



UNIVERSIDAD DE ORIENTE  
NÚCLEO DE SUCRE  
ESCUELA DE CIENCIAS  
DEPARTAMENTO DE MATEMÁTICAS  
PROGRAMA DE LA LICENCIATURA EN INFORMÁTICA

ESPECIFICACIÓN DE PROCESOS DE DESARROLLO DE SOFTWARE A  
TRAVÉS DEL METAMODELO DE INGENIERÍA DE PROCESOS DE  
SOFTWARE (SPEM)  
(Modalidad: Investigación)

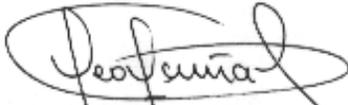
Charli José Vívenes Rengel

TRABAJO DE GRADO PRESENTADO COMO REQUISITO PARCIAL PARA  
OPTAR AL TÍTULO DE LICENCIADO EN INFORMÁTICA

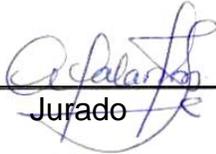
CUMANÁ, 2012

ESPECIFICACIÓN DE PROCESOS DE DESARROLLO DE SOFTWARE A  
TRAVÉS DEL METAMODELO DE INGENIERÍA DE PROCESOS DE  
SOFTWARE (SPEM)

APROBADO POR:



Ing. Leopoldo Acuña  
Asesor Académico



Jurado



Jurado

## DEDICATORIA

En el transcurso de mi vida y mi andar por la Universidad, un grupo de personas fueron un apoyo y motivadores a los que debo gran parte de lo que hoy soy y he logrado alcanzar. A ustedes dedico las siguientes líneas.

En primer lugar, a Jehová Dios; por enseñarme que con Él “haremos proezas” (Salmos 60.12). A su hijo Jesucristo, nuestro Señor y Salvador, por dar su vida a cambio de la nuestra (Juan 3. 16). Junto al Espíritu Santo son la fortaleza y guía para mi vida.

A mis padres, por hacer de mí un hombre de bien a través de los principios que se empeñaron en enseñarme.

Los amigos que hallé en la Universidad: Misael Ferrer, Ingrid Vargas, Beira Vargas, Migdalis Mago, Inés Rondón, Hecnellys Bastardo y, de manera especial a Jessica Hernández y a su madre, la Sra. Eslinda Hernández, por haber tendido su mano para que pudiera continuar y alcanzar esta meta; sin ustedes hubiera sido más difícil.

A mi profesor y amigo, asesor original de este Trabajo de Grado, quien por problemas de salud no pudo acompañarme hasta el final de la meta. Esto se lo debo a Usted y espero en Dios que vuelva a ser el mismo de antes.

Todos los profesores que, tal vez inintencionadamente, fueron ejemplos a seguir. Porque día a día instruyen a quienes decidimos, por voluntad propia o la de Dios, estudiar esta carrera.

## AGRADECIMIENTOS

El desarrollo de este trabajo fue posible gracias al apoyo de las siguientes personas, a quienes deseo agradecer.

Ing. Rafael Caldera, de quien nació la idea original de abordar este tema para mi Trabajo de Grado, por su dirección en todo instante en la mayor parte del trabajo realizado. Por ser además un amigo y consejero y una de las personas de quien he aprendido lecciones invaluable durante el tiempo de compartir el desarrollo del trabajo y las enseñanzas en las aulas de clase.

Ing. Leopoldo Acuña, por su influencia como profesor de algunas cátedras y por brindarme su ayuda para finalizar con esta etapa... Estoy muy agradecido.

Per Kroll, líder del proyecto *Eclipse Process Framework*; Fabio A. Zorzan, de la Universidad Nacional de Río Cuarto en Argentina; Francisco Ruiz González de la Universidad de Castilla-La Mancha y Ma. Angélica Caro Gutiérrez de la Universidad del Bio Bio en Chile, autora del método de análisis y revisión de la literatura (incorporado en una de las fases de la investigación), por responder cada correo electrónico.

## ÍNDICE

	Pág.
DEDICATORIA .....	I
AGRADECIMIENTOS .....	II
ÍNDICE .....	III
LISTA DE TABLAS .....	VII
LISTA DE FIGURAS .....	VIII
LISTA DE ABREVIATURAS .....	XI
RESUMEN .....	XIII
INTRODUCCIÓN .....	1
CAPÍTULO I. PRESENTACIÓN .....	6
PLANTEAMIENTO DEL PROBLEMA .....	6
HIPÓTESIS Y OBJETIVOS DE LA INVESTIGACIÓN .....	9
Hipótesis .....	9
Objetivo general.....	10
Objetivos específicos.....	10
ALCANCE Y LIMITACIONES DE LA INVESTIGACIÓN .....	10
Alcance .....	10
Limitaciones.....	12
CAPÍTULO II. MARCO DE REFERENCIA.....	15
MARCO TEÓRICO.....	15
Antecedentes de la investigación .....	15
Ingeniería de Software.....	16
Definiciones del término “Ingeniería de Software”.....	17
Áreas de conocimiento de la Ingeniería de Software .....	19
Proceso de ingeniería de software .....	22
Definición de proceso.....	25
Modelos del ciclo de vida del software.....	26

