a 14.
- Es realizada por el [Jefe de Producción](#) o por un [oficial planificador](#).

Impacto

- Se estudian las [fallas](#) ocurrida durante la [fabricación](#)
- Se procesa y ordena la información enviada por los [encargados de planta](#)
- Se procesa y ordena la información enviada por la gente del Depósito de tamaño fijo
- Se elaboran 21 [programas de producción](#), uno para cada [turno](#).

Figura 126 – Ejemplo Símbolo Verbo que inicia la Derivación por Proximidad (Caso SNor)

Para buscar información en el documento (en la Figura 124 corresponde a “Doc 1”) se generó una cadena de búsqueda con “Planificación de la Producción”, “Planificar la Producción”, “Planific*” y Produc*. En la búsqueda en el documento se detectaron dos párrafos donde se mencionaba la planificación de la producción. Se presentan a continuación ambos párrafos:

Primer párrafo:

[...]PLANIFICACIÓN DE LA PRODUCCIÓN

La oficina de planificación de la producción está formada por tres personas: el jefe de producción y dos oficiales planificadores. Cualquiera de estas personas está capacitada para planificar la producción, sin embargo, las tareas se dividen de la siguiente manera: el jefe de producción se encarga fundamentalmente de estudiar las fallas ocurridas durante la fabricación y su manera de prevenirlas en el futuro, uno de los oficiales se ocupa de procesar y ordenar la información enviada por los encargados de planta y del depósito de cajas de tamaño fijo, finalmente el otro se ocupa de realizar efectivamente los programas de fabricación.

La planificación de la producción se realiza los días martes y la vigencia de la misma se extiende desde el turno de 14 a 22 del día miércoles hasta el turno de 6 a 14 del miércoles siguiente.

Durante la planificación de producción se elaboran 21 programas de producción, uno para cada turno de la semana. El encargado de planta de cada turno envía al final de su horario de trabajo todas las órdenes de producción terminadas, así como un listado de órdenes de producción parcialmente completadas durante su turno. Es decir, las órdenes incompletas quedan en poder del encargado de planta quién se las entrega al encargado de planta del turno siguiente.

Tan pronto como la orden de compra es aprobada en la oficina comercial de Norpack, una copia de la misma es remitida a la oficina de planificación de la producción.

En esta oficina se elabora una orden de producción que contiene una descripción detallada de la caja a ser fabricada. Si la orden de compra es por más de un tipo de caja se elabora una orden de producción para cada tipo de caja.

La orden de producción contiene información acerca del despliegue de la caja o de los accesorios, es decir, de la caja “desarmada”. Este despliegue es de vital importancia en la planificación de la producción, ya que condiciona la forma en que la caja se habrá de fabricar y el consumo de materia prima que se realizará en el proceso [...]

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 18 – Derivación de Escenarios por Proximidad

Segundo párrafo:

[...] La planificación de la producción se realiza priorizando aquellas órdenes de producción que ocasionen el menor desperdicio de materias primas. Sin embargo, si alguna orden de producción, difícil de complementar con alguna otra, se atrasa en demasía y el plazo de entrega se reduce a menos de 10 días, la misma se transforma en una orden de producción urgente. Las ordenes de producción urgentes son siempre incluidas en los programas de fabricación más próximos posibles. Las cajas de tamaño fijo suelen ser utilizadas como último recurso para complementar ordenes de producción urgentes [...]

Con el símbolo verbo detallado en la Figura 126 y la información del Párrafo 1, se construyó el Escenario derivado descrito en la Figura 127.

Planificar la Producción

Objetivo: Generar los [programas de fabricación](#) para una semana.

Contexto:

- Ubicación Geográfica:** [Oficina de planificación de la producción](#)
- Ubicación Temporal:** martes de 9 a 13 y de 14 a 18
- Precondición:** Debe haber una [orden de compra aprobada](#).

Recursos: copia de la [Orden de compra](#) (debe estar [aprobada](#)), información del [Depósito de Tamaño Fijo](#), información del [encargado de planta](#)

Actores: [Jefe de producción](#), [Oficial Planificador](#)

Episodios:

1. El [oficial planificador](#) analiza las [órdenes de compra](#)
2. Si la [orden de compra](#) tiene más de un tipo de [caja](#) entonces genera una [orden de producción](#) por cada [tipo](#) de [caja](#)
3. Estudia las [fallas](#) ocurrida durante la [fabricación](#)
4. Si una [orden de producción](#) es difícil de complementar con alguna otra o se atrasa en demasía y el plazo de entrega se reduce a menos de 10 días entonces [prioriza la orden de producción urgente](#)
5. Procesa y ordena la información enviada por los [encargados de planta](#)
6. Procesa y ordena la información enviada por el [Depósito de Tamaño Fijo](#)
7. [Prioriza las órdenes de Producción por desperdicio](#)
8. Elabora 21 [programas de fabricación](#)

Excepciones:

Cuando una [orden de producción](#) no puede ser incorporada a ningún [programa de fabricación](#) entonces se la delega a la Papelera del Sudoeste.

Figura 127 – Ejemplo de un Escenario Derivado por Proximidad (Caso SNor)

Cabe destacar que el ejemplo de la Figura 127 existen episodios que aún se desconoce si corresponden a una acción simple o son varios episodios y corresponde crear sub-Escenarios. Por ejemplo:

- 1) ¿Qué significa analizar de la OC, qué datos mira y para qué?
- 2) ¿Cuáles son las acciones y documentos involucrados al estudiar las fallas de fabricación?
- 3) ¿Qué significa procesar y ordenar la información enviada por los encargados de planta? ¿Cuál es la información que recibe y qué sucede si no la recibe?

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 18 – Derivación de Escenarios por Proximidad

- 4) ¿Qué significa procesar y ordenar la información enviada por la gente del Depósito de Tamaño Fijo? ¿Cuál es la información que recibe y en qué casos?
- 5) ¿Cómo se reconoce que la orden de producción es urgente? ¿Lleva alguna marca o dato particular?
- 6) ¿Cómo se crea un programa de fabricación? ¿Qué datos tiene?
- 7) ¿Cómo se crea un programa de fabricación extraordinario y cuáles son las diferencias con un programa de fabricación normal? ¿Cómo se reconoce que es extraordinario?
- 8) ¿Cómo se delega una orden de producción a la Papelera? ¿Cuáles son los pasos a seguir? ¿Dónde se registra que fue delegada? ¿Cómo se realiza el seguimiento de esa orden de producción?
- 9) ...

Puede observarse que durante la descripción del Escenario “Planificar la Producción” aparece la acción mencionada en el Párrafo 2, referida a priorizar. Aún no se conoce la importancia ni el detalle de esta sub-situación, pero se incorporan los episodios correspondientes.

Una vez descritos todos los Escenarios derivados de los verbos se retoma “Priorizar”. Como puede observarse en la Figura 127 que corresponde a la información disponible en el Párrafo 2, existen dos casos que determinan esta acción:

1. Cuando existe una urgencia de fabricación y
2. por el desperdicio de cartón corrugado durante la fabricación.

Esto requiere dos sub-Escenarios, como se puede observar en la Figura 128.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 18 – Derivación de Escenarios por Proximidad

<p>Priorizar Orden de Producción por Desperdicio Objetivo: Priorizar las órdenes de producción según su desperdicio Contexto: Ubicación Geográfica: Oficina de planificación de la producción Ubicación Temporal: lunes a viernes de 9 a 13 y de 14 a 18 Precondición: Debe haber órdenes de producción sin estar incluidas en programas de fabricación. Recursos: Orden de producción Actores: Oficial planificador Episodios:</p> <ul style="list-style-type: none">• El Oficial Planificador determina el desperdicio de materia prima de cada orden de producción.• Ordena las órdenes de producción según el desperdicio en forma decreciente <p>Excepciones</p>	<p>Priorizar Orden de Producción Urgente Objetivo: Priorizar órdenes de producción para asegurar su cumplimiento en el tiempo establecido Contexto: Ubicación Geográfica: Oficina de planificación de la producción Ubicación Temporal: lunes a viernes de 9 a 13 y de 14 a 18 Precondición: Debe haber una orden de producción con menos de 10 días para la entrega Recursos: Orden de producción, programa de fabricación, fecha de entrega (debe ser menos a 10 días) Actores: Oficial planificador Episodios:</p> <ul style="list-style-type: none">• El Oficial Planificador asigna el estado urgente a la orden de producción• busca el programa de fabricación más próximo• Determina en cual incluirla• Incluye la orden de producción <p>Excepciones</p>
--	--

Figura 128 – Ejemplo de Escenarios generados por proximidad (Caso SNor)

Por la relevancia de la situación, en la Figura 129 se modificó el Escenario de la Figura 127 para mostrar la inserción de los sub-Escenarios recién descriptos.

Planificar la Producción

Objetivo: Generar los [programas de fabricación](#) para una semana.
Contexto:
Ubicación Geográfica: [Oficina de planificación de la producción](#)
Ubicación Temporal: martes de 9 a 13 y de 14 a 18
Precondición: Debe haber una [orden de compra aprobada](#).
Recursos: copia de la [Orden de compra](#) (debe estar [aprobada](#)), información del [Depósito de Tamaño Fijo](#), información del [encargado de planta](#)
Actores: [Jefe de producción](#), [Oficial Planificador](#)
Episodios:

9. El [oficial planificador](#) analiza las [órdenes de compra](#)
10. Si la [orden de compra](#) tiene más de un tipo de [caja](#) entonces genera una [orden de producción](#) por cada [tipo](#) de [caja](#)
11. Estudia las [fallas](#) ocurrida durante la [fabricación](#)
12. Si una [orden de producción](#) es difícil de complementar con otra o se atrasa en demasía y el plazo de entrega se reduce a menos de 10 días entonces **PRIORIZAR ORDEN DE PRODUCCION URGENTE**
13. Procesa y ordena la información enviada por los [encargados de planta](#)
14. Procesa y ordena la información enviada por el [Depósito de Tamaño Fijo](#)
15. **PRIORIZAR ORDEN DE PRODUCCION POR DESPERDICIO**
16. Elabora 21 [programas de fabricación](#)

Excepciones:
Cuando una [orden de producción](#) no puede ser incorporada a ningún [programa de fabricación](#) entonces se la delega a la Papelera del Sudoeste.

Figura 129 – Ejemplo de Escenario Derivado por Proximidad modificado (Caso SNor)

Capítulo 19

Información Extemporánea en los procesos de requisitos

Resumen

La información extemporánea es un elemento que aparece en casi todos los procesos, pero casi ninguno la ha tomado en cuenta. Se refiere a aquella información que aparece espontáneamente durante alguna actividad de la IR, no se la espera y de la cual, se desconoce aún su relevancia. Este tipo de información tiene un alto riesgo de perderse. A continuación, se describe un mecanismo para gestionar este tipo de información dentro del Proceso de Requisitos basado en Escenarios, el cual fue presentado oportunamente en la tesis de maestría de la tesista, “Información Extemporánea en los Procesos de Requisitos”.

La Información Extemporánea (IE) [Kaplan09] [Kaplan14a] [Kaplan15] es aquella información que no es posible incorporar al modelo que se está construyendo. En contextos donde la elicitación es guiada por los modelos, esta información que no tiene cabida aún, puede generar un grave problema principalmente cuando es olvidada, se pierde o no se comprende correctamente en el momento de retomarla. Por ejemplo, durante una entrevista, tanto el entrevistado como el ingeniero/a de requisitos pueden apartarse del objetivo de la misma y abordar otros temas, en general el desencadenante es una pregunta o comentario. La información que aparece difícilmente sea útil para el objetivo de la entrevista y no pasará al modelo que se está construyendo. El riesgo de perderla consiste en que se puede perder la noción de su propia existencia. Es justamente en aquellas entrevistas no estructuradas donde se hace más notoria la perniciosa combinación de “información extemporánea - riesgo de perderla”. Cabe aclarar, que la mera aparición de este tipo de información no implica la necesidad de tratarla como información extemporánea, sino que se le debe agregar la percepción del riesgo a perderla. Algo similar puede ocurrir durante la lectura de un documento, pero sin lugar a dudas el espacio más frecuente de la IE es la comunicación oral. Tal es así, que un aspecto que incide en la aparición de IE es justamente la técnica de elicitación utilizada, principalmente aquellas donde participan personas (entrevistas, brainstorming, etc.). Otro aspecto que influye sobre la aparición es la expectativa del usuario para con el nuevo Sistema de Software, siendo directamente proporcional a la cantidad y tipo de IE que aparece. Cuando mayor es su expectativa, mayor cantidad de IE, particularmente información adelantada (IA), o sea aquella para la cual no se tiene construido aún el modelo que debe alojarla. Cuando este modelo existe, su incorporación es casi inmediata y se la denomina información tardía (IT). En la mayoría de los casos, es la IA la más compleja de tratar, pero la IT tiene el riesgo de contener información importante que no fue pensada ni modelada, teniendo como consecuencia la necesidad de tomar decisiones sobre parte de lo construido. Cabe destacar que usualmente la IE es útil cuando se encuentra en manos del ingeniero/a de requisitos y es su responsabilidad no perderla. También se debe aclarar que la existencia de las transcripciones de las entrevistas parece minimizar el riesgo de perderla, pero en realidad, genera un efecto inverso. Por un lado, no realizar un registro inmediato puede producir la pérdida de algunos datos del contexto, la cual es de gran utilidad para recuperarla. Por el otro lado, buscar IE en las

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 19 – Información Extemporánea en los procesos de requisitos

transcripciones requiere de un esfuerzo adicional con un alto riesgo de no identificarlas, o sea perderlas. Por tal motivo, el mejor momento para registrar una IE es específicamente, cuando aparece.

IE en los Procesos de Requisitos

Si bien la IE puede aparecer durante todo el proceso de construcción del software, la mayor incidencia se presenta durante la Ingeniería de Requisitos ya que en esta etapa es cuando se definen los servicios del nuevo Sistema de Software. Contemplar la IE en los procesos de requisitos mejora la completitud de los requisitos y su realidad. En la Figura 130 se representa cualitativamente la cantidad de IA y de IT a lo largo del Proceso de Requisitos. Puede observarse que, en la primera etapa del Proceso de Requisitos, aparece la mayor cantidad de IE, principalmente aquella del tipo IA. Esta cantidad, si bien es un indicador importante en sí mismo, no puede ser analizada de manera aislada. Se debe analizar cualitativamente, observando la combinación cantidad y consecuencias sobre la información elicitada. Por ejemplo, mucha cantidad de IE con una repercusión baja en la información existente, no afecta la consistencia del trabajo que se está realizando. En cambio, poca cantidad de IE con una repercusión alta, puede implicar hasta un cambio en la perspectiva de construcción utilizada hasta el momento.

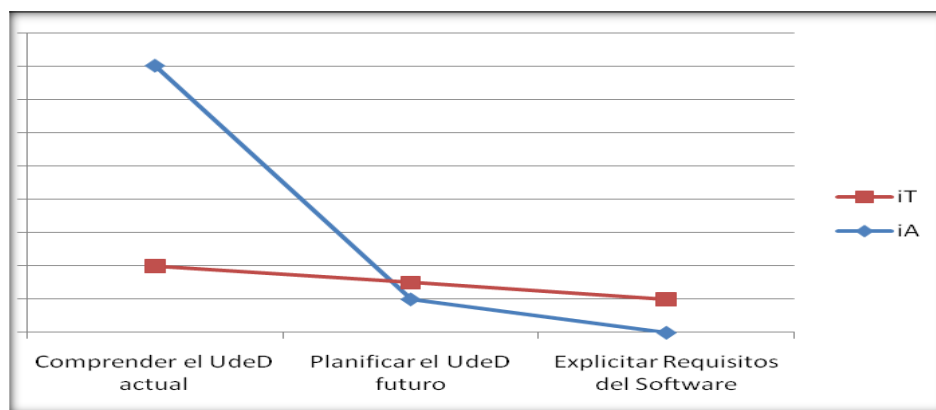


Figura 130– Cantidad de IE en las etapas del Proceso de Requisitos

Un aspecto de la Figura 130 a remarcar es la separación del tratamiento del UdeD, o sea, analizar el UdeD actual y el UdeD futuro por separado. Si bien es un tema en discusión dentro de la Ingeniería de Requisitos, paradójicamente es en la etapa Comprender el UdeD actual donde aparece la mayor cantidad de IA. La aparición tan temprana de estos aspectos no elicitados reduce significativamente el costo de tener que repensar objetivos

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 19 – Información Extemporánea en los procesos de requisitos

o modificar alguna perspectiva de lo modelado. Por el contrario, cuanto más tarde se detecte información con amplia repercusión en los modelos construidos, más alto es el costo de corrección de los modelos involucrados, llegando al extremo de requerir detener el proceso de construcción del software para re-analizar toda la información elicitada.

Información Adelantada (IA)

La preponderancia de las entrevistas sobre otras técnicas de elicitación en relación con la IA es notoria y visible pero no excluyente. Por ejemplo, la lectura de bibliografía del macrosistema (manuales de procedimientos, transcripciones de entrevistas, formularios, etc.) es una fuente de información muy utilizada y rica para modelar, pero suele tener algunas menciones, tal vez desperdigadas, de aspectos necesarios para construir la solución. La condición de desperdigada es relevante en este contexto ya que, si el documento tiene información relacionada con la solución, pero concentrada, meramente habrá que registrar este hecho para retomar la lectura del documento cuando sea el momento adecuado. Pero cuando está desperdigada el riesgo de omisión crece en forma inversa al tamaño de los fragmentos existentes. Otro ejemplo de aparición de IA se presenta al utilizar la técnica de observación para capturar información, el ingeniero/a de requisitos puede enfrentarse a un suceso claramente relacionado con un modelo a ser confeccionado más adelante, pero carece de la certeza de poder volverlo a observar cuando atienda ese modelo, haciendo necesario su registro adelantado. Asimismo, el ingeniero/a de requisitos puede tener ideas potencialmente muy apropiadas en relación con lo que escucha, observa y lee. Esto está relacionado con la memoria de corto plazo, la que le suele jugar malas pasadas a los creativos en cualquier actividad. Lamentablemente, muy poco tiempo después de haberse concebido una idea sólo se recuerda precisamente eso, que se tuvo una idea, pero no se recuerda exactamente cuál. Esto cae completamente en el contexto de la IA, aunque su origen no fue ni el cliente ni el usuario. Por otra parte, es muy difícil determinar en una etapa temprana, si la información elicitada o las ideas del ingeniero/a de requisitos son válidas; si contienen la raíz de lo que luego serán requisitos o sencillamente son aspiraciones no aplicables en el contexto futuro del Sistema de Software.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 19 – Información Extemporánea en los procesos de requisitos

La correcta manipulación de la IE debe afrontar el problema que el ingeniero/a de requisitos está tratando de concentrarse en su objetivo principal y esta información, particularmente la IA, lo perturba en su trabajo, de manera que se corre el riesgo de no hacer ni una cosa ni la otra. Por lo tanto, se debe minimizar esa perturbación y al mismo tiempo aprovechar la IA tanto como sea posible. El problema así visualizado es muy similar al recorrido de un grafo con bifurcaciones, lo que se hace en este caso es seguir el camino principal asegurándose poder retornar al camino alternativo cuando sea oportuno. Es decir, el ingeniero/a de requisitos debe conservarse lo más apegado posible a la actividad principal para lograr el objetivo deseado, pero al mismo tiempo debe encontrar una forma de recordar lo mínimo necesario para retornar al camino secundario posteriormente. Si bien la IE molesta al ingeniero/a de requisitos no se debe intentar eliminarla ya que su existencia puede ser un factor que asegure la calidad de los requisitos del software. La IA evidencia la presencia de muchas ideas que pueden enriquecer el conjunto de los requisitos de software.

La información adelantada aparece cuando:	
el usuario...	el ingeniero/a de requisitos...
describe algún problema al realizar una tarea.	recurre a su experiencia y conocimiento para encontrar solución a diferentes problemas, necesidades o expectativas de los usuarios
menciona necesidades insatisfechas.	
describe expectativas para el futuro.	recurre a su experiencia con otros sistemas de software similares para proponer funcionalidades alternativas
sabe que necesita o requiere del nuevo Sistema de Software.	

Tabla 32 - Principales situaciones donde aparece IA

La presencia de IA puede ser un hecho cuasi-anecdótico o ser un problema significativo en un proyecto de desarrollo de software (ver Tabla 32). La cantidad de información a registrar puede verse multiplicada por la cantidad de ingeniero/as de requisitos y de las fuentes de información involucradas en el proceso. Puede verse agravada por la existencia de múltiples perspectivas y por la complejidad del macrosistema.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 19 – Información Extemporánea en los procesos de requisitos

El registro semi estructurado que se presenta en este capítulo, pretende distraer lo menos posible la atención del ingeniero/a de requisitos asegurando el menor grado de distorsión de la línea principal de trabajo [Perls75], ofreciendo una forma simple de registrar esta información, con un marco homogéneo y ordenado y principalmente dejando la información en forma clara y visible para ser utilizada en el momento oportuno. Cabe aclarar que la IT también es un factor de perturbación, pero su efecto es de menor importancia ya que normalmente se reduce a forzar al ingeniero/a de requisitos a actualizar el modelo afectado y eventualmente algunos de los que le suceden, introduciendo un problema de versionado que no está relacionado con la evolución del UdeD o de la concepción del sistema, sino meramente con el mecanismo constructivo utilizado.

La IA está asociada a dos características que combinadas conforman su perfil en el contexto de la aplicación. Una es cuantitativa y está relacionada con la expectativa del cliente-usuario mientras que la otra es cualitativa y tiene que ver con el rol del entrevistado. Como se muestra en la Figura 131, la expectativa de los clientes-usuarios es determinante para la aparición de IA y es el factor más significativo referido a la cantidad de IA que puede aparecer. Cuando se entrevista a personas que esperan/necesitan algún servicio del nuevo Sistema de Software se obtiene mayor cantidad de IA, mientras que en el otro extremo, cuando el entrevistado no tiene ninguna expectativa con respecto al nuevo software y, en algunos casos, ni siquiera sabe que puede esperar del mismo, la aparición de IA es prácticamente nula.

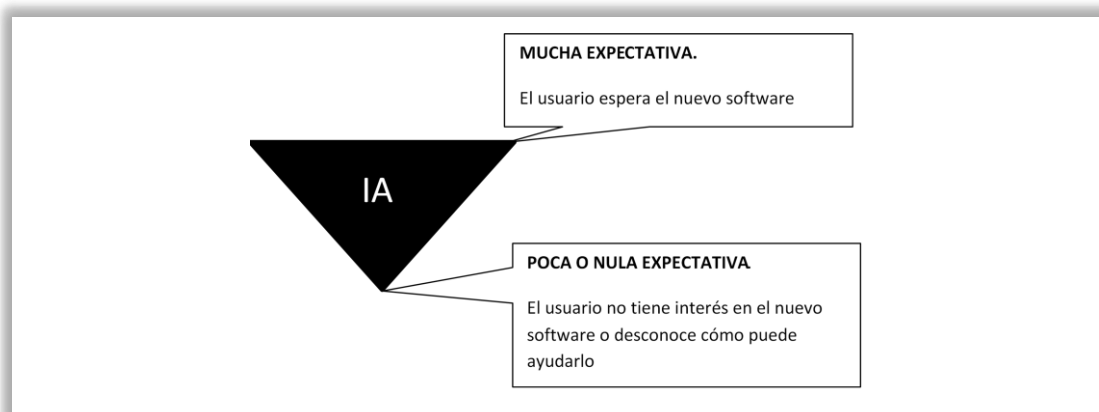


Figura 131 –IA según la expectativa del cliente-usuario

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 19 – Información Extemporánea en los procesos de requisitos

Se debe considerar también que la cantidad de IA aumenta considerablemente según la cantidad de cambios que se planifica realizar en el proceso del negocio. En la Tabla 33 se enumeran algunos factores que pueden desencadenar IA.

Factores Internos	Factores Externos
Incorporación de Gestión de Calidad	Cambios económicos y políticos que afectan el negocio
Tercerización de actividades	Cambios en las preferencias de los clientes
Apertura de nuevas líneas de negocio	Legislación nueva
Cambios en la tecnología de producción	Cambios socio-culturales que afectan a la organización
Aumento en los volúmenes de ventas	
Cambios en los objetivos organizacionales	
Cambios en los productos o servicios	

Tabla 33 - Principales factores de cambio que influyen en la aparición de IA

Información Tardía (IT)

La IT se presenta de manera similar a la IA. Sucede durante la elicitación cuando un cliente-usuario especifica o corrige información existente, cuando el ingeniero/a de requisitos descubre alguna inconsistencia mientras elicitación o cuando descubre una nueva interpretación de algún aspecto interpretado incorrectamente. Esta información es, en general, una aclaración, una explicación, una observación o una modificación a la propia elicitación, pero que afecta lo modelado. Justamente, la existencia del modelo es lo que la caracteriza como información tardía. Que este modelo ya haya sido construido disminuye sensiblemente el nivel de preocupación del ingeniero/a de requisitos respecto a la posibilidad de perderla u olvidarla, ya que su incorporación puede ser en un plazo más breve.

Es importante marcar una diferencia sustancial entre la IT y la IA. La IT nunca cambiará su tipo ya que el modelo que la debe contener existe en el momento de su aparición y seguirá existiendo durante todo el Proceso de Requisitos. No sucede lo mismo con la IA ya que su condición puede cambiar desde el momento de su aparición hasta que es

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 19 – Información Extemporánea en los procesos de requisitos

analizada o incluida. Puede ser analizada sin que el modelo exista y mantener su condición de adelantada. O puede que el modelo ya se halla construido y deba ser tratada como IT. Cabe destacar que el tratamiento de la IE no depende sólo de la existencia del modelo donde será incluida, sino que también juega un rol central la granularidad de la información que contiene.

IE en el Proceso de Requisitos basado en Escenarios

Como se mencionó, es la IA la más frecuente y más perturbadora, principalmente aquella que afecta a los EF. También puede serlo aquella que afecta a la ERS, pero su aparición suele ser mínima o nula. En la Figura 132 se puede observar que durante la Comprensión del UdeD actual aparece la mayor cantidad de IE y en la mayoría de los casos, la más significativa. Durante la Planificación del UdeD futuro se construyen los EF, modelo central del Proceso de Requisitos, y el LEL_R que desambigua los EF y la ERS. Se espera poca IE ya que se está en el núcleo del Proceso y la IA ya fue incorporada y desde ahora, casi toda la IE será IT. La IA que puede aparecer es mínima, posiblemente relacionada con prioridades de fragmentos del futuro sistema o indicaciones que se podrán transformar en RNF.

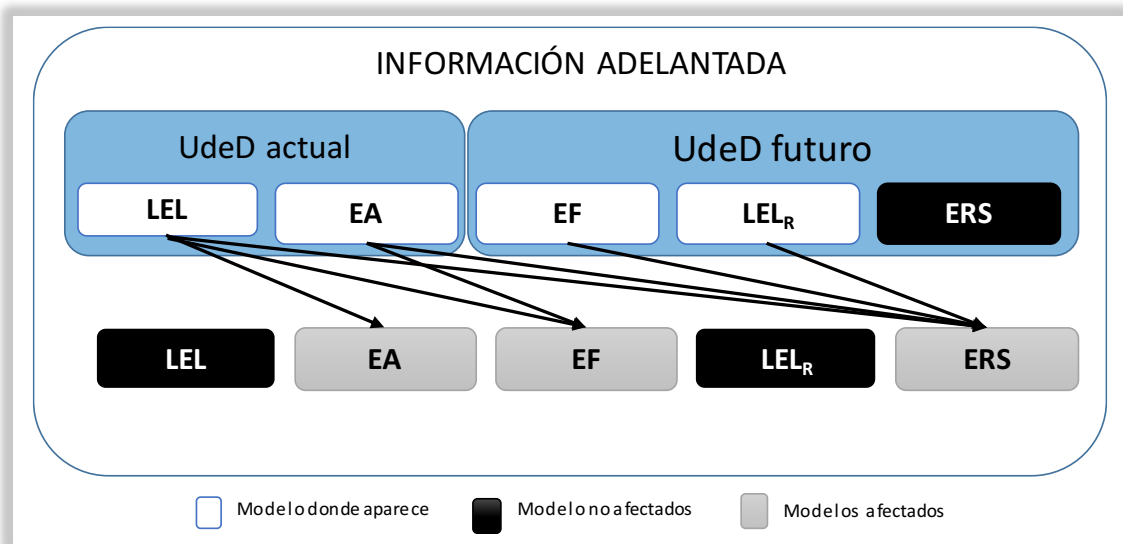


Figura 132 – Información Adelantada en el Proceso de Requisitos

La Figura 133 describe en qué modelo puede aparecer IT y a qué modelos puede afectar. Se puede observar la gran incidencia que tiene este tipo de información en todo el proceso. Ya que el LEL_R se construye con los EF completos, puede aparecer IT referida principalmente a los EF. También puede suceder que se detecte algún error u omisión del

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 19 – Información Extemporánea en los procesos de requisitos

LEL o en los EA. Si se detecta IT puede ser necesario revisar los textos donde fue utilizada para evitar inconsistencias. Por lo tanto, esta IT podría estar dispersa.

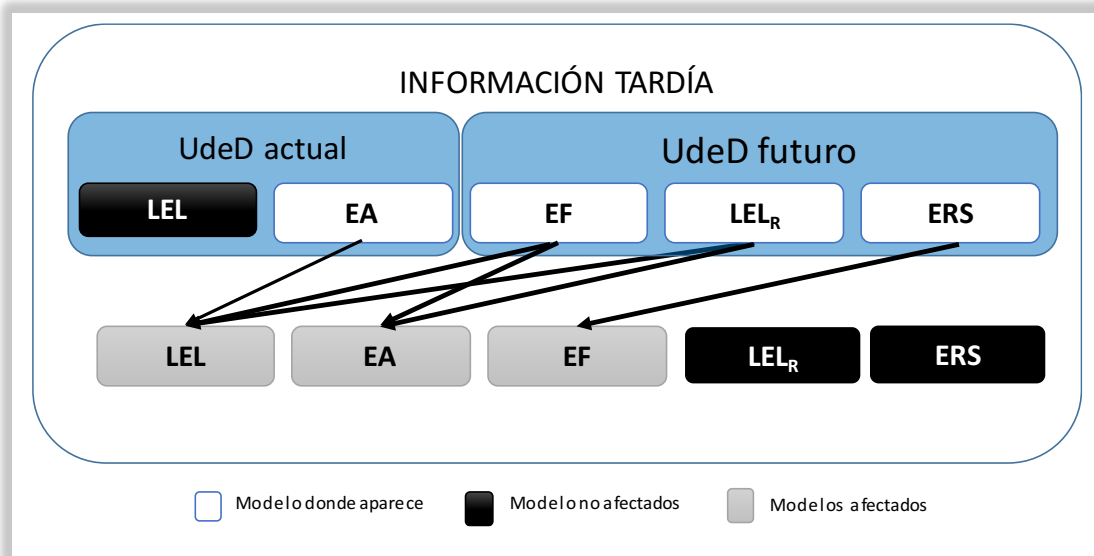


Figura 133 – Información Tardía

En la tercera etapa de la estrategia se construye la ERS. Sólo se puede esperar alguna IT para algún EF. Este tipo de IE puede ser generada durante la extracción de los requisitos de software debido a la detección de alguna omisión o inconsistencia en el EF origen. De existir IT de mucha importancia, se deberán revisar todos los modelos construidos.

A continuación, se presentan algunos ejemplos de obtención de IA durante el Proceso de Requisitos basado en Escenarios. En la Figura 134 se ejemplifica la aparición de IA proveniente de un usuario en el momento de elicitar información del léxico, o sea durante la construcción del LEL. Esta información deberá ser tomada en cuenta en el momento de construir los EF.

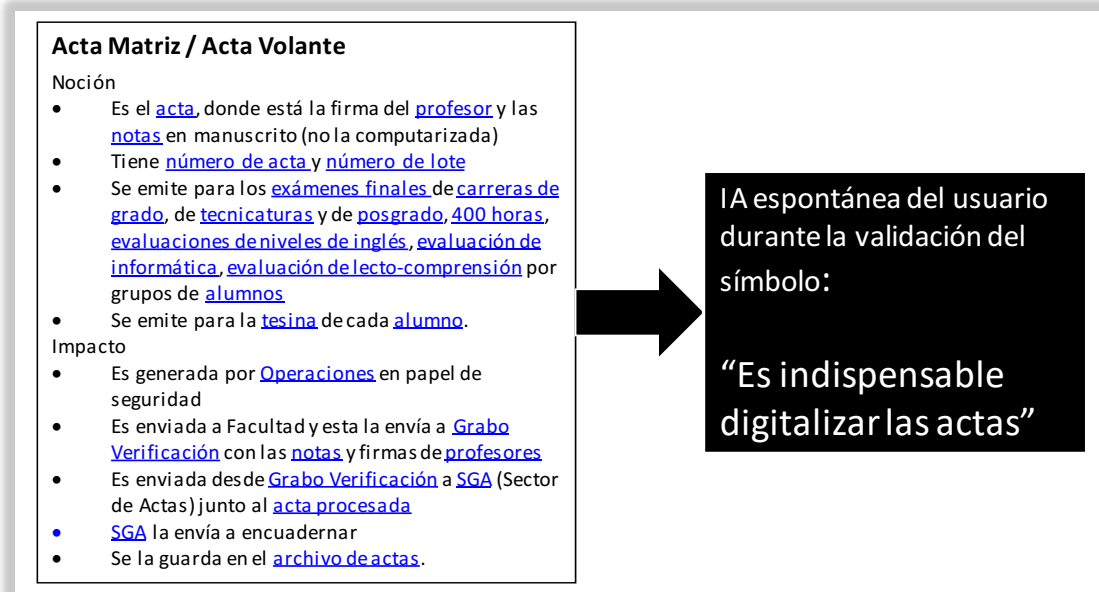


Figura 134 - IA surgida durante la Construcción del LEL (Caso SGA)

En la Figura 135 se muestra un ejemplo de IA que surge durante la construcción de un EA y que, igual que la IA de la Figura 134, deberán ser tenidas en cuenta cuando se construyan los EF.

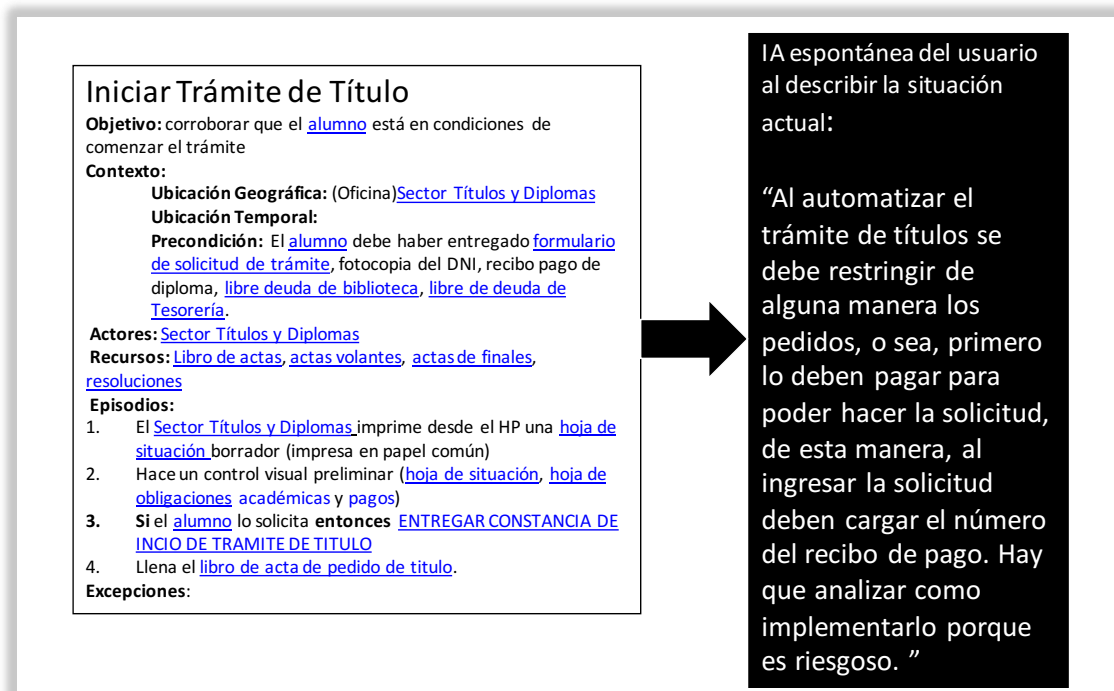


Figura 135 – Ejemplo de IA surgida durante la Construcción de un EA (Caso SGA)

En la Figura 136 se presenta otra IA del mismo Escenario de la Figura 135. Se puede observar la IA y el EF correspondiente. Esta IA debe ser fragmentada antes de ser incluida en los EF. En el ejemplo se presenta el resultado final del Escenario destino.

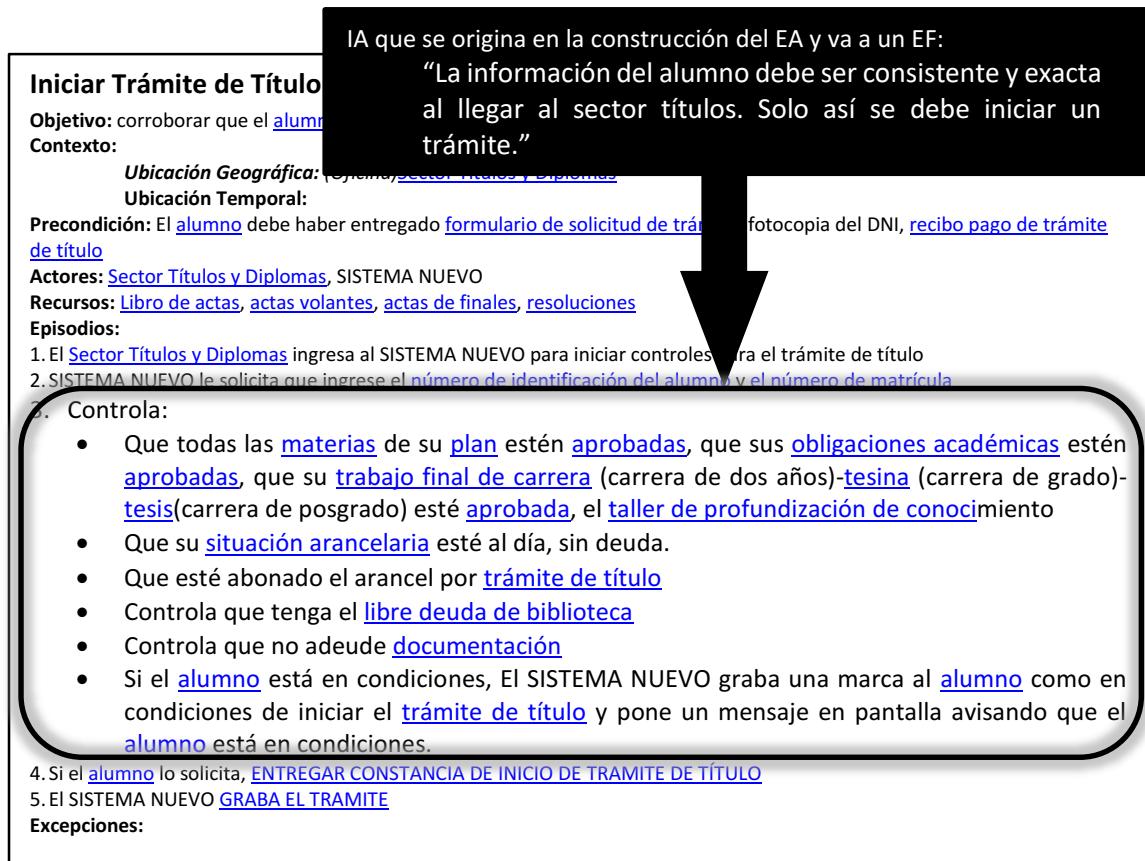


Figura 136 – Ejemplo de EF generado a partir de IA (Caso SGA)

La Figura 137 se puede observar las diferentes interpretaciones que existen de una IA. En la Figura 138 se muestra un ejemplo de un RNF que se detecta a partir de una IA que aparece muy temprano en el Proceso de Requisitos, durante el UdeD actual. Esta IE debió haber sido incorporada a los EF como una restricción, pero puede suceder que por la dispersión de esta IE no se haya incorporado correctamente.