Procesos del ciclo de vida del software .....	26
Proceso de desarrollo de software .....	27
Modelos de procesos de software .....	31
Especificación y/o modelado de procesos de software .....	32
Lenguajes de modelado de procesos .....	34
Meta-modelo de Ingeniería de Procesos de Software .....	36
Características del SPEM .....	40
Mecanismos de trabajo con SPEM .....	49
Meta-modelo SPEM .....	51
Arquitectura general del Meta-modelo SPEM .....	52
Organización de un Repositorio SPEM.....	55
Contenido de Método.....	56
Elementos de Contenido ( <i>Content Element</i> ).....	57
Propiedades principales.....	64
Asociaciones.....	66
Procesos .....	67
Aspectos Generales.....	68
Elementos de Desglose de Trabajo.....	69
Tipos de Procesos .....	76
Patrón de Proceso ( <i>Capability Pattern</i> ) .....	77
Proceso para Despliegue ( <i>Delivery Process</i> ) .....	77
Reutilización y Variabilidad .....	78
Variabilidad de Elementos de Contenido .....	79
No asignado ( <i>not assigned</i> ).....	79
Contribuye ( <i>contributes</i> ) .....	80
Reemplaza ( <i>replaces</i> ).....	82
Amplía ( <i>extends</i> ).....	83
Amplía y Reemplaza ( <i>extends-replaces</i> ) .....	85
Reutilización y Variabilidad de Elementos de Proceso .....	87

No asignado ( <i>not assigned</i> ).....	88
Extensión ( <i>extension</i> ) .....	88
Contribución local ( <i>local contribution</i> ).....	88
Reemplazo local ( <i>local replacement</i> ).....	89
Configuraciones de Método .....	90
MARCO METODOLÓGICO .....	91
Métodos de investigación en Ingeniería de Software .....	93
Metodología para la investigación en Ingeniería de Software .....	96
Búsqueda de documentación.....	96
Determinación del problema .....	97
Creación de la hipótesis.....	98
Definición del método de trabajo.....	98
Resolución, validación y verificación.....	99
Análisis de resultados y elaboración de conclusiones .....	99
Metodología para el análisis y revisión de la literatura .....	100
Planificación de la revisión.....	102
Desarrollo de la revisión.....	102
Publicación de los resultados.....	102
Estudio de caso .....	103
La creatividad como método de investigación .....	104
Justificación del uso del Proceso Unificado como caso de estudio .....	105
CAPÍTULO III. DESARROLLO.....	106
BÚSQUEDA DE LA DOCUMENTACIÓN.....	106
Planificación la revisión .....	107
Identificación de la necesidad de la revisión .....	107
Definición de un protocolo de búsqueda .....	108
Definición de un protocolo de revisión .....	111
Evaluación de la planificación .....	112
Desarrollo de la revisión .....	112

Búsqueda de estudios primarios .....	113
Selección de estudios primarios.....	113
Extracción de datos.....	113
Síntesis de datos.....	114
Publicación de los resultados .....	114
DETERMINACIÓN DEL PROBLEMA .....	114
CREACIÓN DE LA HIPÓTESIS .....	116
DEFINICIÓN DEL MÉTODO DE TRABAJO.....	117
RESOLUCIÓN, VERIFICACIÓN Y VALIDACIÓN .....	118
Revisión y comprensión de la especificación del SPEM.....	119
Selección del caso de estudio .....	119
Elección del mecanismo de implementación de SPEM .....	119
Selección de herramientas y configuración del ambiente de trabajo ...	120
Implementación del SPEM sobre el caso de estudio .....	123
Realizar pruebas.....	124
ANÁLISIS DE RESULTADOS Y ELABORACIÓN DE CONCLUSIONES	124
CAPÍTULO IV. ESPECIFICACIÓN DEL CASO DE ESTUDIO.....	125
ASPECTOS GENERALES .....	125
Disciplinas del proceso .....	125
Roles en el proceso .....	127
Productos del proceso .....	129
Instrucciones.....	131
Tareas.....	132
Propósito y descripción de los Elementos de método .....	134
Relaciones entre Roles, Tareas y Productos de trabajo .....	134
Patrones de proceso.....	139
ASPECTOS SOBRE TRABAJADORES .....	141
Categorización de roles .....	141
ASPECTOS SOBRE PRODUCTOS .....	142