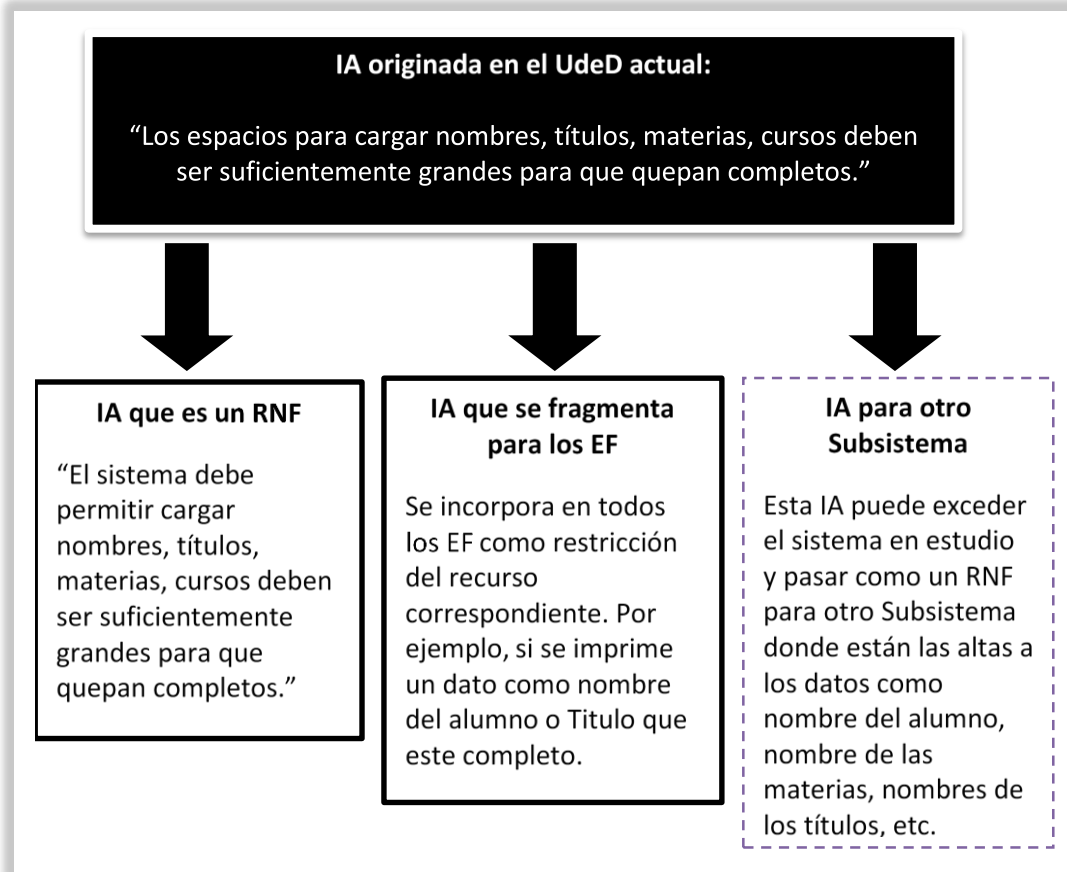


Figura 137 – IA como restricción de un Recurso de un EF y como RNF

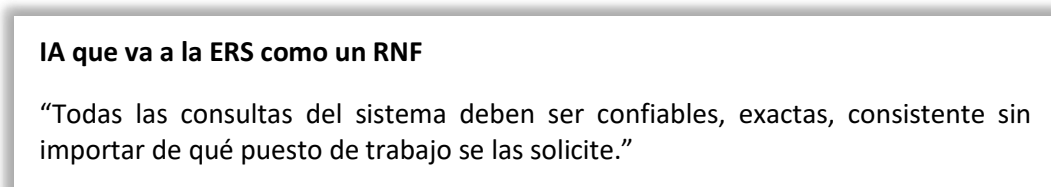


Figura 138 - RNF detectado desde una IA

Ficha de Información Extemporánea (FIE)

A efectos de incorporar la IE a los Proceso de Requisitos, se presenta a continuación como debe estar compuesta una FIE. Si bien la FIE se la utiliza en el entorno de un Proceso de Requisitos en particular, puede ser adaptada a cualquier proceso con sólo realizar una revisión de los datos a registrar para adecuarla a las particularidades de dicho proceso, pero en los casos considerados estas modificaciones han sido mínimas. Para cumplir con el objetivo de adaptabilidad de la ficha a todos los procesos, se presenta a continuación la información necesaria que la misma debe contener. También se incluyen algunas observaciones a tener en cuenta.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 19 – Información Extemporánea en los procesos de requisitos

Los datos que debe contener una FIE puede ser dividida en dos partes. Esta división se basa en un aspecto temporal referido a su uso. En un primer momento se deben completar los datos de apertura de la IE. Esto sucede en el mismo momento en el cual la IE aparece. Un tiempo después, cuando la IE es incluida en algún modelo o descartada, la ficha debe ser cerrada. Como se verá más adelante, algunas fichas se cierran parcialmente, por lo que la operación de cierre puede ocurrir en distintos momentos como sucede cuando una IE debe dispersarse.

A continuación se presenta en la Tabla 34 los datos que deben ser tenidos en cuenta a la hora de generar una FIE independientemente del Proceso de Requisitos que se utilice:

Ítem	Uso dentro del proceso	Valores posibles	Tipo
DATOS DE APERTURA			
Proyecto	Reconocer las fichas que deben ser analizadas juntas	Nombre del proyecto, subsistema o módulo.	A
Identificación de la FIE	Identifica cada ficha unívocamente	Número secuencial + [número de fragmentación]	A
Fecha	Indica el momento de la aparición	Fecha de aparición de la IE	A
Tipo de Origen	Identifica si es información proviene del UdeD o del equipo de desarrollo	Ingeniero/a de requisitos - Fuente de Información + Número de FIE si viene de una fragmentación	R
Nombre del origen	Permite retomar al origen de la IE	Nombre del ingeniero/a de requisitos o nombre/identificación de la Fuente de Información. número de identificación de la fragmentación, si corresponde	R
Tipo IE	Indica si existía el modelo donde debe ser incluida la IE en el momento de su aparición	IT - IA	P
Nivel estimado de importancia	Indica la posible repercusión de la IE en el resto de la información ya elicitada.	Alto - Medio - Bajo.	P

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 19 – Información Extemporánea en los procesos de requisitos

Tratamiento urgente	Indica que debe ser analizada lo más tempranamente posible en el proceso	Si-No	P
Modelo en el que se sugiere incluir	Facilita la inclusión de la IE	Nombre del modelo donde supone el ingeniero/a de requisitos, debe ser incorporado.	P
Ítem del modelo en el que se sugiere incluir	Facilita la inclusión de la IE junto al ítem Modelo	Nombre o número de ítem dentro del modelo.	P
Dispersión	Se la utiliza para alertar que debe ser incluida en varios ítems del modelo.	Si-No	P
Descripción de la IE	Es la información central, lo que debe ser analizado e incluido.	Texto libre.	P
DATOS DE CIERRE			
Ingeniero/a de Requisitos responsable	Identifica responsable de impactar la IE	Nombre el ingeniero/a de requisitos	R
Fecha	Es la fecha en la cual se incluyó la IE en algún modelo o se fragmentó	Fecha de la inclusión en el modelo.	R
Estado	Indica si la FIE fue aceptada o descartada. En el caso de ser descartada el resto de información no es relevante pero se sugiere completar el ítem Comentario con la justificación de su rechazo.	Aceptada-Rechazada	P
Modelo en el cual se incluyó	Indica donde fue impactada la IE.	Nombre del modelo/s	R
Ítem del modelo en el cual se incluyó	Complementa la información del ítem anterior.	Nombre o identificación del ítem del punto anterior	R
Comentario	Complementa información de cierre.	Texto libre	P

Donde Tipo es: A-dministración, R-astreabilidad y P-roceso.

Tabla 34 – Datos requeridos para la FIE

Los ítems del tipo “R” son específicos para la rastreabilidad. Se espera que, en modelos integrados, estos datos sean exportados/factorizados a los modelos de rastros. En este

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 19 – Información Extemporánea en los procesos de requisitos

caso particular, donde se presenta una ficha que es independiente del proceso en donde se la utilice, es necesario incorporar estos datos para que ayuden a la gestión de las fichas. A continuación, se describen algunos comentarios particulares sobre algunos ítems significativos que pueden ayudar a su posterior tratamiento:

Nivel estimado de importancia. A la hora de incorporar la IE en el modelo correcto, la granularidad se traduce como la presunción del ingeniero/a de requisitos con respecto a la repercusión que produce la IE en el resto de la información elicitada. Este ítem puede ser alto, medio o bajo.

La importancia alta puede estar relacionado a una mala comprensión del Objetivo General del Sistema o a objetivos específicos no identificados. Pero también puede suceder que un usuario cambie su punto de vista acerca del proyecto mientras se analizan diferentes alternativas. Esto último se debe a que los procesos de requisitos actúan, en muchos casos, como un mecanismo de aprendizaje y descubrimiento sobre las posibilidades que puede ofrecer un Sistema de Software. La IE con alto impacto requiere que, siempre que sea posible, sea analizada antes de comenzar con el modelo que resulte afectado.

La importancia media se refiere a información relacionada con el Objetivo General del Sistema o sub-objetivos específicos pero que no afectan la solución, sino que son más operativos. Este tipo de IE repercute en distintos lugares del modelo y su complejidad radica en que deben estar siempre visibles mientras se modela, ya que deben ser consultadas con regularidad durante la construcción de todo el modelo.

La importancia baja no trae mayor complejidad debido a que se refiere, en la mayoría de los casos, a detalles o particularidades que se deben incorporar en el modelo y esta inclusión es más efectiva si se realiza una vez completo el modelo en cuestión.

En los casos de importancia alta y en algunos casos de importancia media, es elevada la probabilidad de que se presente fragmentación de la IE.

Modelo en el que se sugiere incluir. Es altamente probable que el ingeniero/a de requisitos sepa, en el momento en el cual aparece una IE, donde se deberá incluir. Cuando un ingeniero/a de requisitos tiene experiencia en el Proceso de Requisitos utilizado, raramente se equivocará en este ítem. Pero si se equivoca, el efecto puede provocar trabajos innecesarios, ya que se estará induciendo a un trabajo erróneo.

Dispersión. Existe IE que será incluida en varios lugares del o de los modelos. Este ítem indica la presunción que tiene el ingeniero/a de requisitos acerca de que esa información se deba incluir en varios ítems.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 19 – Información Extemporánea en los procesos de requisitos

Descripción de la IE. Este ítem es un elemento clave. Su interpretación condiciona su correcta utilización. Por tal motivo se sugiere que la misma sea breve, concreta y precisa. En caso de existir alguna documentación, un Sistema de Software u otro registro relacionado con la IE, debe quedar referenciado en forma clara.

Si bien se espera que la descripción se exprese con la claridad suficiente para comprenderla cuando sea necesario retomarlo, puede suceder que esto no se logre. La posibilidad de contar con descripciones poco útiles es una debilidad que no se ha presentado en los casos utilizados, pero no se puede garantizar que no ocurra nunca. La calibración de la importancia de esta debilidad potencial podrá realizarse si en futuras aplicaciones se descubren situaciones en las cuales se haga presente y entonces en ese momento se podrán definir métricas o precisar heurísticas para controlar dicho problema. En resumen, se está en presencia de un problema potencial acerca del cual no se tiene ninguna certeza de su ocurrencia.

Registro, Tratamiento y Cierre de la FIE

En la Tabla 34 se detalló toda la información necesaria que debe tener una FIE, ahora es necesario analizar cómo se debe incorporar su tratamiento en un Proceso de Requisitos, en este caso en particular, al Proceso de Requisitos basado en Escenarios. Se debe tener en cuenta que existen dos momentos:

- 1) El registro y
- 2) el tratamiento y cierre.

El **registro** de la IE se debe realizar exactamente en el momento en el cual aparece. A tal efecto, se debe utilizar la parte de la FIE con los datos de apertura (ver Figura 139). Cabe recordar que la apertura de la FIE se puede realizar en cualquier momento del Proceso de Requisitos. Con el objetivo de distraer lo menos posible la atención del ingeniero/a de requisitos, es que sólo algunos pocos datos son obligatorios. El resto de los datos pueden ser completados en otros momentos. También se debe tener en cuenta que varios de ellos pueden ser incorporados de manera automática o previa a la elicitación, como es el caso del nombre del proyecto, la identificación de FIE, el nombre del ingeniero, etc. Otra forma de resguardar la atención del ingeniero/a de requisitos fue el diseño de la ficha, donde algunos componentes pueden ser completados seleccionando de una lista de ítems,

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 19 – Información Extemporánea en los procesos de requisitos

dejando la escritura sólo para la Descripción. Eventualmente, se podrá escribir también en el comentario, sólo de ser necesario.

Con la IE registrada se puede dar lugar al *tratamiento* de la misma (ver Figura 143). Cabe destacar que el tratamiento induce al cierre, ya que una vez decidido el modelo donde se debe incluir, el ítem y la forma de hacerlo, es necesario registrar esta información en el *cierre* (ver Figura 140). Durante el tratamiento de la IE se debe analizar y determinar el modelo en el cual debe ser incorporada. La inclusión de IT es en la mayoría de los casos, incorporar dicha información en el modelo correspondiente. La IA tiene algunas particularidades. Excepto que sea para los EF o la ERS, se puede realizar durante la construcción del modelo o una vez completo. En el caso de los EF, el momento de analizar la IE es específicamente cuando se construye la primera versión de los EF. En ese momento se debe identificar la granularidad y la dispersión de la misma. Con respecto a la granularidad, cuando esta es grande, el tratamiento es la **fragmentación** de la IE, donde se divide en varias IE. Esto determina que se deban crear varias FIE. En el segundo caso, la **dispersión**, la IE afecta a varias situaciones, por lo tanto, esa IE debe ser identificada para ser tenida en cuenta durante toda la construcción de los EF.

FICHA DE INFORMACION EXTEMPORÁNEA	
Datos de Apertura	
	Identificación FIE:
*Proyecto:	*Fecha: .../.../....
*Origen: <i>FI – Ing.Req.</i>	Nombre del Origen:
*Tipo IE: <i>IA – IT</i>	*Nivel de impacto: <i>Alto – Medio - Bajo</i>
Tratamiento urgente: <i>Si – No</i>	Dispersión: <i>Si - No</i>
Incluir en el Modelo: Ítem:	
*Descripción de la IE:	
.....	
.....	
.....	
Comentario:	

Figura 139 - Datos de Apertura de la FIE

FICHA DE INFORMACIÓN EXTEMPORÁNEA	
Datos de Cierre	
Identificación FIE:/.....	
*Fecha:/..../....	*Responsable:
*Estado de la FIE: <i>Aceptada - Descartada</i>	
*Modelo en el cuál se incluyó:	Ítem:
Comentario:	

Figura 140 - Datos de Cierre de la FIE

La Figura 141 muestra un ejemplo de fragmentación, donde la Identificación FIE: 6 se cerró y se abrió la Identificación FIE: 9, 10 y 11. En cada una de las FIE nuevas el ítem Nombre de Origen fue completado indicando que proviene de la Identificación FIE: 6.

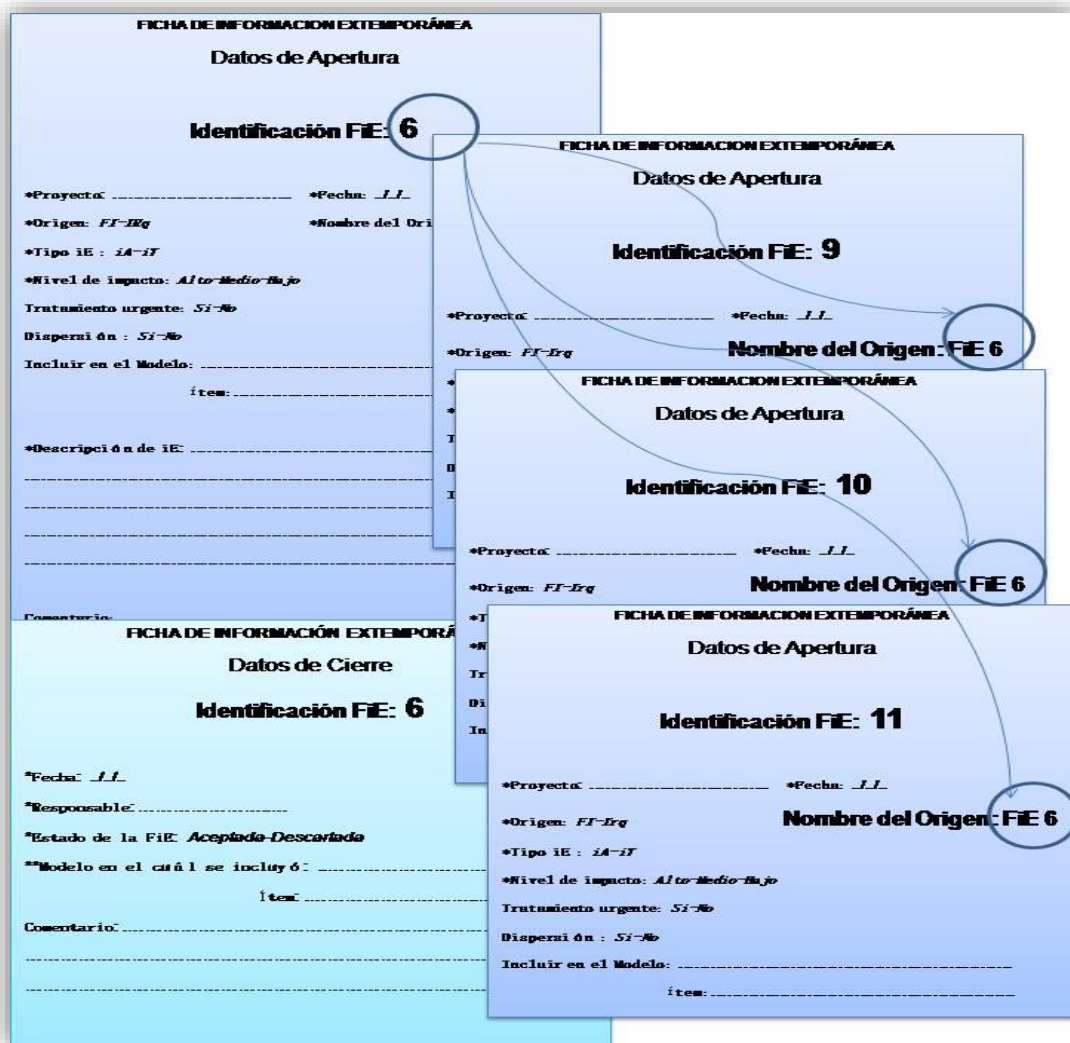


Figura 141 – Ejemplo de Fragmentación de IE

Cuando la FIE tiene dispersión se crean tantos cierres como inclusiones diferentes hubo de la misma IE. En la Figura 142 se muestra la Identificación FIE: 6 con dos cierres, Identificación FIE: 6/1 e Identificación FIE: 6/2.

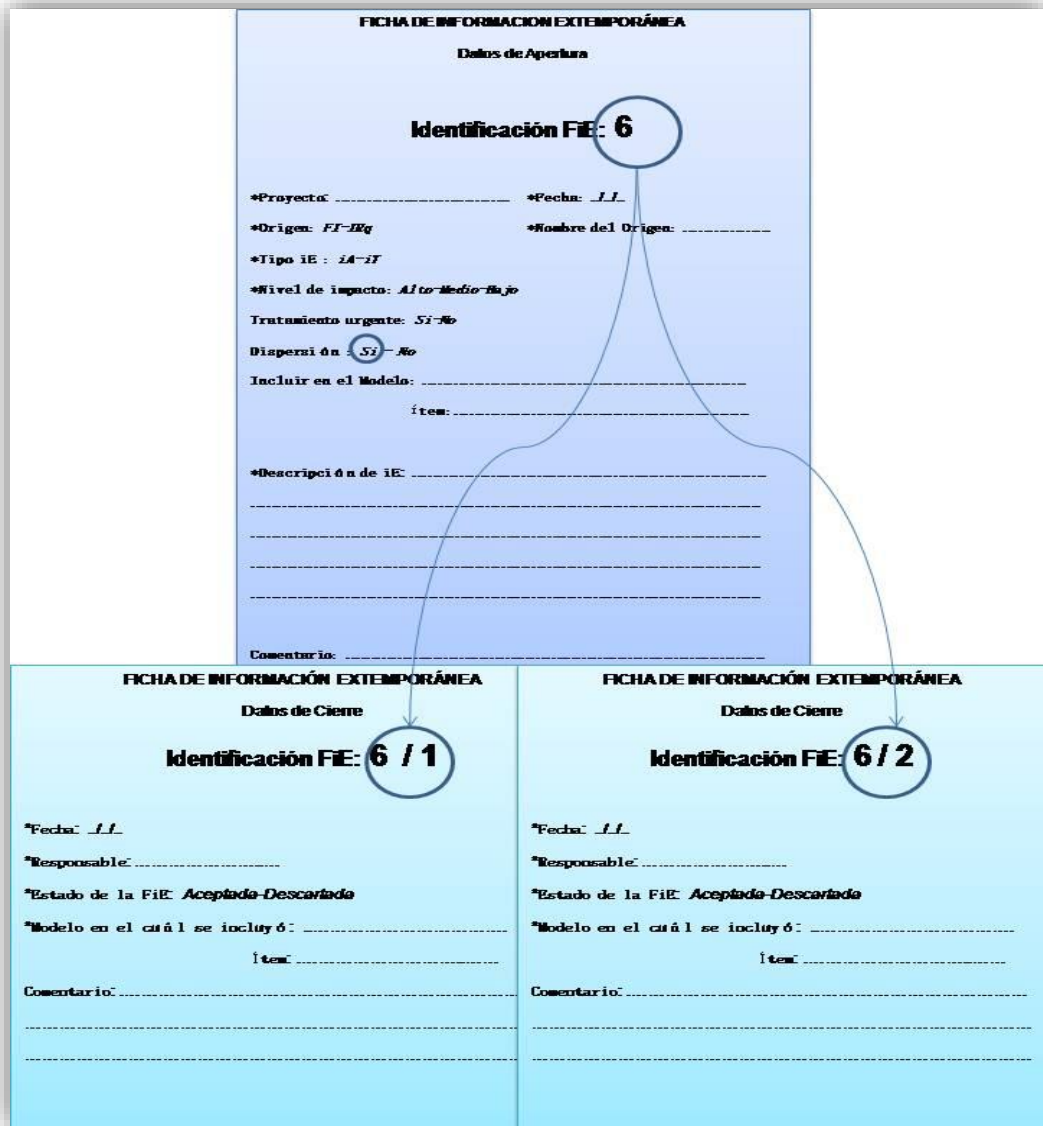


Figura 142 – Ejemplo de Dispersión de IE

Heurística para el tratamiento de la IA:

- 1) Antes de comenzar con el modelo de la solución, separar todas las FIE identificadas como IA.
- 2) Toda la IA que esté contenida en el Objetivo General del Sistema, debe ser estudiada, refinada e incorporada al conocimiento elicitado. Si ésta IA no está contenida en el Objetivo General del Sistema se sugiere, antes de continuar, revisar la pertinencia de la misma y analizar qué produjo esta mirada no contemplada en el objetivo.
- 3) Las FIE que fueron totalmente consolidadas con el conocimiento elicitado, deben ser cerradas. Aquellas que fueron incluidas parcialmente, deben quedar disponibles para ser consultadas durante todo el proceso de construcción del modelo.
- 4) La IA de importancia baja debe ser modificadas a IT para ser incluida una vez finalizado el modelo.
- 5) Las FIE identificadas como de importancia alta y media deben ser estudiadas observando la posibilidad de fragmentación.
- 6) Las FIE identificadas como posible de dispersión deben estar disponibles durante toda la construcción del modelo. Pero aun las que no fueron identificadas como tal, la posibilidad de dispersión debe estar siempre presente.
- 7) (Fin)

Figura 143 – Heurística de Tratamiento de la IA

Una vez que la IE es incorporada al modelo, se debe **cerrar** la FIE (ver **Figura 140** y **Figura 144**) para sacarla de los pendientes del proceso. Este cierre indica, en primer lugar, que la IE fue aceptada o descartada. Cuando IE es aceptada, se debe registrar el modelo y el ítem donde se incluyó. En el caso de ser descartada es conveniente aclarar el motivo de su rechazo.