Categorización de los productos.....	142
ASPECTOS SOBRE PROCESOS, ACTIVIDADES Y TAREAS.....	143
ASPECTOS SOBRE LA PRESENTACIÓN.....	145
Configuración de método.....	145
CONCLUSIONES .....	150
RECOMENDACIONES.....	155
BIBLIOGRAFÍA.....	157
APÉNDICES .....	162
ANEXOS.....	217

## LISTA DE TABLAS

	Pág.
Tabla 1. Identificación de la necesidad de revisión.....	108
Tabla 2. Protocolo de búsqueda.....	110
Tabla 3. Protocolo de revisión.....	<b>¡Error! Marcador no definido.</b>
Tabla 3. Protocolo de revisión (Continuación) .....	112

## LISTA DE FIGURAS

	Pág.
Figura 1. Primeras cinco AC del SWEBOK.....	21
Figura 2. Últimas cinco AC del SWEBOK.....	22
Figura 3. Desglose de temas del AC Proceso de Ingeniería de Software ....	24
Figura 4. Marco conceptual de uso de SPEM.....	37
Figura 5. Idea básica de proceso en SPEM.....	41
Figura 6. Niveles de modelado MOF. ....	41
Figura 7. Aspectos principales para especificar PDS con SPEM.....	44
Figura 8. Separación del Contenido de método y Proceso.....	45
Figura 9. Conceptos para representar la jerarquía de desglose de trabajo. .	46
Figura 10. Ejemplo del mecanismo de <i>plug-ins</i> en SPEM. ....	48
Figura 11. Ejemplo de componente de proceso en SPEM. ....	49
Figura 12. Niveles de modelado y ejemplos de instancias.....	50
Figura 13. Representación gráfica de procesos con el perfil UML de SPEM.	51
Figura 14. Estructura del Meta-modelo SPEM.....	52
Figura 15. Ejemplo de jerarquía para organizar un repositorio SPEM.....	56
Figura 16. Jerarquía de conceptos del <i>Method Content</i> .....	57
Figura 17. Ejemplo de asociaciones de una Tarea.....	59
Figura 18. Jerarquía de conceptos de Proceso.....	68
Figura 19. Flujo de trabajo y anidamiento de actividades entre WBE.....	70
Figura 20. Relación entre Actividades y Roles y Productos de Trabajo.....	72
Figura 21. Ejemplo de una actividad y sus asociaciones.....	72
Figura 22. Ejemplo de Contenido de Método en Uso. ....	75
Figura 23. Ejemplos de variabilidad de tipo “contribuye”. ....	81
Figura 24. Resultado de la variabilidad de la Figura 23. ....	81
Figura 25. Ejemplo de variabilidad de tipo “reemplaza”.....	83

Figura 26. Resultado de la variabilidad de la Figura 25 .....	83
Figura 27. Ejemplo de variabilidad de tipo “amplía” .....	84
Figura 28. Resultado de la variabilidad de la Figura 27 .....	85
Figura 29. Ejemplo de variabilidad de tipo “amplía y reemplaza”.....	86
Figura 30. Resultado de la variabilidad de la Figura 29.....	87
Figura 31. Ejemplo de una configuración de método (elementos en rojo) ....	91
Figura 32. Propuesta metodológica de Marcos (2002) .....	97
Figura 33. Método de Caro y cols. (2005).....	101
Figura 34. Modelo de dominio para la determinación del problema.....	115
Figura 35. Método para la resolución del problema .....	118
Figura 36. Ambiente de trabajo.....	121
Figura 37. <i>UML Profile for SPEM para Enterprise Architect 6.5</i> .....	123
Figura 38. Estructura del Contenido de método para el UP.....	127
Figura 39. Roles abstractos en el <i>Content Package</i> “común”.....	128
Figura 40. Roles del UP en sus respectivos Paquetes de contenido.....	129
Figura 41. Productos de trabajo abstractos del <i>Content Package</i> “común”.131	
Figura 42. Instrucciones en el <i>Content Package</i> “comun”. .....	132
Figura 43. Tareas de la disciplina de Diseño. ....	133
Figura 44. <i>Disciplines</i> del <i>Method Content</i> para el UP.....	133
Figura 45. Propósito y Descripción principal de una tarea.....	134
Figura 46. Productos de trabajos de los que un rol es responsable. ....	135
Figura 47. Asignación de un rol a una categoría definida. ....	135
Figura 48. Asignación de pasos a una tarea.....	136
Figura 49. Asignación de productos de trabajo a una tarea.....	137
Figura 50. Asignación de guías a una tarea. ....	137
Figura 51. Asignación de una tarea a una categoría ( <i>Disciplines</i> ). ....	138
Figura 52 .Relación entre tareas y productos de trabajo con un rol.....	138
Figura 53. Patrones de proceso para el UP.....	139
Figura 54. WBS del patrón de proceso “Análisis”. ....	140