Cabe destacar que puede aparecer IE que no se desee incluir en la versión del Sistema de Software que se va a construir. En este caso se debe descartar la IE, ya que no será utilizada, y se deberá exportar a un sitio de documentación establecido para una nueva versión del software, con el objetivo de ser tratada oportunamente, como podría suceder en un proceso iterativo o en espiral.

Heurística para el Cierre

- 1) Por cada FIE fragmentada o incorporada totalmente a un modelo se debe realizar un cierre de la ficha.
- 2) La IE con dispersión provoca un cierre por cada inclusión. La FIE original debe quedar pendiente durante todo el proceso por si aparece alguna dispersión nueva.
- 3) Cuando la IE ya no puede ser incluida en ningún modelo, se deben archivar. Dejar sólo aquellas fichas que aún tienen probabilidad de ser incorporadas.
- 4) La IE puede quedar incluida en el modelo o descartada. Existe un caso particular de la IE descartada que se presenta cuando la IE es para otra versión del Sistema de Software. En este caso incluir un comentario explicando este caso particular.
- 5) (Fin)

Figura 144 – Heurística de Cierre de la FIE

Es probable que algunas de estas FIE deban ser descartadas. La identificación de FIE que están fuera del estudio del macrosistema es un buen indicador ya que se estaría estableciendo algunos de los límites del sistema. Es un caso similar al de una plantación de girasol en un cuadrado determinado de tierra, en el cual puedo reconocer que se ha utilizado el máximo de las posibilidades cuando algunas de las plantas crecen más allá de sus límites, o sea ante el reconocimiento de que algunas semillas se fueron del terreno.

El Proceso de Requisitos se enriquece cuando el ingeniero/a de requisitos trata a la IE como parte integral de la elicitación de información y se fortalece con la existencia de mecanismos de resguardo y tratamiento adecuados para cada proceso. De esta manera, el ingeniero/a de requisitos siente la presencia de IE como un beneficio.

Capítulo 20

Conclusiones

CONSIDERACIONES FINALES

Como ya se mencionó, el aporte de la presente tesis se puede sintetizar en tres puntos. En primer lugar, el mismo *Proceso de Requisitos basado en Escenarios* en el cual la tesista ha tenido diferentes tipos de intervención durante más de dos décadas (ver Parte 2). En segundo lugar, el estudio empírico realizado al Proceso de Requisitos en general y en particular a los modelos LEL y Escenarios (ver Parte 3). Los problemas centrales detectados están relacionados con la completitud y la consistencia. Finalmente, se han propuesto un conjunto de agregados y mejoras que aumentan significativamente la calidad de los modelos y de todo el proceso (ver Parte 4).

Para garantizar la calidad del Proceso de Requisitos, es necesario cuidar la calidad de cada modelo construido. En este sentido se ha analizado tanto el LEL como los Escenarios y luego, las vinculaciones entre ellos. Como resultado del estudio realizado se han propuesto *nuevas heurísticas* de elicitación, modelado y análisis para el LEL y para los Escenarios. En todos los casos se ha elegido un mecanismo por *proximidad*, el cual permite adquirir nuevo conocimiento a partir de otro previamente incorporado, de una manera incremental. También se ha propuesto una nueva heurística de derivación de EA a partir de LEL y mejoras en la construcción de la primera versión de los EF a partir del LEL y de los EA. Cabe destacar que las heurísticas propuestas sugieren trabajar principalmente con documentos organizacionales, siempre que existan. Además, se propone la utilización de las transcripciones de todas las entrevistas realizadas ya que permiten aprovechar mejor la información que contienen y mejoran la rastreabilidad.

Puede observarse que en todas las heurísticas propuestas se han *eliminado las listas iniciales*, las cuales han sido una característica en las heurísticas existentes. Estas listas se construyen muy tempranamente en el proceso, cuando aún no existe suficiente conocimiento del dominio y esta es una desventaja importante ya que no permite alertar al ingeniero/a de requisitos cuando el camino inicial no es el correcto. La elección de una fuente de información que no sea la “ideal”, cuando es la primera fuente consultada, puede resultar más perjudicial que cuando ya comenzó el proceso. Esto se debe a que las listas, una vez generadas son poco mejoradas durante el proceso. Es probable que el mismo proceso de construcción de los modelos posteriores corrija estos desvíos, pero existe el importante riesgo de propagar involuntariamente errores a lo largo de todo el Proceso de Requisitos. Una ERS de baja calidad afectará la credibilidad del equipo de

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 20 - Conclusiones

desarrollo, y también generará retrasos e insatisfacción en las entregas como un aumento en el costo. Cabe aclarar que las listas pueden ser muy provechosas cuando son utilizadas adecuadamente.

El estudio empírico realizado se sustentó en la realización de casos profesionales de la autora de la presente tesis y de la revisión de centenares de casos de estudio siguiendo esta estrategia, realizados por alumnos de grado en las universidades UTN, UNLaM y UNO como en diferentes cursos de posgrado en UTN y UNLaM.

El Proceso de Requisitos existente tiene una gran omisión en cuanto a la información que maneja, relacionada con las *jerarquías conceptuales* y los *puntos de vista*. Ambos elementos han sido incorporados al LEL, luego a los Escenarios para finalmente estar representados en los requisitos del software. Las heurísticas antes mencionadas incorporan esta información. En la presente tesis se ha propuesto el modelado de esta información para hacerla visible y apta para la validación. Las jerarquías permiten refinar el conocimiento existente en el dominio, mientras que los puntos de vista permiten conocer cómo es realmente el contexto. Estas incorporaciones han generado modelos mucho más completos y requisitos del software más claros y correctos. La incorporación de las jerarquías puede evitar, por ejemplo, la presencia de requisitos ambiguos, como cuando el mismo hace referencia a un genérico y debe hacerlo a una especialización o viceversa. La incorporación de los puntos de vista puede evitar, por ejemplo, requisitos incorrectos como ser implementar en el software un flujo de acción marcado por el “deber ser” que no resulta operativo y que no fue percibido oportunamente. La detección y análisis de los puntos de vista permite que el ingeniero/a de requisitos junto a los usuarios-clientes definan la mejor solución para la organización.

También se ha incorporado un nuevo glosario que surge de la evolución del LEL en el contexto futuro y que se lo ha denominado *LEL de Requisitos (LEL_R)*. Este modelo representa la evolución “aparente” del vocabulario para describir el proceso del negocio con el sistema de software incluido. Este glosario del UdeD futuro es indispensable para reducir a su máxima expresión la ambigüedad de los EF y de la ERS. Se ha propuesto la elicitación, modelado y análisis del mismo, y se ha planteado una heurística de construcción. También se ha descrito como se debe incorporar el LEL_R durante la construcción de los EF.

Finalmente, se ha incorporado al Proceso de Requisitos la *información extemporánea (IE)*. Este trabajo corresponde a la tesis de maestría de la autora de la presente tesis. La

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Capítulo 20 - Conclusiones

La importancia de este mecanismo es complementar el proceso de requisitos con aquella información que aparece cuando no se la espera pero que de no ser incluida oportunamente puede generar una importante omisión de información que afecte directamente a los requisitos del software. La IE permite asegurar la completitud de la información al incorporar el mecanismo al propio proceso de requisitos.

TRABAJO FUTURO

Se enuncia a continuación una serie de temas tratados en la presente tesis que quedan abiertos para futuras investigaciones:

- **Elicitación:** mejorar la detección de los usuarios indispensables para garantizar que la IR cuente con toda la información necesaria. Se ha trabajado en la generación de nodos cuya distancia es el conocimiento de cada usuario. Esta información se obtiene a partir de cuestionarios. Este mecanismo no se ha probado aún.
- **Validación de Escenarios:** es necesario mejorar la validación de los escenarios, principalmente de los EF. Se está trabajando en agrupamientos de escenarios según su destino. Este es un aspecto sobre el que se tiene una gran confirmación experimental, el agrupamiento de los escenarios favorece la legibilidad del conjunto, sin embargo, no existe un sólo agrupamiento posible, por ejemplo, un agrupamiento pensado para la validación no es el mejor para facilitar el diseño del sistema. Se considera que la validación de los EA es muy diferente a la de los EF, ya que los EA representan un contexto existente, mientras que los EF representan un contexto que será real en un tiempo. En este sentido, la validación tiene diferente nivel de complejidad según sea el usuario o los desarrolladores. Por ejemplo, los EA son más fáciles de comprender para los usuarios ya que es el contexto existente, el cual conocen. En este caso la complejidad es para los ingenieros/as de requisitos. En cambio, en el caso de los EF, la complejidad mayor es para los usuarios ya que deben imaginar el contexto futuro que aún es inexistente.
- Se espera probar en casos reales el uso de puntos de vista en el LEL y Escenarios, como, así como también las nuevas heurísticas de la Construcción del LEL y de la Derivación de EA.

ANEXO Verificación de consistencias en el LEL y en los Escenarios

Proceso de Inspección

La Ingeniería de Software ha utilizado ampliamente las inspecciones como técnica de verificación y validación, primero de programas y luego de otros documentos creados durante el proceso de desarrollo de software [Fagan76] [Barnard94]. En el presente trabajo se propone un proceso de inspección para la verificación del LEL y de los Escenarios en la fase de Ingeniería de Requisitos. Esta inspección logra una importante mejora en la calidad del LEL producido y de los Escenarios, mediante tres actividades principales: planeamiento, preparación y reunión. Posteriormente, a partir de las Discrepancias, Errores y Omisiones (DEO) detectadas, son corregidos por sus autores y, de ser necesario, se realizan re-inspecciones.

Los métodos que se describen a continuación han sido diseñados para su utilización experimental en forma manual a los efectos de permitir una rápida revisión de los mismos. Los formularios están acompañados por un conjunto de guías y recomendaciones acerca de cómo deben ser llenados y qué aspectos del LEL o de los Escenarios que está siendo inspeccionado deben ser estudiados.

La fase de planeamiento consiste en la selección del material a inspeccionar, la elección de los participantes, la identificación de los roles (inspector, moderador y escriba) y la preparación del material a inspeccionar: el LEL o los Escenarios, los formularios de inspección a completar y la guía de instrucciones.

La fase de preparación es realizada por el inspector una vez que recibe el material y una copia del plan de inspección. La preparación consiste primero en la lectura cuidadosa de las instrucciones y luego en completar los formularios de inspección, registrando toda DEO detectada.

La fase de reunión apunta principalmente a confirmar o rechazar las DEO detectadas y secundariamente a descubrir nuevas DEO. En la reunión participan un moderador, un escriba, el inspector y los autores. Los autores convocados a la reunión realizan posteriormente las correcciones necesarias al LEL o a los Escenarios.

Las siguientes secciones detallan la Metodología de Verificación de Consistencia del Léxico Extendido del Lenguaje y sus respectivos formularios. Luego la Metodología de Verificación de Consistencias entre Escenarios de un Punto de Vista y sus respectivos formularios.

Metodología de Verificación de Consistencia del Léxico Extendido del Lenguaje

Tabla LEL. REPORTE DE CONSISTENCIAS EN EL LEL

Objetivo: Detectar discrepancias, errores y omisiones en cada símbolo del LEL:

- ◆ Se debe disponer de un LEL ordenado y clasificado.
- ◆ Se debe disponer de una versión del LEL en un medio en el que se pueda agregar información.

Proceso General:

- ◆ Numerar correlativamente los símbolos.
- ◆ Completar las páginas según el siguiente orden:

- ◆ Página I
- ◆ Página II
- ◆ Para cada uno de los símbolos:
 - ◇ Página III
 - ◇ Página IV
 - ◇ Página V
 - ◇ Página VI
 - ◇ Página VII
 - ◇ Página VIII
 - ◇ Página IX
 - ◇ Página X
 - ◇ Página XI
 - ◇ Página XII

Agregar en la numeración de cada página el número de símbolo. La Página XIII se debe llenar al mismo tiempo que se llenan las anteriores, ya que toda observación, corrección o información dudosa debe registrarse en esta planilla, tan pronto como se detecte y debe hacerse referencia a la página donde se originó.

- ◆ En el caso de las Páginas III a XII utilizar tantas hojas como sea necesario numerándolas correlativamente.
 - ◆ Una vez que se han procesado todas las Páginas de todos los símbolos, archivar las mismas siguiendo estrictamente los números de página.
 - ◆ Una vez finalizada la verificación, revisar los comentarios registrados en las Páginas XIII y realizar en la Página XIV un comentario general acerca de la calidad del LEL y sobre todo otro aspecto que corresponda.

Análisis:

En todas las páginas, especialmente de la III en adelante, es posible detectar inconvenientes de diferente naturaleza en los símbolos bajo estudio. En algunos casos, estos inconvenientes son subsanables en el sentido que con la información disponible es posible discernir con seguridad la corrección a realizar; mientras que en otros es necesario recurrir al autor de los mismos o eventualmente recabar información del Universo de Discurso.

Todo problema debe ser catalogado como subsanable sólo si existe una razonable confianza en la corrección que se propone, en caso contrario se lo debe considerar no subsanable.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

Pág. I. CARÁTULA.

Objetivo: Identificar el proyecto y LEL a consistir.

Pasos:

- ◆ Registrar la información básica y disponible del proyecto y del LEL
- ◆ Al finalizar el proceso, registrar el esfuerzo de verificación del LEL

Análisis:

No corresponde.

Pág. II. RESUMEN CUANTITATIVO DEL CONJUNTO DE SIMBOLOS.

Objetivo: Brindar una idea general del volumen de información presente en LEL.

Pasos:

- ◆ Registrar en la columna CANTIDAD los totales para cada rubro.
- ◆ Registrar en la fila TOTAL DE SIMBOLOS la cantidad de símbolos del LEL sin incluir los sinónimos
- ◆ Registrar en la fila SIMBOLOS TIPO SUJETO la cantidad de símbolos clasificados como tal
- ◆ Registrar en la fila SIMBOLOS TIPO OBJETO la cantidad de símbolos clasificados como tal
- ◆ Registrar en la fila SIMBOLOS TIPO ESTADO la cantidad de símbolos clasificados como tal
- ◆ Registrar en la fila SIMBOLOS TIPO VERBO la cantidad de símbolos clasificados como tal
- ◆ Registrar en la fila OTROS TIPOS la cantidad de símbolos no clasificados como sujeto, objeto, estado o verbo
- ◆ Registrar en la fila SINONIMOS la cantidad de símbolos que tienen sinónimos o la sumatoria de los sinónimos de todos los símbolos
- ◆ Registrar en la fila TOTAL DE NOCIONES la cantidad de nociones de todos los símbolos
- ◆ Registrar en la fila TOTAL DE IMPACTOS la cantidad de impactos de todos los símbolos
- ◆ Registrar en la fila TOTAL DE REFERENCIAS la cantidad de vínculos existentes en todos los símbolos
- ◆ Registrar en la fila TOTAL DE TERMINOS la cantidad, sin repetición
- ◆ Registrar en la fila TOTAL DE TERMINOS DEL VM la cantidad de palabras que no son símbolos del LEL utilizadas en las descripciones de los símbolos, sin repetición

Análisis:

En los casos donde no se encuentran símbolos tipo Sujeto, Objeto y Verbo debe ponerse atención debido a la alta probabilidad de falta de información. En caso de cantidad de nociones o impactos menor al Total de Símbolos es un indicativo de símbolos incompletos. En caso de una cantidad mayor de Símbolos Verbo en relación a los otros tipos, puede considerarse que el LEL está descripto poniendo énfasis en lo procedural.

Pág. III. COMPROBACIÓN SINTÁCTICA.

Objetivo: Verificar si los componentes de un símbolo se encuentran correctamente escritos.

Pasos:

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

- ◆ Para cada uno de los componentes del símbolo, comparar su texto con la sintaxis definida para el mismo
- ◆ Registrar en la columna FALTANTE toda omisión de elementos sintácticos obligatorios. Registrar estas anomalías en la Página XIII
- ◆ Registrar en la columna SOBRENTE todo elemento incluido en el componente que no se aparea con la sintaxis. Registrar estas anomalías en la Página XIII
- ◆ Para cada componente verificar si los faltantes registrados son irreparables a partir de los restantes componentes del símbolo; verificar también si los sobrantes pueden ser eliminados con razonable seguridad. Transcribir en la Página XIII las conclusiones de estas verificaciones

Análisis:

En la mayoría de los casos la existencia de sobrantes debe considerarse como texto a ser retirado del componente. Sin embargo, debe prestarse atención de que el sobrante sea alguna fracción que deba incluirse en otro componente. Por otra parte, en la mayoría de los casos los faltantes son irreparables.

Pág. IV. ANÁLISIS DE LA NOCIONES E IMPACTOS

Objetivo: permite detectar símbolos sin nociones o impactos y encontrar sinónimos.

Pasos:

- ◆ Registrar en la columna SÍMBOLO los números de los símbolos
- ◆ Registrar en la columna CANT NOCIÓN y CANT IMPACTO la cantidad de nociones e impactos de cada símbolo
- ◆ Para todos los símbolos restantes registrar en la columna SÍMBOLO el número de símbolo
- ◆ Para todos los símbolos restantes registrar en la columna CANT NOCIÓN y CANT IMPACTO la cantidad de nociones e impactos del mismo
- ◆ Analizar semánticamente cuantas nociones son iguales en cada dupla y registrarlo en la columna NOCIONES IGUALES
- ◆ Analizar semánticamente cuantos impactos son iguales en cada dupla y registrarlo en la columna IMPACTOS IGUALES
- ◆ Registrar en la columna OBS si existe alguna alerta en cuanto a la similitud de los símbolos. Registrar estas observaciones en la Página XIII

Análisis:

Encontrar símbolos con nociones e impactos similares da indicios de dos circunstancias bien diferentes: sinónimos no reconocidos como tales o símbolos omitidos cuya noción e impacto se repiten en otros símbolos. Los defectos que se detectan a través de este formulario se refieren a símbolos incompletos (omisión de nombres, noción o impacto) y símbolos omitidos (definiciones incorporadas en otros símbolos).

Pág. V, VI, VII, VIII y IX . VERIFICACIÓN DE TIPO OBJETO, VERIFICACIÓN DE TIPO SUJETO, VERIFICACIÓN DE TIPO VERBO, VERIFICACIÓN DE TIPO ESTADO Y VERIFICACIÓN DE TIPO

Objetivo: Estos formularios son similares en formato, difiriendo en la definición del tipo específico para cada formulario. Ayudan a detectar si el tipo asignado a cada símbolo es correcto y si su noción e impacto coincide con el contenido especificado para su tipo.

Pasos:

- ◆ Para los tipos nuevos (agregados a la clasificación general) determinar el nombre del tipo en el título y el contenido de la noción e impacto

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

- ◆ Registrar en la columna SÍMBOLO el número de símbolo a analizar
- ◆ Registrar en la columna TIPO ERRÓNEO si el símbolo está bien o mal clasificado teniendo en cuenta la definición del tipo asignado al símbolo. En caso de mal clasificado registrarlo en la Página XIII
- ◆ Determinar si existe alguna parte de la noción del símbolo que sobre o falte con respecto a la definición del tipo correspondiente. Si se encontrara alguno mencionar en la columna NOCIÓN en subcolumna SOBRANTE o FALTANTE la porción de texto correspondiente
- ◆ En el caso de un sobrante determinar en la columna GENERO si el sobrante puede corresponder a otro símbolo o si corresponde al componente Impacto. Registrar estas anomalías en la Página XIII
- ◆ En caso de Faltante indicar que parte de la definición se omitió. Registrar estas anomalías en la Página XIII
- ◆ Determinar si existe alguna parte del impacto del símbolo que sobre o falte con respecto a la definición del tipo correspondiente. Si se encontrara alguno mencionar en la columna IMPACTO en subcolumna SOBRANTE o FALTANTE la porción de texto correspondiente.
- ◆ En el caso de un sobrante determinar en la columna GENERO si el sobrante puede corresponder a otro símbolo o si corresponde al componente Noción. En caso de Faltante indicar que parte de la definición se omitió. Registrar estas anomalías en la Página XIII

Análisis:

En los casos donde se encuentra Tipos Erróneos puede indicar un error en la clasificación del símbolo o un error en la descripción del mismo. Este último caso sucede cuando la definición no se corresponde con el símbolo o cuando existe una confusión entre noción e impacto. Cuando se detectan sobrantes o faltantes es un indicativo de símbolos incompletos según la definición del tipo o que las descripciones de dichos símbolos son incompatibles con el tipo asignado.

Pág. X. IDENTIFICACIÓN DE SÍMBOLOS.

Objetivo: detectar símbolos no destacados en la definición de otros símbolos, símbolos utilizados con otro significado y el uso parcial del nombre del símbolo en la definición de otros símbolos.

Pasos:

- ◆ Registrar en la columna SÍMBOLO toda porción de texto destacada como símbolo del LEL, excluyendo los usos duplicados y los sinónimos
- ◆ Para cada una de los presuntos símbolos del LEL comprobar que símbolos hacen referencia a él y registrarlo en la columna SÍMBOLO QUE LO MENCIONAN
- ◆ Para cada símbolo de la columna SÍMBOLO QUE LO MENCIONAN, buscar si existe un homónimo al símbolo o a algún sinónimo donde se omitió destacarlo como tal. Si se encontrara alguno, indicar la cantidad de referencias detectadas en la columna SIN DESTACAR subcolumna NOCIÓN o IMPACTO según corresponda. Registrar esta anomalía en la Página XIII
- ◆ Para cada símbolo de la columna SÍMBOLO QUE LO MENCIONAN, buscar si existe algún uso del símbolo con un significado diferente al registrado en el LEL. Si se encontrara alguno registrarlo en la columna MAL USO y en la Página XIII
- ◆ Para cada símbolo de la columna SÍMBOLO QUE LO MENCIONAN, buscar si existen usos parciales del símbolo, o sea donde un texto identificados como

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

símbolos no corresponde exactamente a un símbolo o a un sinónimo del mismo. Si se encontrara alguno registrar en la columna USO PARCIAL y en la Página XIII

Análisis:

Detectar símbolos sin destacar cuando el uso es correcto, tan solo muestra la omisión del mecanismo de identificación de los símbolos. Pero si no fue destacado como símbolo y además está usado con otro significado, es necesario reemplazar dicho texto por otro. El uso parcial del nombre del símbolo puede corresponder a otro nombre del mismo símbolo, o sea a un sinónimo, o que se trata de un símbolo no detectado, generalmente con un significado más amplio, o que se trata de un error en el uso del nombre del símbolo, proveniente del Ingeniero de Requisitos o de la fuente de información.

Pág. XI. VERIFICACIÓN DEL PRINCIPIO DE CIRCULARIDAD.

Objetivo: Este formulario permite detectar símbolos que no son mencionados por otros símbolos o símbolos que no mencionan a ningún otro símbolo.

Pasos:

- ◆ Registrar en la columna SÍMBOLO el número de símbolo a analizar
- ◆ Registrar en la columna REFERENCIA A OTROS SÍMBOLOS en la subcolumna NOCIÓN e IMPACTO cuantos símbolos son referenciados en cada componente
- ◆ Registrar en la columna SÍMBOLOS QUE LO MENCIONAN cuantos símbolos referencian al símbolo de la primera columna

Análisis:

Símbolos sin referencias en su noción e impacto puede indicar la presencia de símbolos que no pertenecen al Universo de Discurso o simplemente, la inclusión de definiciones parciales de símbolos en otros símbolos eludiendo la referencia directa a dichos símbolos del LEL. Una tercera posibilidad puede indicar la presencia de símbolos cuya descripción incluye la definición de otros símbolos no detectados como tales en el Universo de Discurso.

Pág. XII. VERIFICACIÓN DEL PRINCIPIO DE VOCABULARIO MINIMO.

Objetivo: se utiliza para encontrar términos utilizados en la definición de símbolos que no corresponden al vocabulario mínimo ni pertenecen al LEL.

Pasos:

- ◆ Registrar en la columna SÍMBOLO el número de símbolo a analizar
- ◆ Para cada símbolo, registrar en la columna TERMINOS cuantos términos utilizados sin contar los símbolos del LEL son utilizados y sin repetirlos
- ◆ Para cada símbolo, registrar en la columna REFERENCIAS cuantas referencias son utilizadas en el símbolo sin repetición
- ◆ Para cada símbolo, registrar en la columna TERMINOS DE VM cuantos términos pertenecientes al VM son utilizados en la descripción del símbolo
- ◆ Registrar en la columna OBS una marca cuando:
 - ◆ el uso de términos es mayor al uso de términos del VM
 - ◆ las referencias son nulas
 - ◆ las referencias son muy escasas y el uso de términos elevado
 - ◆ la cantidad de términos del VM es muy escasa

Análisis:

Cuando las referencias son escasas puede indicar omisiones de símbolos en el LEL. En particular cuando el uso del VM es excesivo puede corresponder a dos circunstancias bien

Tesis doctoral “**Proceso de Requisitos Validado Empíricamente**”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

diferentes. Por un lado, puede indicar omisiones de referencias a símbolos por inclusión de la noción o el impacto de dichos símbolos en la definición de otros símbolos sin hacer uso del nombre del símbolo. Por otro lado, puede indicar que se está en presencia de un símbolo que no tiene relevancia en el Universo de Discurso.

REPORTE DE CONSISTENCIAS DEL LEL

Proyecto:

LEL Inspeccionado

Grupo/Autor(es):

Versión:

Fecha:

Esfuerzo de construcción:

Datos Generales de la Inspección

Inspector:

Fecha de Verificación:

Esfuerzo de Verificación:

RESUMEN CUANTITATIVO DEL CONJUNTO DE
SIMBOLOS

	CANTIDAD
TOTAL DE SÍMBOLOS*	
SÍMBOLOS TIPO SUJETO	
SÍMBOLOS TIPO OBJETO	
SÍMBOLOS TIPO ESTADO	
SÍMBOLOS TIPO VERBO	
OTROS TIPOS	
SINÓNIMOS	
TOTAL DE NOCIONES	
TOTAL DE IMPACTOS	
TOTAL DE REFERENCIAS	
TOTAL DE TERMINOS&	
TOTAL DE TERMINOS DEL VM&	

* sin incluir sinónimos

& sin repetición

COMPROBACIÓN SINTÁCTICA

SÍMBOLO:

COMPONENTE	SINTAXIS	FALTANTE	SOBRANTE
NOMBRE	<i>Palabra frase / [sinónimos]</i>		
NOCIÓN	<i>[Sujeto] + verbo + predicado</i>		
IMPACTO	<i>[Sujeto] + verbo + [predicado]</i>		

Número secuencial

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

TABLA LEL

Pág. IV-----

#

ANÁLISIS DE LAS NOCIONES E IMPACTOS

SÍMBOLO [§]	CANT NOCIÓN	CANT IMPACTO	SÍMBOLO [§]	CANT NOCIÓN	CANT IMPACTO	NOCIONES IGUALES [%]	IMPACTOS IGUALES [^]	OBS [*]

Número secuencial § Número de símbolo de la lista % Cantidad de Nociones iguales ^ Cantidad de Impactos iguales * Marcar en caso de observaciones

Tesis doctoral "Proceso de Requisitos Validado Empíricamente"

Anexo - Verificación de consistencias en el LEL y en los Escenarios

TABLA LEL

Pág. V -----
#

VERIFICACIÓN DE SÍMBOLOS TIPO OBJETO

NOCIÓN: *DEFINICIÓN DEL OBJETO / OTROS OBJETOS CON LOS QUE SE RELACIONA*

IMPACTO: *ACCIONES QUE PUEDEN SER APLICADAS AL OBJETO*

SÍMBOLO@	TIPO ERRONEO^	NOCIÓN			IMPACTOS		
		SOBRANTE	GENERO*	FALTANTE	SOBRANTE	GENERO*	FALTANTE

Número secuencial

@ Número de símbolo

^ SI o NO

* ∉ no pertenece al símbolo

↔ corresponde al otro componente

Tesis doctoral "Proceso de Requisitos Validado Empíricamente"

Anexo - Verificación de consistencias en el LEL y en los Escenarios

TABLA LEL

Pág. VI-----
#

VERIFICACIÓN DE SÍMBOLOS TIPO SUJETO

NOCIÓN: **DEFINICION DEL SUJETO**
IMPACTO: **ACTIVIDADES QUE REALIZA O RECIBE**

SIMBOLO@	TIPO ERRONEO^	NOCIÓN			IMPACTOS		
		SOBRANTE	GENERO*	FALTANTE	SOBRANTE	GENERO*	FALTANTE

Número secuencial

@ Número de símbolo

^ SI o NO

* ∉ no pertenece al símbolo

↔ corresponde al otro componente

VERIFICACIÓN DE SÍMBOLOS TIPO VERBO

NOCIÓN: QUIEN REALIZA LA ACCIÓN / CUANDO OCURRE

IMPACTO: QUE PROCESOS ESTAN INVOLUCRADOS / CONSECUENCIAS DE LA ACCION / CAMBIOS DE ESTADO QUE PROVOCA LA ACCION / NUEVAS ACCIONES QUE DESENCADENA / RESTRICCIONES QUE AFECTAN A LA REALIZACION DE LA ACCION

SÍMBOLO@	TIPO ERRONEO^	NOCIÓN			IMPACTOS		
		SOBRANTE	GENERO*	FALTANTE	SOBRANTE	GENERO*	FALTANTE

Número secuencial

@ Número de símbolo

^ SI o NO

* ∉ no pertenece al símbolo

↔ corresponde al otro componente

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

TABLA LEL

Pág. VIII-----

#

VERIFICACIÓN DE SÍMBOLOS TIPO ESTADO

NOCIÓN: ***DEFINICION DEL ESTADO / ACCIONES QUE LLEVARON AL ESTADO***
IMPACTO: ***ACCIONES Y ESTADOS QUE PUEDEN OCURRIR A PARTIR DE ESTE ESTADO***

SÍMBOLO [@]	TIPO ERRONEO [^]	NOCIÓN			IMPACTOS		
		SOBRANTE	GENERO*	FALTANTE	SOBRANTE	GENERO*	FALTANTE

Número secuencial

@ Número de símbolo

[^] SI o NO

* ∉ no pertenece al símbolo

↔ corresponde al otro componente

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

TABLA LEL

Pág. IX.-----

#

VERIFICACIÓN DE SÍMBOLOS TIPO _____

NOCIÓN:

IMPACTO:

SÍMBOLO@	TIPO ERRONEO^	NOCIÓN			IMPACTOS		
		SOBRANTE	GENERO*	FALTANTE	SOBRANTE	GENERO*	FALTANTE

Número secuencial

@ Número de símbolo

^ SI o NO

* ∉ no pertenece al símbolo

↔ corresponde al otro componente

IDENTIFICACIÓN DE SÍMBOLOS

SÍMBOLO [#]	SÍMBOLO [%] QUE LO MENCIONA	SIN DESTACAR [*]		MAL USO [^]	USO PARCIAL [^]
		NOCIÓN	IMPACTO		

Número secuencial % Número de símbolo * Cantidad de referencias ^ SI o NO

Metodología de Verificación de Consistencias entre Escenarios

Tabla 2. **REPORTE DE CONSISTENCIAS ENTRE ESCENARIOS**

Objetivo: Detectar discrepancias, errores y omisiones entre el conjunto de escenarios de un mismo autor.

Precondiciones:

- El responsable de la verificación debe conocer el LEL.
- Esta verificación se debe realizar después de la Verificación de Consistencia interna de Escenarios.
- El soporte de los escenarios debe permitir la inserción de información.

Proceso General:

- Completar las páginas I a XI en el orden de su numeración y de acuerdo a las indicaciones dadas para cada una.
- En el caso de las páginas III a XIII, utilizar tantas hojas como sea necesario numerándolas correlativamente.
- Una vez completada cada página o simultáneamente al llenado de la misma, realizar un análisis de los datos obtenidos.
- Registrar en la página XII, en el cuadro CORRECCIONES PROPUESTAS todas las recomendaciones posibles orientadas a mejorar la descripción de los escenarios, y en el cuadro DUDAS toda cuestión no discernible y que requiera la validación con el cliente/usuario o con el autor del escenario. En ambos casos, hacer referencia a la página que originó la observación.
- Finalizada la verificación, revisar los comentarios registrados en las páginas XII y realizar en la página XIII un comentario general, basado en el nivel de correcciones y de dudas detectados, sobre la calidad del conjunto de escenarios y sobre todo otro aspecto que corresponda y que permita tomar decisiones sobre los pasos a seguir.

Análisis:

En las páginas II a XII es posible detectar inconvenientes de diferente naturaleza en el conjunto de escenarios bajo estudio. En algunos casos, estos inconvenientes son subsanables en el sentido que con la información disponible es posible discernir con seguridad la corrección a realizar; mientras que en otros es necesario recurrir al autor/es de los mismos o eventualmente recabar información del Universo de Discurso. Todo problema debe ser catalogado como subsanable sólo si existe una razonable confianza en la corrección que se propone, en caso contrario se la debe considerar no subsanable.

Pág. I. **CARÁTULA.**

Objetivo: Identificar el proyecto y el conjunto de escenarios a consistir.

Pasos:

- Registrar la información básica y disponible del proyecto y del conjunto de escenarios.
- Al finalizar el proceso, registrar el esfuerzo de verificación del conjunto de escenarios.

Análisis:

No corresponde.

Pág. II. RESUMEN CUANTITATIVO DEL CONJUNTO DE ESCENARIOS.

Objetivo: Brindar una idea general del volumen de información presente en el conjunto de escenarios.

Pasos:

- Clasificar los escenarios según su nivel de jerarquía en Escenarios Integradores¹⁶, Escenarios¹⁷ y Subescenarios¹⁸.
- Registrar en la columna CANTIDAD los totales para cada rubro sin incluir los usos repetidos, salvo el caso de episodios. En el caso de excepciones, contabilizar tanto las correspondientes a episodios como a escenarios. En el caso de restricciones, incluir aquellas adjuntas al contexto, a los recursos y a los episodios. Utilizar el LEL para contabilizar los símbolos que pertenecen a la clase Sujeto y a la clase Objeto.

Análisis:

En el caso de una baja cantidad de Subescenarios, evaluar el hecho como posible indicador de descripciones de escenarios excesivamente planas.

En el caso de un valor bajo en el promedio de episodios por escenario, considerar la posible existencia de una alta disgregación de situaciones.

En el caso de un valor alto en el promedio de episodios por escenario, considerar la posible existencia de una excesiva concentración de situaciones en un sólo escenario.

En el caso de valores bajos en excepciones y restricciones, considerar la posibilidad de omisiones.

En el caso en que el uso del vocabulario del LEL deje una cantidad significativa sin mencionar, considerar la posibilidad de omisión de detalles y/o situaciones.

En el caso que la cantidad de sujetos del LEL sea significativamente mayor que la cantidad de actores, considerar la posibilidad de omisión de situaciones o de no uso del vocabulario del LEL. En el caso inverso, considerar la posibilidad de estar involucrando situaciones fuera del alcance del problema o de la incompletitud del LEL.

En el caso que la cantidad de objetos del LEL sea significativamente mayor que la cantidad de recursos, considerar la posibilidad de omisión de información o de no uso del vocabulario del LEL. En el caso inverso, considerar la posibilidad del estudio de acciones fuera del alcance del problema o de un excesivo nivel de detalle o de la incompletitud del LEL.

Pág. III. COMPOSICIÓN DE ESCENARIOS INTEGRADORES.

Objetivo: Asegurarse que los escenarios identificados como integradores estén correctamente clasificados.

Pasos:

¹⁶ Escenario Integrador: aquél que representa una visión global de la totalidad o parte del problema. Sus episodios son Escenarios.

¹⁷ Escenario: aquél que representa una situación particular y se encuentra en el primer nivel de la jerarquía. Puede contener o no Subescenarios.

¹⁸ Subescenario: aquél que es a su vez un episodio o una excepción de otro Escenario o Subescenario.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

- Confeccionar una lista con todos los escenarios identificados como Escenarios Integradores y registrarlo en la columna ESCENARIO INTEGRADOR, ordenados por número de escenario.
- Registrar en la columna TOTAL DE EPISODIOS la cantidad total de episodios que componen cada Escenario Integrador.
- Registrar en la columna ESCENARIOS la lista de episodios identificados como Escenarios (en letra mayúscula y pertenecientes a la clasificación Escenario) que componen el Escenario Integrador.
- Para cada Escenario Integrador, registrar en la columna SUBESCENARIOS la lista de episodios identificados como Subescenarios (en letra mayúscula y pertenecientes a la clasificación Subescenario) y transcribirla también a la página XII con las observaciones que correspondan.
- Para cada Escenario Integrador, registrar en la columna SIMPLES la lista de episodios que representan acciones simples y transcribirla también a la página XII con las observaciones que correspondan.

Análisis:

Para cada Escenario Integrador, examinar todos los episodios simples existentes con el fin de determinar si se trata de uno de cuatro casos siguientes:

- a) episodios que no han sido destacados como escenarios, o
- b) un escenario mal clasificado como Escenario Integrador, o
- c) acciones que manifiestan la existencia de situaciones total o parcialmente ignoradas en el conjunto de escenarios, o
- d) acciones incluidas innecesariamente en el Escenario Integrador por ya existir en algún Escenario o Subescenario.

Si no se puede asegurar ninguna de estos casos, entonces proponer una revisión exhaustiva del Escenario Integrador y de los episodios involucrados.

Si se puede asegurar que se trata de un episodio que no ha sido destacado como un Escenario, la corrección del error es simple ya que sólo consiste en destacar el episodio como Escenario.

Si se puede asegurar que se trata de un episodio que no ha sido destacado como un Subescenario, entonces no alcanza con sólo destacar el episodio como Subescenario ya que éstos no deben existir en un Escenario Integrador y, por lo tanto, analizar la redefinición del Escenario Integrador o del Subescenario.

Si se puede asegurar que se trata de un escenario mal clasificado, reclasificar el Escenario Integrador como Escenario y en caso que la columna Escenarios no esté vacía, reclasificar cada Escenario como Subescenario.

Si se puede asegurar que se trata de acciones total o parcialmente ignoradas, recomendar el agregado de los escenarios necesarios o la inclusión de dichas acciones en los escenarios correspondientes.

Si se puede asegurar que se trata de acciones incluidas innecesariamente en el Escenario Integrador, recomendar la eliminación de estos episodios.

Si el Escenario es Integrador y existen Subescenarios, la corrección del error es importante ya que requiere redefinir el Escenario Integrador o los Subescenarios incluidos.

Pág. IV. **COMPROBACIÓN DE SUBESCENARIOS.**

Objetivo: Detectar la inexistencia de subescenarios mencionados como tales en episodios o excepciones y detectar subescenarios existentes no destacados en episodios o excepciones.

Pasos:

- Registrar en la columna CANDIDATO A SUBESCENARIO aquellos episodios y excepciones destacados como Subescenarios dentro de Escenarios y otros Subescenarios, sin incluir usos duplicados.
- Verificar la existencia del candidato a Subescenario dentro del conjunto de escenarios y registrarlo en la columna EXISTE. Si no exista el Subescenario, registrar este hecho en la página XII.
- Agregar en la columna CANDIDATO A SUBESCENARIO la lista de Subescenarios, clasificados en la página II y que no han sido registrados en el primer paso, y confirmar su existencia en la columna EXISTE.
- Para cada candidato a Subescenario, contabilizar en la columna CANT el total de apariciones del mismo en los episodios y excepciones. Si no se encontrara ninguna ocurrencia, registrar este hecho en la página XII.
- Registrar en la columna NO DESTACADO la lista de escenarios/episodios/excepciones donde los candidatos a Subescenarios no se encuentran destacados e informarlo también en la página XII para su corrección.

Análisis:

Para aquellos candidatos a Subescenarios no encontrados, observar la cantidad de ocurrencias del mismo dentro de los escenarios, para poder determinar si se trata de una acción simple o de una situación que involucra más de una acción y que, por lo tanto, requiere la existencia de un escenario.

En el caso de Subescenarios con ocurrencia nula en episodios y/o excepciones, analizar la posibilidad que se trate de un escenario mal clasificado.

Para aquellos candidatos a Subescenarios confirmados (encontrados) y que no han sido destacados en algún escenario, la corrección es simple ya que consiste en destacar toda ocurrencia del mismo.

Pág. V. **CONSISTENCIA DE PRECONDICIONES EN EL CONTEXTO.**

Objetivo: Detectar errores o discrepancias en las precondiciones de los escenarios u omisión de información para satisfacer dichas precondiciones.

Pasos:

- Registrar en la columna ESCENARIO la lista de escenarios que presenten al menos una precondición, ordenada por número de escenario.
- Completar en la columna PRECONDICION un renglón por cada precondición del escenario.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

- Clasificar el origen de las precondiciones en internas¹⁹ o externas²⁰ y transcribirlo a la columna ORIGEN. En caso de no poder catalogarse el origen, completar con la palabra “Error” y registrarlo también en la página XII.
- Registrar en la columna QUIEN LA SATISFACE la lista de escenarios que satisfacen las precondiciones internas, indicando de ser posible el o los números de episodios involucrados. En caso de no poder establecerse esto, registrarlo en la página XII con las observaciones que correspondan.