Figura 55. Estructura del patrón “Iteración de la Fase de Inicio [1..n]” .....	140
Figura 56. Categorización ( <i>Role Sets</i> ) de los roles del UP. ....	141
Figura 57. Categorización de productos de trabajo. ....	142
Figura 58. Descripción de la Configuración de método “my.up”. ....	145
Figura 59. Organización del contenido para la especificación del UP. ....	147
Figura 60. Selección de Paquetes y <i>Plug-in</i> para especificación del UP. ...	147
Figura 61. Vista de navegación de la especificación del UP. ....	148
Figura 62. <i>Homepage</i> de la especificación resultante para el UP. ....	149

## LISTA DE TABLAS

Tabla 1. Identificación de la necesidad de revisión.....	108
Tabla 2. Protocolo de búsqueda. ....	110
Tabla 3. Protocolo de revisión.....	111

## LISTA DE ABREVIATURAS

CASE	<i>Computer Aided Software Engineering</i>
CMMI	<i>Capability Maturity Model Integration</i>
HTML	<i>HyperText Markup Language</i>
IBM	<i>International Business Machines</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ISO	<i>International Organization for Standardization</i>
MDA	<i>Model-Driven Architecture</i>
MOF	<i>Meta Object Facility</i>
OCL	<i>Object Constraint Language</i>
OMG	<i>Object Management Group</i>
PML	<i>Process Modeling Language</i>
RUP	<i>Rational Unified Process</i>
SPEM	<i>Software Process Engineering Metamodel</i>
SWEBOK	<i>Software Engineering Body of Knowledge</i>
UML	<i>Unified Modeling Language</i>
UP	<i>Unified Software Development Process</i>
XML	<i>eXtensible Markup Language</i>
XP	<i>eXtreme Programming</i>

## RESUMEN

Con el desarrollo de este trabajo se logró sistematizar una experiencia en la aplicación de principios y tecnologías inherentes a la Ingeniería de Software (IS) en el contexto del Metamodelo de Ingeniería de Procesos de Software (SPEM, *Software Process Engineering Meta-model*) a través de su implementación en el Proceso Unificado de Desarrollo de Software (UP, *Unified Process*), introducido como caso de estudio que permitió ensayar una alternativa válida para la definición, especificación y mantenimiento de métodos, metodologías y/o procesos de desarrollo de software impulsados como resultados provenientes de proyectos de investigación que desarrollan miembros del Centro de Ingeniería de Software del Programa de la Licenciatura en Informática del Núcleo de Sucre de la Universidad de Oriente; contribuyendo además al fortalecimiento de las actividades de docencia e investigación en la academia. Para lograr este objetivo se adaptó una propuesta metodológica para la investigación en IS, que plantea cinco etapas, incorporando otros métodos y tareas necesarias y combinando éstos con técnicas, modelos, lenguajes y herramientas de software para enriquecer y complementar los resultados de este trabajo. En la etapa de búsqueda de la documentación se obtuvo el basamento teórico, sistematizado en el marco referencial que sustenta este trabajo; la etapa de determinación del problema permitió precisar concretamente el universo de discurso para la investigación, representado a través de un modelo de dominio expresado en UML; en la etapa de creación de la hipótesis se especificó en lenguaje natural las características deseables de la especificación del caso de estudio elegido que se implementó a través de los elementos conceptuales del SPEM; en la etapa de definición del método de trabajo se establecieron cada una de las actividades que permitirían lograr la especificación del caso de estudio haciendo uso de la experiencia adquirida y la creatividad como herramientas que permitieron configurar todo el proceso de investigación; finalmente, la etapa de resolución, verificación y validación consistió en la puesta en práctica de las actividades anteriormente establecidas que permitieron especificar, verificando y validando, los requisitos impuestos en la etapa de creación de la hipótesis.

## INTRODUCCIÓN

El desarrollo de software es un proceso orgánico complejo, así como un proceso iterativo de aprendizaje, en el que intervienen una serie de factores que deben combinarse de manera armónica para obtener como resultado un producto de calidad y usable. La Ingeniería de Software (IS) es el marco que comprende todos los aspectos de este proceso e integra entre dichos factores: un conjunto de métodos, técnicas y herramientas de software, entre otros, con el fin de desarrollar productos nuevos o mejorar los existentes, por lo que uno de sus objetivos de décadas ha sido establecer directrices y principios que sean sistemáticos, predecibles y repetibles a fin de mejorar la productividad y la calidad (Pressman, 2002).

Este conjunto de principios y directrices establecidos por la IS es lo que se conoce como proceso de desarrollo de software (procesos de software o simplemente procesos, PDS). En este sentido, se entiende un PDS como el conjunto de actividades y pautas de trabajo parcialmente ordenadas que permiten transformar las necesidades de los usuarios en un producto software. El proceso define “quién” está haciendo “qué”, “cuándo”, y “cómo” alcanzar un objetivo determinado (Jacobson y cols., 2000) y es la base que establece el contexto en el cual se aplican los métodos de la IS (Pressman, 2002).

Hasta la fecha, conviven una variedad de propuestas para abordar el desarrollo de software, permitiendo clasificar de manera general entre procesos ágiles o pesados, según sea el número de productos de trabajo (artefactos) consumidos y generados (así como la documentación relacionada) en cada actividad del proceso. Empero la forma en que se

especifican tales procesos difiere de acuerdo con el enfoque y/o con el creador de los mismos. En el contexto de los sistemas basados en computadoras (y en software), el término “especificación” tiene significados diferentes para personas distintas; una especificación puede ser un documento escrito en lenguaje natural, un conjunto de modelos gráficos, un modelo matemático formal, una colección de escenarios de uso, un prototipo o cualquier combinación de éstos (Pressman, 2005). En correspondencia con los PDS, una especificación puede entenderse como una descripción de los aspectos estructurales y de comportamiento de un proceso (Botella y cols., 2004). Miers (citado por García y Rodríguez, 2001) concuerda en que “la descripción de un proceso puede tomar muchas formas” y añade que “es posible usar un lenguaje gráfico (PML, *Process Modeling Language*) para exponerlo dado que los modelos gráficos, ayudan a representar el proceso en estudio al disminuir la complejidad, propiciar un entendimiento común entre los participantes y permitir el estudio de alternativas”.

Sin embargo, pese a la existencia de múltiples PML, no coexiste una terminología común entre los diferentes enfoques. Además se utilizan notaciones que aparentan ser propias del creador de los mismos y en muchos casos se auxilian de notaciones propias del modelado de software, ocasionando redundancias e inconsistencias en las especificaciones de las diferentes propuestas de PDS. La existencia de notaciones, terminologías y/o significados diferentes para las mismas palabras o palabras distintas con un mismo significado (p. e., una “fase” de *Fusion* es llamada “*core workflow*” en el RUP (*Rational Unified Process*) y “dominio” en el Método de Servicios Globales de IBM) dificulta la comunicación entre los miembros de un equipo de desarrollo, propiciando una situación similar a la existente antes del esfuerzo de unificación de algunos de los métodos de desarrollo y notaciones empleadas para el modelado de software que dan paso al Lenguaje de

Modelado Unificado (UML, *Unified Modeling Language*) a finales de los 90 (Rumbaugh y cols., 2000).

Es así como en reconocimiento a la necesidad de unificación de la terminología y notaciones utilizadas para la especificación de PDS, comienzan a surgir propuestas provenientes de instituciones y organismos de estandarización como el OMG (*Object Management Group*), quien promueve SPEM (*Software Process Engineering Metamodel*) como un PML estándar para la especificación de procesos, fomentando una visión y una cultura común y de esta forma proveer una solución a la necesidad planteada (OMG, 2005).

De esta manera, la aparición del SPEM representa un objeto de estudio relevante para el Centro de Ingeniería de Software (CIS) del Programa de la Licenciatura en Informática del Núcleo de Sucre de la Universidad de Oriente, debido a la falta de dominio conceptual y tecnológico de los mecanismos inherentes a la implementación de algún PML. El SPEM es una alternativa válida a tomar en cuenta para la especificación adecuada de métodos, metodologías y/o PDS impulsados como resultados provenientes de proyectos de investigación que actualmente desarrollan integrantes del CIS, entre los que se cuentan profesores y estudiantes regulares y tesisistas que desarrollan actividades de investigación. Por esta razón, el objetivo de este trabajo está centrado en el estudio y comprensión del SPEM, con la finalidad de sistematizar el conocimiento adquirido para ser aplicado posteriormente en diferentes trabajos de investigación que se adelantan desde el CIS, además de la incorporación de los resultados al programa de estudio y actividades de investigación de la academia.

Desde el punto de vista científico y de acuerdo con Arias (2006) y Sabino

(1978; 1994; 2000), el nivel y el diseño de investigación abordados para el desarrollo de este trabajo son exploratorio-descriptivo y de campo, respectivamente, porque se propone alcanzar una visión general del objeto de estudio obteniendo los datos directamente de la realidad; a través de la experiencia empírica con el objeto de estudio; estableciendo y configurando una marco tecnológico de trabajo que permitió la realización de la experiencia.