Análisis:

En el caso de precondiciones con error en el origen, considerar la posibilidad de corrección de la precondición o su eliminación.

En el caso que una precondición interna no sea satisfecha por al menos un escenario, considerar la posibilidad de corregir la precondición o analizar la falta de un escenario o la falta de episodios dentro de escenarios existentes, para satisfacer dicha precondición interna.

Pág. VI. **RELACIÓN ENTRE ESCENARIOS Y SUBESCENARIOS.**

Objetivo: Detectar errores en las precondiciones de Subescenarios u omisión de información en los escenarios que contienen Subescenarios, considerando los vínculos jerárquicos Escenario-Subescenario y Escenario Integrador-Escenario.

Pasos:

- Confeccionar una lista con cada par Escenario-Subescenario y registrarlo en las columnas ESCENARIO y SUBESCENARIO respectivamente, ordenados por número de escenario. En el caso de presentar el Subescenario precondiciones, anotar en la columna PRECONDICIÓN SUBESCENARIO en un renglón separado cada una. En el caso de no presentar, registrar este hecho en la página XII.
- Agregar a la lista anterior todos los pares Escenario Integrador-Escenario y tratarlos en lo sucesivo como si fueran Escenario-Subescenario respectivamente. Registrar para estos pares la misma información indicada en el paso previo.
- Establecer la relación existente entre la precondición del Subescenario y la precondición del Escenario y/o los episodios anteriores a su mención en el Escenario, y registrarlo en la columna RELACIÓN CON EL ESCENARIO. De no poder establecerse esta relación, registrarlo en la página XII con las observaciones que correspondan.

Análisis:

En el caso que el Subescenario no presente ninguna precondición, considerar la posible omisión de la misma. Para ello evaluar uno de los siguientes casos:

- a) Subescenario que no es primer episodio del Escenario, o
- b) Subescenario que es primer episodio del Escenario.

Si el Subescenario no es el primer episodio del Escenario, analizar la secuencia de los episodios anteriores en el Escenario. Si esta secuencia es mandatoria,

¹⁹ Precondición Interna: aquella precondición satisfecha por algún escenario. Precondición de origen local.

²⁰ Precondición Externa: aquella precondición proveniente del macrosistema. Precondición de carácter general.

fabrica naturalmente la precondition del Subescenario. En el caso contrario analizar el caso b).

Si el Subescenario es el primer episodio del Escenario y éste presenta precondition, analizar dicha precondition para evaluar si no es también la precondition del Subescenario.

En el caso que la precondition del Subescenario no sea satisfecha por al menos un episodio del Escenario o que no coincida con la precondition del Escenario, evaluar uno de estos casos:

- a) error u omisión en la precondition del Subescenario, o
- b) error u omisión en la precondition del Escenario, o
- c) omisión de información o episodios en el Escenario para satisfacer dicha precondition.

Si no se puede asegurar ninguno de estos casos, entonces recomendar una revisión general del Escenario y del Subescenario.

Si se puede asegurar que se trata del caso a) o del b) sugerir la corrección de la precondition del Subescenario o del Escenario respectivamente o ambas condiciones.

Si se puede asegurar que se trata del caso c) considerar la posibilidad de incluir en el Escenario los episodios faltantes o la información omitida en episodios existentes.

Pág. VII. CONTROL DE SUPERPOSICIÓN DE ESCENARIOS.

Objetivo: Detectar la superposición de información entre escenarios.

Pasos:

- Confeccionar una lista de pares de escenarios todos contra todos, excluyendo los Escenarios Integradores y registrarlos en las columnas ESCEN_i y ESCEN_j respectivamente, ordenados por número de escenario.
- Para cada par, registrar en la columna A_{∪ij} la unión de Actores, en la columna A_{∩ij} el total de coincidencias de Actores, en la columna R_{∪ij} la unión de Recursos, en la columna R_{∩ij} el total de coincidencias de Recursos, en la columna C_{∪ij} la unión de Contextos y en la columna C_{∩ij} el total de coincidencias de Contextos.
- Para cada par de escenarios, calcular el Índice de cercanía entre el escenario *i* y el escenario *j* como:

$$I_{ij} = (\alpha * C_{\cap ij} + \beta * A_{\cap ij} + \gamma * R_{\cap ij}) / (\alpha * C_{\cup ij} + \beta * A_{\cup ij} + \gamma * R_{\cup ij})$$

donde:

$$C_{\cap ij} = | \text{Cont}(E_i) \cap \text{Cont}(E_j) |, A_{\cap ij} = | \text{Act}(E_i) \cap \text{Act}(E_j) |, R_{\cap ij} = | \text{Rec}(E_i) \cap \text{Rec}(E_j) |$$
$$C_{\cup ij} = | \text{Cont}(E_i) \cup \text{Cont}(E_j) |, A_{\cup ij} = | \text{Act}(E_i) \cup \text{Act}(E_j) |, R_{\cup ij} = | \text{Rec}(E_i) \cup \text{Rec}(E_j) |$$

α, β, γ , factores de peso, y

Cont(E_k) = Conjunto de enunciados del contexto del escenario *k*

Act(E_k) = Conjunto de actores del escenario *k*

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

$Rec(E_k)$ = Conjunto de recursos del escenario k

- Para cada par cuyo Índice supere un determinado valor, registrar en la columna OBJETIVO el total de coincidencias en el Objetivo (considerar partes comunes) y en la columna EPISODIO el total de coincidencias en los Episodios.
- Para los pares considerados en el paso anterior, marcar en la columna OVERLAP aquellos que presenten un alto grado de coincidencia y registrar estos pares en la página XII.

Análisis:

El índice es útil para reducir el esfuerzo en la cantidad de comparaciones, ya que sólo se examinan los pares que podrían tener superposición.

Si un par de escenarios tiene coincidencia total o parcial en sus objetivos, existe una superposición entre ellos. Esta superposición se podrá corregir, eliminando completamente un escenario o redefiniendo los objetivos de uno o ambos escenarios y los restantes componentes.

Si un par de escenarios tiene objetivos disjuntos, pero tienen episodios comunes, esto puede deberse a la inclusión indebida de estos episodios en uno de los escenarios. La verosimilitud de esta suposición crece con la cantidad de episodios comunes.

Pág. VIII. **VERIFICACIÓN DE OCURRENCIA DE ACTORES.**

Objetivo: Detectar actores no incluidos en el LEL y con gran participación en los escenarios.

- Armar una lista con los actores que participan en el conjunto de escenarios, sin incluir los usos duplicados y registrarlos en la columna ACTOR.
- Para cada actor, registrar en las columnas ESCENARIOS y EPISODIOS el total de escenarios y el total de episodios respectivamente en los que participa y registrar en la columna LEL si pertenece al LEL.
- Para los actores que no pertenezcan al LEL, registrarlos en la página XII, indicando si tienen mucha o poca participación en los escenarios y episodios, con las observaciones que correspondan.

Análisis:

En el caso de Actores con una alta participación en el total de escenarios y en el total de episodios y que no son símbolos del LEL, considerar la posibilidad de omisión del actor como símbolo en el LEL.

En el caso de actores con baja participación en escenarios y que no pertenecen al LEL, analizar los episodios en los que actúan y considerar la posibilidad que dichos episodios estén fuera del alcance de la aplicación.

Si el actor es un símbolo del LEL no se verifica su pertenencia a la clase Sujeto, ya que esta verificación es realizada en la Tabla 1.

Pág. IX. **VERIFICACIÓN DE OCURRENCIA DE RECURSOS.**

Objetivo: Detectar recursos no incluidos en el LEL y con gran participación en los escenarios.

- Armar una lista con los recursos empleados en el conjunto de escenarios, sin incluir los usos duplicados y registrarlos en la columna RECURSO.

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

- Para cada recurso, registrar en las columnas ESCENARIOS y EPISODIOS el total de escenarios y el total de episodios respectivamente en los que se menciona y registrar en la columna LEL si pertenece al LEL.
- Para los recursos que no pertenezcan al LEL, registrarlos en la página XII, indicando si tienen mucha o poca participación en los escenarios y episodios, con las observaciones que correspondan.

Análisis:

En el caso de Recursos con una alta participación en el total de escenarios y en el total de episodios y que no son símbolos del LEL, considerar la posibilidad de omisión del recurso como símbolo en el LEL.

En el caso de recursos con baja participación en escenarios y que no pertenecen al LEL, analizar los episodios en los que participan y considerar la posibilidad que dichos episodios estén fuera del alcance de la aplicación o que haya un excesivo detalle de información.

Si el recurso es un símbolo del LEL no se verifica su pertenencia a la clase Objeto, ya que esta verificación es realizada en la Tabla 1.

Pág. X. SÍMBOLOS DEL LEL NO UTILIZADOS.

Objetivo: Detectar omisiones en el conjunto de escenarios mediante el análisis del grado de cubrimiento de los símbolos del LEL.

Pasos:

- Registrar en la columna SÍMBOLO aquellos símbolos del LEL no empleados en ningún escenario, preservando el orden establecido en el LEL.
- Registrar en la columna LEL la clase a la que pertenece el símbolo.
- Contabilizar el total de impactos que presenta y registrarlo en la columna IMPACTOS.
- Para aquellos símbolos con más de un impacto, contabilizar el total de referencias a otros símbolos que se mencionan en sus impactos, sin incluir los usos duplicados y registrarlo en la columna REF.

Análisis:

En el caso de ser alta la cantidad de símbolos del LEL no utilizados, considerar la posibilidad de haber hecho un escaso uso del LEL en la descripción de los escenarios.

En el caso de Sujetos en el LEL no utilizados, considerar la posibilidad de escenarios omitidos o incompletos.

En el caso de Objetos en el LEL no utilizados y que hacen mucha referencia a otros símbolos, considerar la posibilidad de escenarios incompletos.

En el caso de Verbos en el LEL no utilizados y que hacen mucha referencia a otros símbolos, considerar la posibilidad de escenarios omitidos o incompletos.

En el caso de Estados en el LEL no utilizados y que hacen mucha referencia a otros símbolos, considerar la posibilidad de acciones omitidas o incompletas.

Pág. XI. CONTROL DE IMPACTOS VERSUS ESCENARIOS/EPISODIOS.

Objetivo: Detectar situaciones omitidas en los escenarios y/o discrepancias entre los sujetos y los actores de determinadas acciones.

Pasos:

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”

Anexo - Verificación de consistencias en el LEL y en los Escenarios

- Completar la columna SUJETO con cada símbolo del LEL perteneciente a la clase Sujeto que presente al menos un impacto, preservando el orden establecido en el LEL.
- Para cada símbolo, registrar cada uno de los impactos en un renglón separado en la columna IMPACTO.
- Identificar los escenarios que satisfacen cada impacto y especificar en la columna ESCENARIOS/EPISODIOS la lista de escenarios y de episodios involucrados y en la columna ACTORES la lista de los actores que participan en dichas acciones.

Análisis:

En el caso que un impacto no sea satisfecho por al menos un escenario, considerar la posibilidad de la omisión de una situación.

En el caso que el sujeto del impacto no coincida con ningún actor del escenario, considerar la posibilidad de una discrepancia entre quienes ejecutan las acciones.

REPORTE DE CONSISTENCIAS ENTRE ESCENARIOS

Proyecto:
Responsable:
Fecha de Verificación:
Esfuerzo de Verificación:
Grupo/Autor(es):
Versión:
Fecha:
Técnica de construcción:
Esfuerzo de construcción:
Versión del LEL:

RESUMEN CUANTITATIVO DEL CONJUNTO DE
 ESCENARIOS

	CANTIDAD
ESCENARIOS	
ESCENARIOS INTEGRADORES	
ESCENARIOS	
SUBESCENARIOS	
ACTORES*	
RECURSOS*	
EPIODIOS ⁺	
EXCEPCIONES ^{o*}	
RESTRICCIONES ^{o*}	
REFERENCIAS UNICAS DEL LEL*	
SÍMBOLOS DEL LEL SUJETO	
SÍMBOLOS DEL LEL OBJETO	

⁺ Incluyendo los episodios que son subescenarios ^{*} Sin incluir los usos duplicados ^o Totales

TABLA 2

Pág. III.-----

#

COMPOSICIÓN DE ESCENARIOS INTEGRADORES

ESCENARIO INTEGRADOR [@]	TOTAL DE EPISODIOS	LISTA DE EPISODIOS		
		ESCENARIOS [°]	SUBESCENARIOS [°]	SIMPLES [°]

[@] Número de escenario # Número secuencial [°] Lista de episodios

Tesis doctoral “Proceso de Requisitos Validado Empíricamente”
 Anexo - Verificación de consistencias en el LEL y en los Escenarios

TABLA 2

Pág. V.-----
#

CONSISTENCIA DE PRECONDICIONES EN EL CONTEXTO

ESCENARIO [@]	PRECONDICIÓN	ORIGEN [§]	QUIEN LA SATISFACE [*]

[@] Número de escenario # Número secuencial [§] Origen de Precondición: Interna, Externa, Error ^{*} Lista de escenarios, completar cuando Origen = I

TABLA 2

RELACIÓN ENTRE ESCENARIOS Y SUBESCENARIOS

ESCENARIO [@]	SUBESCENARIO [@]	PRECONDICIÓN SUBESCENARIO	RELACIÓN CON EL ESCENARIO ^{\$}

[@] Número de escenario # Número secuencial ^{\$} Precondición y/o lista de episodios del escenario u otro componente

TABLA 2

CONTROL DE SUPERPOSICIÓN DE ESCENARIOS

ESCENi [@]	ESCENj [@]	ACTOR		RECURSO		CONTEXTO		Iij [*]	OBJETIVO ⁺	EPISODIO ⁺	OVERLAP [%]
		A _{∪ij} ^{&}	A _{∩ij} ⁺	R _{∪ij} ^{&}	R _{∩ij} ⁺	C _{∪ij} ^{&}	C _{∩ij} ⁺				

[@] Número de escenario # Número secuencial ⁺ Total de coincidencias [&] Unión de ítems ^{*} $I_{ij} = (\alpha * C_{\cap ij} + \beta * A_{\cap ij} + \gamma * R_{\cap ij}) / (\alpha * C_{\cup ij} + \beta * A_{\cup ij} + \gamma * R_{\cup ij})$

[%] Marcar en caso de overlap

TABLA 2

Pág. VIII.-----
#

VERIFICACION DE OCURRENCIA DE ACTORES

ACTOR*	ESCENARIOS ^o	EPISODIOS ^o	LEL ⁺

Número secuencial ° Totales * Sin incluir los usos duplicados

+ Símbolo del LEL (Sí/No)

TABLA 2

Pág. XI.-----

#

CONTROL DE IMPACTOS VERSUS ESCENARIOS/EPISODIOS

SUJETO	IMPACTO	ESCENARIOS/EPISODIOS %	ACTORES

Número secuencial %Lista de Escenarios / Episodios

TABLA 2

Pág. XII-----
#

OBSERVACIONES

Nro. Página	Escenario	Corrección propuesta	Dudas

Número secuencial

TABLA 2

Pág. XIII-----

#

COMENTARIO GENERAL

.....
.....
.....
.....
.....
.....
.....
.....
.....

Número secuencial

Referencias Bibliográficas

- [Ackerman89] Ackerman, A. F., Buchwald, L. S., & Lewski, F. H. (1989). Software inspections: an effective verification process. *IEEE software*, 6(3), 31-36.
- [Al-Fedagui 20] Al-Fedagui, Sabah, “Modeling the Semantics of States and State Machines”, Julio 2020.
- [Aguilar88] Joyanes Aguilar, L. “Metodología de la programación. Diagramas de flujo, algoritmos y programación estructurada”. Primera edición. Editorial Mc Graw-Hill. México. p.14, 39.1988.
- [Alonso08] Pablo de Andrés Alonso, Valentín Azofra Palenzuela; “El enfoque multi stakeholder de la responsabilidad social corporativa: De la ambigüedad conceptual a la coacción y al intervencionismo”. *Revue Sciences de Gestion – Management Science – Ciencias de Gestión*, ISSN:1634-7056), nº 66, pp. 69-90, 2008.
- [Alspaugh99] Alspaugh, T.A., Antón, A.I., Barnes, T., Mott, B.W., “An Integrated Scenario Management Strategy”, *International Symposium On Requirements Engineering (RE’99)*, Limerick, Irlanda, IEEE Computer Society Press, pp.142-149, 1999.
- [Antonelli16] Antonelli, L., Rossi, G., & Oliveros A. “A Collaborative Approach to Describe the Domain Language through the Language Extended Lexicon”. *Journal of Object Technology* 15.3, 2016: 3-1, 2016.
- [Antonelli02] “Fuentes utilizadas por desarrolladores de software en Argentina para elicitar requerimientos”, L.Antonelli, A.Oliveros, *Workshop on Requirements Engineering*, Valencia, Spain , 2002.
- [Antonelli13] Leandro Antonelli, Gustavo Rossi y Julio Cesar Sampaio do Prado Leite; “Buenas prácticas en la especificación del dominio de una aplicación”; *WER* 13; 2013.
- [Atkinson00] C Atkinson, J Bayer, D Muthig, “Component-based product line development: The KobrA approach”, Springer, 2000.
- [Balzer83] Balzer, R., Cheatham, T.E., Green, C., “Software technology in the 1990s: using a new paradigm”, *IEEE Computer*, Noviembre 1983.
- [Basili75]V. Basili and J. Turner, “Iterative Enhancement: A Practical Technique for Software Development,” *IEEE Trans. Software Eng.*, pp. 390- 396, Dec.

1975.

- [Basili94] Barnard, J., Price, A.: Managing Code Inspection Information. IEEE Software. pp 59-69. 1994.
- [Basili81] Basili, V.R., Weiss, D., "Evaluation of a Software Requirements Document by Analysis of Change Data, Fifth International Conference on Software Engineering, Los Alamitos, CA, IEEE Computer Society Press, pp 314-323, 1981.
- [Beck00] Beck, K., "Extreme Programming Explained: Embrace Change", Addison-Wesley, 2000.
- [Bell76] Bell, T.E., Thayer, T. A., "Software Requirements: are they really a problem?", Second International Conference on Software Engineering, 1976.
- [Ben Achour99] Ben Achour, C., Rolland, C., Maiden, N.A.M., Souveyet, C., "Guiding Use Case Authoring: Results of an Empirical Study", International Symposium On Requirements Engineering (RE'99), Limerick, Irlanda, IEEE Computer Society Press, pp.36-43, 1999.
- [Berry02] Berry, D., "The Inevitable Pain of Software Development, Including of Extreme Programming, Caused by Requirements Volatility", International Workshop on Time-Constrained Requirements Engineering (TCRE'02), Essen, Alemania, 2002, <http://www-di.inf.puc-rio.br/~julio/tcre-site/p2.pdf>
- [Bjørner06] D. Bjørner, "Software Engineering 3, Domains, Requirements, and Software Design", Springer, ISBN-10 3-540-21151-9 Springer Berlin Heidelberg New York ISBN-13 978-3-540-21151-8 Springer Berlin Heidelberg New York, pp 107, 2006.
- [Boehm88] Boehm, B.W., "A Spiral Model of Software Development and Enhancement", IEEE Computer, Vol.21, N°5, pp.61-72, Mayo 1988.
- [Boehm66] Boehm, C. and Jacopini, G., "Flow Diagrams, Turnign Machines, and Languages with Only Two Foramation Rules." CACM, vol. 9, no. 5, pp. 366-371, 1966.
- [Boehm76] Boehm, B.W., "Software Engineering", IEEE Transaction computers,

Vol.25, N°12, pp.1226-1241, Diciembre 1976.

- [Boehm81] Boehm, B.W., “Software Engineering Economics”, Englewood Cliffs, NJ: Prentice-Hall, 1981.
- [Boehm84] Boehm, B. W. (1984). Verifying and validating software requirements and design specifications. IEEE software, 1(1), 75.
- [Booch94] Booch, G., “Scenarios”, Report on Object Analysis and Design, Vol.1, N°3, pp.3-6, 1994.
- [Booch91] “Object Oriented Design with Applications”, Booch, G., The Benjamin Cumming Publishing Company, Inc., Redwood City, 1991.
- [Booch92] Booch, G., “Object-Oriented Analysis and Design”, The Benjamin Cummings Publishing Company, Redwood City, CA, 1992.
- [Bourne92] Bourne. Object-Oriented Engineering. Irwin Inc., Homewood, Illinois, 1992.
- [Brackett90] J.W. Brackett, Software Requirements, (SEI-CM-19-1.2, ADA235642), Pittsburgh, Software Engineering Institute, Carnegie Mellon University, 1990.
- [Breitman01] Breitman, K.K., Leite, J.C.S.P., “Requirements Elicitation through Scenarios”, 5th IEEE International Symposium on Requirements Engineering (RE’01), tutorial, Toronto, Canadá, Agosto 2001.
- [Brown97] A.W. Brown; K. Short, “On components and objects: the foundations of component-based development”, Proceedings Fifth International Symposium on Assessment of Software Tools and Technologies, DOI: 10.1109/AST.1997.599921, 1997.
- [Brietman05] Brietman K, Liete J., Berry D. “Supporting Software Evolution”, Requirements Engineering, Volume 10, Issue 2, pp 112-131, May 2005.
- [Calero06] Calero, C., Ruiz, F., Piattini, M., Preface, in “Ontologies for Software Engineering and Software Technology”, Calero, C., Ruiz, F., Piattini, M., Preface, Springer, New York, 2006
- [Carroll95] Carroll, J., “Introduction: The Scenario Perspective on System Development”, en el libro Scenario-Based Design: Envisioning Work and Technology in System Development, editor J. Carroll, John Wiley & Sons,

Nueva York, pp.1-18, 1995.

[Castro01] Jaelson Castro, Manuel Kolp, and John Mylopoulos, “A Requirements-Driven Development Methodology”, K.R. Dittrich, A. Geppert, M.C. Norrie (Eds.): CaiSE 2001, LNCS 2068, pp. 108–123, Springer-Verlag Berlin Heidelberg 2001.

[CHAOS95] CHAOS Report, Standish Group, <http://www.standishgroup.com/>, Enero 1995

[Clarke02] Clarke S., Walker R. J. “Towards a standard design language for AOSD”. In: Proceedings of the 1st International Conference on Aspect-Oriented Software Development (AOSD), Enschede, The Netherlands, April 2002.

[CMMI06] Software Engineering Institute, “Capability Maturity Model Integration”, CMMI-DEV v1.2, CMU/SEI-2006-TR-008, Carnegie Mellon University, 2006, <http://www.sei.cmu.edu/cmami/>, accedida el 14-10-2007.

[Coad91] Coad, P., Yourdon, E., “Object-Oriented Analysis”, 2ª edición, Yourdon Press, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[Cockburn97] A. Cockburn, “Using Goal-Based Use Cases”, Journal of Object-Oriented Programming (JOOP), Vol.10, Nº 7, pp. 56-62, 1997.

[Cockburn02] Cockburn, A., “Agile Software Development”, Addison-Wesley, 2002.

[Cross01] Rob Cross, Andrew Parker, Laurence Prusak y Stephen Borgatti, “Knowing What We Know: Supporting Knowledge Creation and Sharing in Social Networks”, Organizational Dynamics, Vol 30, N2, pp 100-120, 2001.

[Daly77] Daly, E., “Management of Software Development”, IEEE Transactions Computers, Vol.3, Nº3, Mayo 1977.

[Dardenne93] A. Dardenne, A. van Lamsweerde, and S. Fickas, “Goal-directed requirements acquisition”, Science of Computer Programming, Vol.20, pp. 3-50, 1993.

[Davis03] A. Davis, “The Art of Requirements Triage”, IEEE Computer, Vol.36, Nº3, 2003, pp. 42-49.

[Davis83] William S. Davis, “Systems Analysis and Design: A Structured Approach”

- ISBN-13: 978-0201102710, 1983.
- [Davis93] Davis, A. M., “Software Requirements: Objects, Functions and States”, Prentice Hall, Englewood Cliffs, NJ, 2ª edición, 1993.
- [Davis99] Davis, A., Leffingwell, D., “Making Requirements Management Work For You”, Crosstalk, The Journal of Defense Software Engineering, Vol.12, Nº4, Abril 1999.
- [DEFENSE01] Defense Acquisition University, “Systems Engineering Fundamentals”, Defense Acquisition University Press, ISBN: 0160732905 EAN: 9780160732904, January 2001.
- [DeMarco78] DeMarco, Tom. “Structured Analysis and System Specification” Yourdon. ISBN 978-0-917072-07-9, 1978.
- [Deming50] Deming, W.E., “Elementary Principles of the Statistical Control of Quality”, JUSE, 1950.
- [Deming82] Deming, W.E.; 1982; “Out of the Crisis”; ISBN: 9780262535946, Editorial: The MIT Press, 2018.
- [Dietz06] Dietz, J.L.G., “Enterprise Ontology. Theory and Methodology”, Springer, New York, 2006.
- [Dijkstra68] Dijkstra, E. W., “Goto Statement Considered Harmful.” Communications of the ACM vol. 11 no. 5, pp. 147-149, May 1968.
- [DOD5000] US Department of Defense, “DoD Regulation Guidance 5000.2-R: Mandatory Procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System (MAIS) Acquisition Programs”, estados Unidos, Abril 2002.
- [Doorn98] Doorn, J., Kaplan, G., Hadad, G., Leite, J.C.S.P., “Inspección de Uso de escenarios en el Desarrollo de Software Referencias 436 escenarios”, WER’98 - Workshop de Engenharia de Requisitos, Maringá, Paraná, Brasil, 1998, pp.57-69.
- [Doorn02] Doorn, J.H., Hadad, G.D.S., Kaplan, G.N., “Comprendiendo el Universo de Discurso Futuro”, WER’02 - Workshop en Ingeniería de Requisitos, Valencia, España, pp.117-131, Noviembre 2002.

- [Doorn03] Doorn, J.H., Ridao, M., “Compleitud de Glosarios: un Estudio Experimental”, WER’03 - Workshop en Ingeniería de Requisitos, Piracicaba-SP, Brasil, pp.317-328, Noviembre 2003.
- [Drusinsky08] Drusinsky, D., Michael, J. B., & Shing, M. T. “A visual tradeoff space for formal verification and validation techniques”. IEEE Systems Journal, 2(4), 513-519. 2008.
- [Duan06] C. Duan and J. Cleland-Huang, “Visualization and Analysis in Automated Trace Retrieval”, First International Workshop on Requirements Engineering Visualization, REV’06, 2006.
- [ESD96] The World Bank Participation Sourcebook, Environmentally Sustainable Development Publication, The World Bank Washington D.C., pp 5-8, 194. 1996.
- [ESD98] Participation and Social Assessment: Tools and Techniques, Manufactured in the United States of America, ISBN 0-8213-4186-3, pp 4-5, 1998.
- [Fagan74] Fagan, M., “Design and Code Inspections and Process Control in the Development of Programs”, IBM Corporation, Reporte Técnico TR 21.572, Nueva York, Diciembre 1974.
- [Fagan76] Fagan, M.E., “Design and Code Inspections to reduce Errors in Program Development”, IBM Systems Journal, Vol.15, N°3, pp.182-211, 1976.
- [Fagan86] Fagan, M.E., “Advances in Software Inspections”, IEEE Transactions on Software Engineering, Vol.12, N° 7, pp. 744-751, 1986.
- [Faulk92] Faulk, S., et al., “The Core Method for Real-Time Requirements”, IEEE Software, Vol.9, N°5, pp.22-33, Septiembre 1992.
- [Firesmith03] Firesmith Donald, “Specifying Good Requirements”, in Journal of Object Technology, vol. 2, no. 4, July-August 2003.
- [Fitzgerald87] Brian Fitzgerald; Nancy L. Russo; Erik Stolterman, “Information Systems Development: Methods-In-Action”, Published by McGraw-Hill Education, ISBN 10: 0077098366 ISBN 13: 9780077098360, 2002.
- [Floyd84] Floyd CA, “Systematic Look at Prototyping”. In: Budde R., Kuhlenkamp K., Mathiassen L., Züllighoven H. (eds) Approaches to Prototyping.

- Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-69796-8_1,1984.
- [Freeman84] Freeman, R., “Strategic Management: A stakeholder approach”. Boston, MA: Pitman. Pitman Series in Business and Public Policy Techniques”, Prentice-Hall, Englewood Cliffs, NJ, 1979.
- [Femmer14] Femmer, H., Fernández, D. M., Juergens, E., Klose, M., Zimmer, I., & Zimmer, J. (2014, June). Rapid requirements checks with requirements smells: two case studies. In RCoSE (pp. 10-19).
- [Gane77] Gane, C., Sarson, T., “Structured Systems Analysis: Tools and Techniques”, Prentice-Hall, Englewood Cliffs, NJ, 1977.
- [GAO92] US General Accounting Office, “Mission Critical Systems: Defense Attempting to Address Major Software Challenges”, GAO/IMTEC-93-13,1992.
- [Gardner86] Gardner J., Rachlin R., Allen Sweeney H. W., “Handbook of Strategic Planning”, ISBN-13: 978-0471881278, ISBN-10: 0471881279, Wiley; 99a edición, 22 Abril 1986.
- [Gervasi05] V. Gervasi, “Reasoning about Inconsistencies in Natural Language Requirements”, ACM Transactions on Software Engineering and Methodology, Vol.14, N°3, pp. 277-330, Julio 2005.
- [Gervasi98] A. Gervasi, y B. Nuseibeh, “Managing Inconsistent Specifications: Reasoning, Analysis and Action”, ACM Transactions on Software Engineering and Methodology, Vol. 7, N° 4, pp.335–367, Octubre 1998.
- [Gilb81] T. Gilb, “Evolutionary Development,” ACM Software Eng. Notes, Apr. 1981, p. 17
- [Gilb85] T. Gilb, “Evolutionary Delivery versus the ‘Waterfall Model’,” ACM Software Requirements Eng. Notes, July 1985
- [Gladden82] Gladden, G.R., “Stop the life cycle, I want to get off”, Software Engineering Note, Vol 7, Nro 2, pp 35 ACM SIGSOFT, April 1982.
- [Glass69] R. Glass, “Elementary Level Discussion of Compiler/Interpreter Writing,” ACM Computing Surveys, Mar. 1969, pp. 64-68.
- [Goedicke 99] Goedicke, M., Meyer, T., Taentzer, G., “ViewPoint-oriented

- Software Development by Distributed Graph Transformation: Towards a Basis for Living with Inconsistencies”, IEEE International Symposium on Requirements Engineering, RE’99, Limerick-Irlanda, pp.92-99, 1999.
- [Goguen93] Goguen, J.A., Linde, Ch., “Techniques for Requirements Elicitation”, IEEE First International Symposium on Requirements Engineering, RE’93, IEEE Computer Society Press, Los Alamitos, CA, pp.152-164, 1993.
- [Gomaa81] Gomaa, H., Scott, D.B., “Prototyping as a Tool in the Specification of User Requirements”, Fifth International Conference on Software Engineering, IEEE Computer Society Press, pp.333-342, Marzo 1981.
- [Gonzalez10] Carlos Hernán González Campo, “Stakeholders: una aplicación de la teoría de los stakeholdes a los negocios electrónicos”, Estudios Gerenciales Print version ISSN 0123-5923 estud.gerenc. vol.26 no.114 Cali Jan./Mar. 2010.
- [Gotel94] Gotel, O.C.Z., Finkelstein, A.C.W., “An analysis of the requirements traceability problem”, ICRE’94, First IEEE International Conference on Requirements Engineering, IEEE Computer Society Press, Colorado Springs, pp.94-101, Abril 1994.
- [Gotel07] O.C.Z. Gotel, F. T. Marchese And S. J. Morris, “On Requirements Visualization”, Second International Workshop on Requirements Engineering Visualization, REV 2007, 2007.
- [Gottesdiener99] Gottesdiener, E., “Business Rules as Requirements”, Software Development, Vol.7, N°12, Diciembre 1999.
- [Gough95] Gough, P. A., Fodemski, F. T., Higgins, S. A., Ray, S. J., “Scenarios - An Industrial Case Study and Hypermedia Enhancements”, RE95: Proceedings of the International Symposium on Requirements Engineering, IEEE Computer Society Press, Los Alamitos, California, pp.10-17, 1995.
- [Graham94] Graham. Object-Oriented Methods. Addison- Wesley Publishing Company, London, 1994.
- [Greer99] D. Greer, D. Bustard, and T. Sunazuka, “Prioritisation of system changes using cost-benefit and risk assessments”, Fourth IEEE International Symposium on Requirements Engineering, pp. 180-187, 1999.

- [Hadad97] Hadad, G., Kaplan, G., Oliveros, A., Leite, J.C.S.P., “Construcción de escenarios a partir del Léxico Extendido del Lenguaje”, XXVI JAIIO - SoST’97 Simposio en Tecnología de Software, Buenos Aires, pp.65-77, 1997.
- [Hadad99] Hadad G., Doorn J., Kaplan G., Leite J., Enfoque Middle-Out en la Construcción e Integración de escenarios, WER 99, 1999.
- [Hadad07] Hadad, G.D.S., Doorn, J.H., “Creating Software System Context Glossaries”, Encyclopedia of Information Science and Technology, Idea Group Publishing, 2º edición, 2007.
- [Hadad08] G.D.S. Hadad, “Uso de escenarios en la Derivación de Software”. Tesis doctoral, Facultad de Ciencias Exactas de la Universidad Nacional de La Plata, Argentina, 2008.
- [Hadad09a] Hadad Graciela, Doorn Jorge y Kaplan Gladys, “Explicitar Requisitos del Software usando escenarios”, WER 09, 2009.
- [Hadad09b] Hadad Graciela, Doorn Jorge, Ridao Marcela y Kaplan Gladys “Facilitando la Asignación de Prioridades a los Requisitos”, WER 09, 2009.
- [Highsmith02] Highsmith, J., “Agile Software Development Ecosystems”, Addison-Wesley, 2002
- [Howden79] Howden, W. E. (1979). Empirical studies of software validation. *Microelectronics Reliability*, 19(1-2), 39-47.
- [IEEE610] IEEE Std 610-1990, “IEEE Standard Glossary of Software Engineering Terminology”, IEEE, Nueva York, 1990.
- [IEEE830] IEEE Std 830-1998, “IEEE Recommended Practice for Software Requirements Specifications (ANSI)”, IEEE, Nueva York, 1998.
- [IEEE1233] IEEE Std 1233, 1998 Edition, “IEEE Guide for Developing System Requirements Specifications”, IEEE Std 1233a-1998 Approved 8 December 1998
- [ISO9001] ISO9001-2015, “ISO 9001:2015 Sistema de Gestión de Calidad”, 2015.
- [Jackson94] Jackson, M., “The Role of Architecture in Requirements Engineering”, ICRE’94, First International Conference on Requirements Engineering, IEEE Computer Society, Colorado Springs, p.241, Abril 1994.
- [Jackson95] Jackson, M., “Software Requirements & Specifications. A lexicon of

- practice, principles and prejudices”, Addison-Wesley, Reading, MA/ACM Press, Nueva York, 1995.
- [Jacobson92] Jacobson, I., Christerson, M., Jonsson, P., Overgaard, G., Reading, “Object-oriented Software Engineering - A Use Case Driven Approach”, MA: Addison Wesley, Nueva York: ACM Press, 1992.
- [Jacobson99] Jacobson, I., Booch, G., Rumbaugh, J., “The Unified Software Development Process”, Addison-Wesley, Reading, MA, 1º edición, Febrero 1999.
- [Jalote05] Pankaj Jalote, “An Integrated Approach to Software Engineering”, Third Edition, Springer, ISBN-10: 0-387-20881-X(HB) ISBN-10: 0-387-28132-0 (eBook) ISBN-13: 978-0387-20881-7 (HB) ISBN-13: 978-0387-28132-2 (eBook), 2005.
- [Jarke98] Jarke, M., “Requirements tracing”, Communications of the ACM, Vol.41, Nº12, pp.32-36, Diciembre 1998.
- [Kantorowitz97] Kantorowitz, E. Guttman, A., Arzi, L. “The Performance of the N-Fold Inspection Method”, Requirements Engineering Journal, Vol.2, Springer-Verlag London, p.p.152-164, 1997.
- [Kaplan00] Kaplan, G.N., Hadad, G.D.S., Doorn, J.H., Leite, J.C.S.P., “Inspección del Léxico Extendido del Lenguaje”, WER’00 – III Workshop de Engenharia de Requisitos, Río de Janeiro, Brasil, pp.70- 91, Julio 2000.
- [Kaplan06] Kaplan Gladys, Hadad Graciela y Doorn Jorge, “Notas de Clases sobre Ingeniería de Requisitos”. Apunte publicado en UTN/UNLaM/UNO. 2006.
- [Kaplan08] Kaplan Gladys, Doorn Jorge, Hadad Graciela, “Una visión cognitiva de Modelos de Requisitos de Software”, WICC 08, 2008.
- [Kaplan09] Gladys N. Kaplan, Jorge H. Doorn, “Handling Extemporaneous Information in Requirements Engineering”, “Encyclopedia of Information Science & Technology”, IGI Global Books (Mehdi Khosrow-Pour, Information Resources Management Association, USA), Second Ed., Vol 2, pp. 789-794. Total de Páginas 5266. ISBN: 978-1-60566-026-4, 2009.
- [Kaplan10] Kaplan Gladys, Doorn Jorge, Hadad Graciela; “Ambigüedad Léxica en los Modelos de Requisitos en Lenguaje Natural”, WICC, 2010.

- [Kaplan11] Gladys Kaplan, Jorge Doorn, Graciela Hadad, “Granularidad de la Información Extemporánea en los Procesos de Requisitos”, WIIC, 2011.
- [Kaplan13a] Gladys Kaplan, Jorge Doorn, Nora Gigante; “Construcción del LEL de Requisitos”, WICC 13, 2013.
- [Kaplan13b] Kaplan G., Doorn J., Gigante, N., “Evolución Semántica de los Glosarios en los Procesos de Requisitos”, CACIC14. 2013.
- [Kaplan14a] Kaplan Gladys, Tesis de Maestría “Información Extemporánea en los Procesos de Requisitos”, Director Jorge Doorn. Universidad Nacional de La Matanza, Defendida y aprobada en Septiembre de 2014.
- [Kaplan14b] Gladys Kaplan, Jorge Doorn, Nora Gigante, “Elicitación de Conocimiento Guiada por Modelos”, CACIC, UNLaM, 2014.
- [Kaplan15] Kaplan G., Doorn J., “Advanced & Delayed Information in Requirements Engineering” in “Encyclopedia of Information Science & Technology”, IGI Global Books (Mehdi Khosrow-Pour, Information Resources Management Association, USA), Third Ed. Cantidad de Páginas 10384 Vol. 10, ISBN13: 9781466658882, ISBN10: 1466658886, EISBN13: 9781466658899, 2015.
- [Kaplan17] Kaplan Gladys y Doorn Jorge, “Jerarquías Naturales en el Contexto del Proceso de Requisitos”, In Proceedings of WER’17: IXX Workshop de Engenharia de Requisitos, 2017.
- [Kaplan19] Kaplan G., Doorn J, “Similitudes y Diferencias entre el Análisis de Sistemas y la Ingeniería de Requisitos”, REDDI- UNLaM Revista Digital del Dpto de Ingeniería e Investigaciones Tecnológicas, 2019.
- [Kaplan20] Kaplan G., Doorn J., “Reducción de la Subjetividad en los procesos de requisitos”, WICC20, 2020.
- [Kaplan20] Kaplan G., Tua D., Blanco G, “Desagregación de la información en los escenarios futuros”, WICC20, 2020.
- [Kaplan21] Kaplan G., Meghetti J.P, Blanco G, “Ingeniería de Requisitos para Organizaciones enfocadas en los Procesos”, CACIC 2021, 2021.
- [Kazhamiakin04] Kazhamiakin, R., Pistore, M., & Roveri, M. (2004, September). Formal verification of requirements using spin: A case study on web

- services. In Proceedings of the Second International Conference on Software Engineering and Formal Methods, 2004. SEFM 2004. (pp. 406-415). IEEE.
- [Kendall97] Kendall y Kendall, “Análisis y Diseño de Sistemas”, Tercera Edición, Prentice Hall, México, 1997.
- [Kitchenham07] Kitchenham, B. and Charters, S. Guidelines for Performing Systematic Literature Reviews in Software Engineering, Technical Report EBSE 2007-001, Keele University and Durham University Joint Report. 2007.
- [Kłosowski18] P. Kłosowski, "Deep Learning for Natural Language Processing and Language Modelling", Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), pp. 223-228, Poznan, 2018.
- [Kotonya98] G. Kotonya, y I. Sommerville, “Requirements Engineering: Processes and Techniques”, John Wiley & Sons, 1998.
- [Kovitz98] Kovitz, B.L., “Practical Software Requirements: A manual of Content and Style”, Greenwich, CT: Manning Publications Co., Octubre 1998.
- [Lamsweerde98] A. van Lamsweerde, “Inferring Declarative Requirements Specifications from Operational Scenarios”, IEEE TSE, Vol.24, Nº12, pp. 1089-1114. , 1998
- [Lapeña19] Lapeña R., Pérez F., Cetina C., Pastor Ó. “Improving Traceability Links Recovery in Process Models Through an Ontological Expansion of Requirements”. In: Giorgini P., Weber B. (eds) Advanced Information Systems Engineering. CAiSE 2019. 2019.
- [Larman03] Craig Larman, Victor R. Basili, “Iterative and Incremental Development: A Brief History”, pp. 47-56, vol. 36, DOI Bookmark: 10.1109 / MC.2003.1204375, IEEE, June 2003.
- [Leite89] Leite, J.C.S.P., “Application Languages: A Product of Requirements Analysis”, Computer Science Department of PUC-Rio, Brazil, 1989.
- [Leite90] Leite, J.C.S.P., Franco, A.P.M., “O Uso de Hipertexto na Elicitação de Linguagens da Aplicação”, Anais de IV Simpósio Brasileiro de Engenharia de Software, SBC, SBC, pp 134-149, 1990.