Para lograr el objetivo descrito, la investigación se desarrolló siguiendo los principales lineamientos de la propuesta de Marcos (2002) para realizar investigación en IS. Se configuró cada una de las fases de la propuesta metodológica incluyendo otros métodos, añadiendo tareas complementarias e incorporando un caso de estudio, representado por el Proceso Unificado de Desarrollo de Software (UP, *Unified Process*), sobre el cual se implementan los elementos conceptuales y notacionales del SPEM, permitiendo además enriquecer la metodología utilizada. De esta manera se obtienen, conjuntamente, otros resultados relevantes como la especificación del UP que se enriquece al incorporar la notación del SPEM; la redefinición y sistematización de la estrategia metodológica para la realización de investigación en IS (sin precedentes en nuestra academia), y el dominio tecnológico y conceptual del SPEM plasmado como parte del marco teórico de este trabajo.

En cuanto a la organización de este informe de Trabajo de Grado, su contenido se presenta en cuatro (4) capítulos. En el Capítulo I se aborda el planteamiento del problema, se establecen los objetivos, el alcance y las limitaciones de la investigación realizada. En el Capítulo II, se expone el marco teórico que presenta los antecedentes y el basamento conceptual de la investigación y el marco metodológico que presenta cada uno de los

métodos que componen la metodología de investigación aplicada. El Capítulo III corresponde al desarrollo de la investigación propiamente dicha; se presentan los resultados obtenidos en cada una de las fases de la metodología adoptada para el desarrollo del trabajo y cada uno de los productos de trabajo generados en cada actividad de la misma. En el Capítulo IV se expone el proceso de especificación del caso de estudio a través del SPEM. Finalmente, se presentan las conclusiones obtenidas, se realizan algunas recomendaciones, se incluye la bibliografía con las obras referenciadas y los apéndices y anexos que serán de utilidad al lector.

# **CAPÍTULO I. PRESENTACIÓN**

## **PLANTEAMIENTO DEL PROBLEMA**

Cuando se desarrolla un producto software es importante seguir una serie de pasos predecibles que ayuden a obtener resultados de calidad y a tiempo, por lo que reconocidos autores han coincidido en que es necesario contar con métodos, metodologías y/o PDS para guiar a los desarrolladores conforme los productos se hacen cada día más grandes y complejos. Al respecto, Montilva (1992) afirma que este aspecto es uno de los elementos esenciales para el desarrollo exitoso de sistemas de información (SI) basados en computadoras.

Empero, pese a que en la actualidad existe una variedad de propuestas para abordar el desarrollo de software, también existe una diversidad de lenguajes y notaciones para describirlas, lo que dificulta la comunicación entre los miembros de un equipo de desarrollo que “hablan idiomas distintos” debido a la existencia de notaciones, terminologías y/o significados diferentes para las mismas palabras; propiciando una situación similar a la existente antes del esfuerzo de unificación que dio paso al UML. Muchos de los métodos de desarrollo de software que se describen en libros de texto, artículos, material de capacitación, estándares, regulaciones y otras formas de documentación, usualmente se exponen y representan de diferentes maneras y en algunos casos se proporciona información no muy clara a todo aquel que los utiliza; sin hacer referencia explícita a cómo llevarlos a cabo y pareciera que la experticia del lector fuera el condicionante del proceso.

Se puede constatar que existen propuestas como las planteadas por Lloréns

(1991), Senn (1992), Kendall y cols (1997) y Pressman (2002) que se han especificado a través del lenguaje natural, con el que se pierde poder expresivo y pueden quedar sin tratar –o quedar sólo vagamente descritos– muchos detalles que pueden ser precisados con notaciones formales. Por otro lado, existen propuestas descritas con notaciones aparentemente propias de cada autor que incluyen símbolos gráficos auxiliares como en Whitten y cols. (2003) o utilizan Diagramas de Flujo de Datos propios de la técnica de Análisis Estructurado de Sistemas de Montilva (1982). Adicionalmente, otros casos como METRICA versión 3 (2001), el RUP y otras metodologías y/o PDS se basan en un conjunto de ideas y conceptos subyacentes, que no están definidos de manera explícita, es decir, no es evidente el meta-modelo subyacente utilizado. Además, el formato en que se maneja la información sobre tales procesos son documentos de texto en lenguaje natural. Esto tiene la gran desventaja de que toda la manipulación (creación, revisión, reutilización, adaptación, etc.) y generación de documentación (publicación) para dichas metodologías es puramente manual.

Investigaciones realizadas en el ámbito académico indican que los estudiantes de programas de licenciatura entienden poco el significado de "programar en grande"; es decir, aplicar los principios de la IS al desarrollo de un producto por un equipo de personas, durante el cual la coordinación efectiva entre los participantes permite lograr un sistema de software exitoso (Upchurch y Sims-Knight, 1997). Esta situación, asociada a la confusión e interpretación incorrecta de propuestas cuyas especificaciones no obedecen a notaciones estándares, que ofrezcan precisión y claridad, dificulta la comunicación y el aprendizaje. En el caso concreto del Programa de la Licenciatura en Informática del Núcleo Sucre de la UDO, donde existe un amplio historial y experiencia en el desarrollo de sistemas de software (por

parte de estudiantes tesistas en sus Trabajos de Grado, así como por estudiantes regulares en las cátedras respectivas del programa de estudio) se carece del conocimiento de lineamientos y/o estándares para la especificación adecuada de procesos y/o metodologías de desarrollo de software que puedan proporcionarse en su currículo y líneas de investigación. Así mismo, tampoco existen métodos de desarrollo de software propios basados en mencionada experiencia, que puedan ser reutilizados, mantenidos y enriquecidos con resultados de desarrollos posteriores.

Algunos síntomas y manifestaciones de la problemática planteada se evidencian en la interpretación incorrecta de las metodologías y/o PDS utilizados para el desarrollo de SI, producto de los vacíos existentes en las especificaciones en lenguaje natural de los mismos; la ausencia de retroalimentación y de enriquecimiento a través de los resultados obtenidos en investigaciones, trabajos de grado y proyectos de fin de curso desarrollados en las cátedras; el establecimiento de caminos inadecuados que se degradan con el paso del tiempo y la falta de sistematización y síntesis de los resultados obtenidos. Todo esto demuestra que, una de las principales necesidades de la academia es la comprensión y el dominio tecnológico y conceptual de los mecanismos necesarios para la definición y especificación adecuada de metodologías y/o PDS.

En este sentido, es así que para las líneas de investigación que se adelantan desde el CIS de la UDO, el dominio y la experiencia en el uso de lenguajes estándares para el modelado y especificación adecuada de PDS representa un objeto de estudio relevante. Por tanto, este trabajo de investigación se enfoca en el estudio y comprensión del SPEM como una alternativa válida para la especificación de procesos; atendiendo a la recomendación del OMG, para su posterior aplicación en la descripción de los resultados de

investigación y propuestas provenientes del CIS. Al mismo tiempo se contribuye a fortalecer las actividades de investigación y docencia del Programa de la Licenciatura en Informática, impulsando el desarrollo dentro y desde la academia; acercándose progresivamente al conocimiento y dominio de las tecnologías inherentes a la IS. De esta manera, se realiza la implementación del SPEM en la especificación del UP como caso de estudio para lograr la sistematización de una experiencia en la aplicación de los principios, conceptos y tecnologías en el contexto del SPEM.

## **HIPÓTESIS Y OBJETIVOS DE LA INVESTIGACIÓN**

### **Hipótesis**

Este trabajo de grado es un trabajo científico experimental no en el sentido estricto de que busque verificar el cumplimiento, en el mundo físico o natural que nos rodea, de una hipótesis definida previamente. Se trata de un trabajo de ingeniería y, por tanto, la hipótesis formulada se refiere directamente a la posibilidad de sistematizar la experiencia de implementación del objeto de estudio a través de la incorporación de un caso de estudio que permita alcanzar este objetivo y poder así diseñar, construir o mejorar un artefacto (físico o mental) que ayuda a las personas a resolver un problema o a minimizarlo. En este sentido, la hipótesis de estudio ha sido la siguiente:

Es posible sistematizar la experiencia de implementación del Meta-modelo de Ingeniería de Procesos de Software (SPEM) a través de la incorporación de un caso de estudio que permita adquirir conocimiento sobre el mismo para ser tenido en cuenta como una alternativa válida para su aplicación en la especificación de métodos, metodologías y/o PDS que resulten de investigaciones provenientes del CIS de la UDO.

## **Objetivo general**

El objetivo general de la investigación es lograr la especificación de un PDS mediante la aplicación de los principios, conceptos y tecnologías en el contexto del SPEM y de esta manera poder sistematizar la experiencia para su posterior aplicación en la descripción de los resultados de investigación provenientes del CIS de la UDO.

## **Objetivos específicos**

Para la consecución del objetivo general se plantearon los siguientes objetivos específicos:

- Revisión y comprensión de la especificación del SPEM.
- Seleccionar el caso de estudio.
- Elegir el mecanismo de implementación del SPEM.
- Seleccionar la herramienta software de soporte y configurar el ambiente de trabajo.
- Implementar el SPEM sobre el caso de estudio.
- Realizar pruebas.

## **ALCANCE Y LIMITACIONES DE LA INVESTIGACIÓN**

### **Alcance**

El desarrollo de este trabajo se enmarca dentro de una serie de proyectos de investigación que adelantan miembros del CIS y que han recibido apoyo y

aprobación del Consejo de Investigación de la UDO para su ejecución. De manera específica, forma parte esencial de la labor de investigación que viene realizándose desde hace varios años, bajo la coordinación del asesor del presente trabajo, y que ha dado lugar a otros proyectos de trabajo de grado.