- [Leite97] Leite, J.C.S.P., Rossi, G., Balaguer, F., Maiorana, V., Kaplan, G., Hadad, G., Oliveros, A., “Enhancing a Requirements Baseline with Scenarios”, Requirements Engineering Journal, Springer-Verlag London Ltd., Vol.2, N°4, pp.184-198, 1997.
- [Leite98] Leite J.C.S.P, Leonardi, M.C., “Business rules as organizational Policies”, IEEE Ninth International Workshop on Software Specification and Design, IEEE Computer Society Press, pp.68-76, 1998.
- [Leite00] Leite, J.C.S.P., Hadad, G.D.S., Doorn, J.H., Kaplan, G.N., “ Scenario Construction Process”, Requirements Engineering Journal, Springer-Verlag London Ltd., Vol.5, N°1, pp. 38-61, 2000.
- [Leite 01] Leite, J.C.S.P., “Gerenciando a Qualidade de Software com Base em Requisitos”, en el libro *Qualidade de Software: Teoria e Prática*, editores A.R.Rocha, J.C. Maldonado y K.C. Weber, Prentice-Hall, San Pablo, capítulo 17, 2001, pp.238-246.
- [Leite03] Leite J., Doorn J., Hadad G., Kaplan G., Using Scenario Inspections on Different Scenarios Representations, *Monografias em Ciência da Computação*, Departamento de Informática, PUC-Rio, 2003.
- [Leite03b] Leite and Breitman, “Experiences Using Scenarios to Enhance Traceability”, 2nd International Workshop on Traceability in Emerging Forms of Software Engineering TEFSE’ 03, 2003.
- [Leite04] “Perspectives on Software Requirements: An introduction”, Leite, J.C.S.P., Doorn, J.H., en el libro “Perspectives on Software Requirements”, Kluwer Academic Publishers, EEUU, ISBN: 1-4020-7625-8, Capítulo 1, 2004.
- [Leite05] J.C.S.P. Leite, J.H. Doorn, G.D.S. Hadad, y G.N. Kaplan, “Scenario Inspections”, Requirements Engineering Journal, Vol.10, N°1, Springer-Verlag, pp. 1-21, 2005.
- [Li12] Li Haomin & Wu Jianhua, “Civil Aircraft System Integration Research”, The 2nd International Conference on Computer Application and System Modeling, 2012.
- [Litvak13] Litvak CS, Hadad GDS, Doorn JH, “Mejoras semánticas para estimar la

- Completitud de Modelos en Lenguaje Natural”, Primer Congreso Nacional de Ingeniería Informática / Sistemas de Información, Córdoba, 2013.
- [Loucopoulos95] Loucopoulos, P., Karakostas, V., “System Requirements Engineering”, McGraw-Hill, Londres, 1995.
- [Lutz93] Lutz, R., “Analyzing Software Requirements Errors in Safety-Critical Embedded Systems”, IEEE First International Symposium on Requirements Engineering, RE’93, IEEE Computer Society Press, Los Alamitos, CA, pp.126-133, 1993.
- [Marca89] David A. Marca and Clement L. McGowan, “SADT: Structured Analysis and Design Technique”. McGraw-Hill. ISBN 0-07-040235-3, 1989.
- [Martin88] James Martin and Carma McClure, “Structured Techniques: The Basis for Case”, Prentice Hall; Edición: Revised, Subsequent, ISBN-10: 0138549362, ISBN-13: 978-0138549367,1988.
- [Martin90] Martin, J., Tsai, W.T., “N-fold Inspection: A Requirements Analysis Technique”, Communications of the ACM, Vol.33, Nº2, 1990, pp.225- 232.
- [Martin92] J. Martin, J. Odell. “Object-Oriented Analysis and Design”. Prentice Hall, Englewood Cliffs, New Jersey, 1992.
- [Martin93] Martin, James; Odell, James, “Principles of Object-Oriented Analysis and Design”, Prentice Hall; Edición: 1st Edition ISBN-10: 0137208715, ISBN-13: 978-0137208715, ASIN: B000UZBX5Q,1993
- [Maté04] Maté J.L., Silva A., “Requirements Engineering for Sociotechnical Systems”, Information Science Publishing, ISBN: 1-59140-506-8, pp 2, pp 319, 2004.
- [McCracken80] Daniel McCracken y Michael Jackson, “A minority dissenting position” in “Systems analysis and design” William W. Cotterman, Institute for Certification of Computer Professionals, IEEE Computer Society, Association for Computing Machinery, 1980
- [McMenamin84] Stephen M. McMenamin y John F. Palmer, “Essential Systems Analysis”, Prentice Hall PTR, ISBN-10: 0132879050, ISBN-13: 978-0132879057, 1984
- [Meilir88] Meilir Page-Jones, “Practical Guide to Structured Systems Design”,

- Prentice Hall, (2nd Edition), ISBN-13: 978-8120314825. ISBN-10: 8120314824, 1988.
- [Mille13] Mille, S., Burga, A., Wanner, L. “AnCora-UPF: A multi-level annotation of Spanish”. Publisher: Charles University in Prague, Matfyzpress, Prague, Czech Republic, Volume: Proceedings of the Second International Conference on Dependency Linguistics, pp. 217, 2013.
- [Mills87] Mills, Harlan D.; Dyer, M.; and Linger, R. C., "Cleanroom Software Engineering". The Harlan D. Mills Collection. 1987.
- [Mylopoulos92] J. Mylopoulos, L. Chung, and B.A. Nixon, “Representing and Using Nonfunctional Requirements: A Process-Oriented Approach”, IEEE TSE, Vol.18, Nº6, pp. 483-497, 1992.
- [Nguyen08] Nguyen, N. T. “Advanced methods for inconsistent knowledge management”. Springer, London, 2008.
- [Nentwich03] Nentwich, C., Emmerich, W., Finkelstein, A., Ellmer, E. “Flexible consistency checking”. ACM Transaction on Software Engineering and Methods, 12(1), pp. 28-63, 2003.
- [Nassi73] I. Nassi and B. Shneiderman; “LOWCHART TECHNIQUES FOR STRUCTURED PROGRAMMING”, technical contributions, 1973.
- [Ober98] Ober, R., Probasco, L, Ericsson, M., “Applying Requirements Management with Use Cases”, Rational Software Corporation, 1998.
- [Oliveros12] Oliveros, J. Zuñiga, R. Wehbe, S. Rojo, y F. Sardi, “Enseñanza de elicitation de requerimientos”, Congreso Argentino de Ciencias de la Computación (CACIC), Bahía Blanca, 2012.
http://sedici.unlp.edu.ar/bitstream/handle/10915/23852/Documento_completo.pdf?sequence=1&isAllowed=y
- [Palmer96] Palmer, J.D., “Traceability”, en Software Engineering, editores M. Dorfman y R.H. Thayer, IEEE Computer Society Press, 1996, pp.266- 276. Reimpreso en “Software Requirements Engineering”, editores R.H. Thayer y M. Dorfman, IEEE Computer Society Press, 2ª edición, Los Alamitos, CA, pp.364-374, 1997.
- [Parviainen05] Parviainen, P., Tihinen, M., van Solingen, R., “Requirements

- Engineering: dealing with the Complexity of Sociotechnical Systems Development”, in Requirements Engineering for Sociotechnical Systems, J.L. Maté & A. Silva (eds.) , Chapter I, pp1-20, 2005, ISBN 1-59140-506-8.
- [Park99] J. Park, D. Port, and B. Boehm, “Supporting Distributed Collaborative Prioritization for Win-Win Requirements Capture and Negotiation”, International Third World Multi-conference on Systemics, Cybernetics and Informatics (SCI'99), Vol.2, Orlando, FL, pp. 578-584, 1999.
- [Perls75] Frederick S. Perls, "Dentro y fuera del tarro de la basura", Editorial Cuatro Vientos, Chile, 1975.
- [Pinheiro04] Francisco A. C. Pinheiro, “REQUIREMENTS TRACEABILITY”, en el libro “Perspectives on Software Requirements”, Kluwer Academic Publishers, EEUU, ISBN: 1-4020-7625-8, Capítulo 5, 2004.
- [Pinheiro96] Pinheiro, F.A.C., Goguen, J.A., “An object-oriented tool for tracing requirements”, IEEE Software, Special issue of papers from ICRE'96, Vol.13, N°2, pp.52-64,1996.
- [Porter95] Porter, A.A., Votta, Jr., L.G., Basili, V.R., “Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment”, IEEE TSE, Vol.21, N°6, pp.563-575, 1995.
- [Pouloudi99] Pouloudi, A.. “Aspects of the stakeholder concept and their implications for information systems development”, Proceedings 32nd Annual Hawaii International Conference on System Sciences (HICSS'99).1999.
- [Potts94] Potts, C., Takahashi, K., Antón, A. I., “Inquiry-Based Requirements Analysis”, IEEE Software, Vol.11, N°2, pp.21-32 , 1994.
- [Potts95] C. Potts, “Using Schematic Scenarios to Understand User Needs”, DIS'95-Symposium on Designing Interactive Systems: Processes, Practices and Techniques, ACM Press, University of Michigan, pp. 247-256, 1995.
- [Pressman05] Roger Pressman, “Ingeniería de Software. Un enfoque práctico”, McGraw Hill, Sexta Edición, 2005.
- [Regnell99] Regnell, B., “Requirements Engineering with Use Cases – a Basis for Software Development”, Ph.D. Thesis, Reporte Técnico 132, Introducción,

- Department of Communication Systems, Lund University, pp.7-42, 1999.
- [Ridao01] Ridao, M., “Uso de Patrones en el Proceso de Construcción de escenarios”, tesis de Maestría en Ingeniería de Software, Facultad de Informática, Universidad Nacional de La Plata, Noviembre 2001. Director: Prof. Jorge H.Doorn (Universidad del Centro de la Provincia de Buenos Aires), 2001.
- [Ridao06] Ridao, M., Doorn, J.H. “Estimación de Completitud en Modelos de Requisitos Basados en Lenguaje Natural”. En: IX Workshop on Requirements Engineering (WER’06), Brasil, pp. 151—158, 2006.
- [Ridao20] Ridao M., Doorn J., Kaplan G., “Desagregación de la información en los escenarios futuros”, WICC20, 2020.
- [Riel96] Arthur J. Riel, “Object-Oriented Design Heuristics”, Addison-Wesley Professional; Edición: 1, ISBN-10: 020163385X, ISBN-13: 978-0201633856, 1996.
- [Rolland98] Rolland, C., Souveyet, C, Ben Achour, C., "Guiding Goal Modeling Using Scenarios", IEEE TSE, Vol.24, N°12, pp.1055–1071, 1998.
- [Ross 97] Ross, R., “The Business Rule Book: Classifying, Defining and Modeling Rules”, Business Rule Solutions, LLC, 2º edición, 1997.
- [Ross76] D. Ross, K. Schoman Jr., “Structured analysis for Requirements Definition”, IEEE Computer Society Press Los Alamitos, CA, USA ©1976.
- [Royce70] Royce, W.W., “Managing the Development of Large Software Systems: concepts and techniques”, IEEE WESCON, Los Angeles, CA, Agosto 1970.
- [Rumbaugh05] Rumbaugh, J., Jacobson, I., Booch, G., “The Unified Modeling Language Reference Manual”, Addison-Wesley, Reading, MA, 2º edición, 2005.
- [Rumbaugh91] Rumbaugh, J., M. Blaha, W. Premlani, F. Eddy, W. Lorensen, “Object-Oriented Modeling and Design”, Englewood Cliffs, NJ: Prentice Hall International, 1991.
- [RUP98] Rational Software, “RUP: Rational Unified Process”, IBM Corporation, <http://www-306.ibm.com/software/rational>, 1998.
- [Ryan14] Ryan, Mike, “On a Useful Taxonomy of Phases, Modes, and States in Systems Engineering”, Abril 2014.

- [Sawyer01] Sawyer, P., Kotonya, G., “Software Requirements”, SWEBOK, Guide to the Software Engineering Body of Knowledge, editores P. Bourque y R. Dupuis, IEEE Computer Society, Los Alamitos, CA, Capítulo 2, 2001, pp.31-56, http://www.swebok.org/stoneman/trial_1_00.html, accedida el 20-02-2005.
- [Schneider98] Schneider, G., Winters, J., “Applying Use Cases, A Practical Guide”, Addison-Wesley, Reading, MA, 1998.
- [Schneider92] Schneider, G., Martin, J., Tsai, W.T., “An experimental study of fault detection in user requirements documents”, ACM Transaction on Software Engineering and Methodology, Vol.1, Nº2, pp.188-204, 1992.
- [Shewhart39] Shewhart, W. A. 1939. Statistical Method from the Viewpoint of Quality Control. Department of Agriculture. Dover, 1986, page 45
- [Sebastian15] Alberto Sebastián, Graciela Hadad, Mejoras a un Modelo Léxico mediante Mapas Conceptuales, CACIC 2015 – XXI Congreso Argentino de Ciencia de la Computación, At Universidad Nacional del Noroeste, Junín, Buenos Aires, Argentina. 2015.
- [Senn91] James Senn, “Análisis y Diseño de Sistemas de Información”, Segunda Edición, McGraw Hill, 1991.
- [Sessions92] Roger Sessions, “Class Construction in C and C++: Object-Oriented Programming Fundamentals”, 1 marzo 1992.
- [Sharp99] Helen Sharp, Anthony Finkelstein & Galal Galal; “Stakeholder Identification in the Requirements Engineering Process, pp 1, http://eprints.ucl.ac.uk/744/1/1.7_stake.pdf”, 1999.
- [Smith98] Smith, Barry, “Basic concepts of formal ontology”, en N. Guarino (ed.), “Formal ontology in information systems”, 1998, IOS Press, pp.19-28
- [Simula67] Dahl and Nygaard at the IFIP TC 2 Working Conference on Simulation Languages in Lysebu near Oslo, 1967.
- [Sommerville07] Sommerville, I., “Ingeniería de Software”, Addison-Wesley, pp 20-122, 7ª edición, 2007.
- [Sommerville11] Ian Sommerville, “Ingeniería de Software”, Pearson, Edition 9, 2011.

- [Standish94] The CHAOS report. Disponible en https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf
- [Sternberg99] Sternberg, Elaine, “The Stakeholder Concept: A Mistaken Doctrine. Foundation for Business Responsibilities”, Issue Paper No. 4, November 1999, Available at SSRN: <https://ssrn.com/abstract=263144> or <http://dx.doi.org/10.2139/ssrn.263144>
- [Strode05] Strode, D.E., The Agile Methods: An Analytical Comparison of Five Agile Methods and an Investigation of Their Target Environment, MSc Thesis in Information Systems, Massey, University, Palmerstin North, Nueva Zelanda, 2005.
- [Takoshima15] Takoshima, A., & Aoyama, M. (2015, December). Assessing the quality of software requirements specifications for automotive software systems. In 2015 Asia-Pacific Software Engineering Conference (APSEC) (pp. 393-400). IEEE.
- [Thayer97] Thayer R.H., Dorfman, M., “Software Requirements Engineering”, IEEE Computer Society Press, 2ª edición, Los Alamitos, CA, 1997
- [Ting15] Ting M., Kadir R.A., Sembok T.M.T., Ahmad F., Azman A. “Adaptive Learning for Lemmatization in Morphology Analysis”. In: Fujita H., Selamat A. (eds) Intelligent Software Methodologies, Tools and Techniques. SoMeT 2014. Communications in Computer and Information Science, vol 513. Springer, Cham, 2015.
- [Unhelkar05] Unhelkar B., Verification and validation for quality of UML 2.0 models, vol. 42. John Wiley & Sons, 2005.
- [Uzair08] Uzair Akbar Raja and Kashif Kamran, “Framework for Requirements Traceability - TLFRT supporting pre-RS & post-RS traceability”, tesis de Maestría, Suiza, 2008
- [Vasilecas06] Vasilecas, O., Bugaite D., Trinkunas, J. On Approach for Enterprise Ontology Transformation into Conceptual Model, International Conference on Computer Systems and Technologies - CompSysTech'06.

2006

- [Varzi94] Varzi, Achille: “On the boundary between mereology and topology” en “Philosophy and the cognitive sciences. Proceedings of the 16th international Wittgenstein symposium”, Viena, Hölder-Pichler-Tempsky, pp. 423-442 , 1994.
- [Visconti07] Visconti, M., Cook, C., An Ideal Process Model for Agile Methods, LNCS, ISBN 978
- [Wang19] Wang, Fei and Yang, Zhibin and Huang, Zhi-Qiu and Liu, Cheng-Wei and Zhou, Yong and Bodeveix, Jean-Paul and Filali, Mamoun An Approach to Generate the Traceability Between Restricted Natural Language Requirements and AADL Models. (2019) IEEE Transactions on Reliability, 1 (1). 1-20. ISSN 0018-9529
- [Weidenhaupt98] Weidenhaupt, K., Pohl, K., Jarke, M., Haumer, P., “Scenarios in System Development: Current Practice”, IEEE Software, 1998, pp.34- 45.
- [Wetherbe84] James C. Wetherbe, “Systems Analysis and Design: Traditional, Structured, and Advanced Concepts and Techniques”. West Pub. Co. ISBN 0-314-77858-6
- [Whitenack94] Whitenack, B.G. Jr., “RAPPEL: A Requirements Analysis Process Pattern Language for Object Oriented Development”, Knowledge Systems Corp., 1994.
- [Whitten03] Whitten,J., Bentley,L., Dittman, K., “Systems Analysis and Design Methods”, Mc Graw-Hill / Irwin, 6ª edición, 2003.
- [Wieringa95] Wieringa, R.J., “An introduction to requirements traceability”, Reporte Técnico IR-389, Faculty of Mathematics and Computer Science, University of Vrije, Amsterdam, Septiembre 1995.
- [Winkler10] Winkler, S., von Pilgrim, J. “A survey of traceability in requirements engineering and model-driven development”. Softw Syst Model 9, 529–565 (2010). <https://doi.org/10.1007/s10270-009-0145-0>. 2010
- [Wirfs-Brock90] Wirfs-Brock, R., Wilkerson, B., Wiener, L., “Designing Object Oriented Software”, Prentice-Hall, Englewood Cliffs, N.J., 1990.
- [Wirfs-Brock95] Wirfs-Brock, R., “Designing Objects and Their Interactions: A Brief

- Look at Responsibility-Driven Design”, Scenario-Based Design: Envisioning Work and Technology in System Development, editor J. Carroll, John Wiley, Nueva York, 1995, pp. 337-359.
- [Wirth71] Niklaus Wirth, “Program development by stepwise refinement” , ACM New York, NY, USA , Volume 14 Issue 4, April 1971, Pages 221-227
- [Wohlrab18] Wohlrab R., Pelliccione P., Knauss E., Gregory S.C., “The Problem of Consolidating RE Practices at Scale: An Ethnographic Study”. In: Kamsties E., Horkoff J., Dalpiaz F. (eds) Requirements Engineering: Foundation for Software, 2018.
- [Wohlrab20] Wohlrab, R., Knauss, E., Steghöfer, J. et al. “Collaborative traceability management: a multiple case study from the perspectives of organization, process, and culture. Requirements”, Eng 25, 21–45. <https://doi.org/10.1007/s00766-018-0306-1>, 2020
- [Wood89] Wood, J., Silver, D. “Joint Application Design: How to Design Quality Systems in 40% Less Time”, Computer Science, 1989.
- [Wood95] Wood, J. and D. Silver, “Joint Application Development”, Wiley and Sons, Inc. New York, 1995.
- [Young04] Ralph Young, “The Requirements Engineering Handbook”, ISBN: 1-58053-266-7, pp.1.2, pp.49-55, Artech House, 2004.
- [Yourdon75] Yourdon, E., Constantine, L., “Structured Design”, Yourdon Press, 1975
- [Yourdon78] Edward Yourdon, Larry L. Constantine, "Structured Design: Fundamentals of la Discipline of Computer Program and Systems Design", Yourdon Press, N.Y., 1978.
- [Yourdon79] Yourdon E., “Structured Analysis and System Specification”, ISBN-13: 978-0138543808, ISBN-10: 0138543801, Prentice Hall; Edición 1, 1979
- [Yourdon88] Yourdon E., “Modern Structured Analysis”, Prentice Hall; Edición 1, ISBN-10: 0135986249, ISBN-13: 978-0135986240, 1988
- [Yu93] Yu, E., “Modeling organizations for information systems requirements engineering”, IEEE First International Symposium on Requirements Engineering, IEEE Computer Society Press, San Diego, Enero 1993, pp.34-

41.

[Yu95] Yu, E., “Modeling Strategic Relationships for Process Reengineering”, Ph.D. thesis, Department of Computer Science, University of Toronto, Canada, 1995.

[Zave97] Zave, P., Jackson, M., “Four Dark Corners of Requirements Engineering”, ACM Transactions on Software Engineering and Methodology, Vol.6, Nº1, 1997, pp.1-30.

[Zave84] Zave, P., “The Operational Versus the Conventional Approach to Software Development”, Communications of the ACM, 27, 104-118, 1984.

[Zorman95] Zorman, L., “Requirements Envisaging by Utilizing Scenarios (Rebus)”, Ph.D. Dissertation, University of Southern California, 1995.