En este contexto, se presenta la especificación de un caso de estudio, representado por el UP, utilizando el SPEM bajo el mecanismo de meta-modelado porque permite representar métodos, metodologías y/o PDS con la ventaja de que no sólo abarca la representación, sino también las definiciones semánticas y restricciones estructurales permitiendo la creación de bibliotecas de métodos de modo que se cuente con un repositorio en el cual se tenga almacenada toda la información de experiencias pasadas que se puedan reutilizar y adaptar a nuevos proyectos. Como herramienta para la especificación se utilizó *Eclipse Process Framework (EPF) Composer*, desarrollada bajo la plataforma Eclipse que implementa el SPEM bajo el mencionado mecanismo; con la ventaja añadida de ser una herramienta de código abierto.

En este sentido, se logra la sistematización de una experiencia en la aplicación de tecnologías en el contexto del SPEM, que permitirá su consideración como alternativa válida a tomar en cuenta para la posterior aplicación en la especificación de métodos, metodologías y/o PDS resultantes de trabajos de investigación provenientes del CIS de la UDO; así como la consideración de este tema de estudio para ser incorporado en el programa de estudio del Programa de la Licenciatura en Informática, contribuyendo de esta manera a fortalecer la investigación y docencia, así como impulsar el desarrollo dentro y desde la academia; acercándose progresivamente al conocimiento y dominio de las tecnologías inherentes a la

IS.

### **Limitaciones**

El desarrollo de este trabajo constituyó un reto importante para el autor del mismo, y como en todo reto hubo que enfrentar dificultades y problemas que pueden considerarse como limitaciones que estuvieron presentes en el desarrollo de la investigación.

La percepción que el autor ha tenido del problema planteado sufrió modificaciones y ha evolucionado a lo largo del tiempo. Las características del problema abordado, el método de trabajo seguido y, sobre todo, el propio proceso de aprendizaje por descubrimiento han hecho que los planteamientos y puntos de vista sobre este trabajo no hayan permanecido constantes durante los años que ha durado su desarrollo. Desde la idea original, planteada por el asesor del trabajo, se han producido cambios y evoluciones significativos. Sería largo y tedioso explicar cada paso de dicha evolución, pero puede resumirse diciendo que conforme se ha ido estudiando cada vez más profundamente el problema, se ha ido descubriendo (en realidad confirmando) que es complejo y multidisciplinario. En un momento determinado se acordó que el problema se debía tratar desde una perspectiva parcial. De ahí se derivó la necesidad de la incorporación de un caso de estudio.

Inicialmente no era claro el camino a seguir para realizar investigación en IS. Por ello, parte del proceso de investigación consistió en la búsqueda, estudio y configuración de una estrategia de investigación que pudiera ser adaptada a este trabajo, con el objeto de tener una guía para la consecución de los objetivos propuestos.

Así mismo la ausencia de estudios y experiencias previas en el Programa de la Licenciatura en Informática de la UDO sobre el tema abordado y la falta de conocimiento previo sobre el problema abordado, por parte del autor, hicieron necesario la investigación de fuentes bibliográficas que, en su parte más importante, se encuentran disponibles vía Web en los sitios de publicación de autores y grupos de investigación relevantes de universidades e instituciones fuera de nuestras fronteras. Por otro lado, es de sobra conocido que el mundo de la informática tiene al inglés como primera lengua; en esa lengua se desarrollan y publican la mayoría de los trabajos y la mayor parte de la información técnica que manejan sus profesionales.

Otra de las limitaciones presente en todo el proceso de estudio, fue la falta de una buena conexión a Internet en la Red Académica de la UDO de la que se beneficia al Programa de la Licenciatura en Informática y específicamente las instalaciones donde tuvo lugar la mayor parte del desarrollo del trabajo inicial de investigación. Esto condujo a la inversión de una buena parte del tiempo que esperó este trabajo para ver la luz; puesto que el acceso a los documentos, publicaciones y artículos de investigación disponibles en los sitios web de sus autores y la descarga de las herramientas de software eran bastante deficientes; lo que supone paciencia debido a los problemas recurrentes que genera una conexión a Internet con tales características (p. e., archivos y descargas corrompidas).

La utilización de las herramientas de software propietarias (o privativas, atendiendo a los argumentos de la comunidad del software libre) fue restringida, debido a que las descargas se encuentran disponibles como versiones de prueba que no incluyen toda su potencialidad y establecen un tiempo de uso corto que resulta insuficiente para la práctica.

Finalmente, la especificación del UP obtenida como uno de los productos de trabajo en esta investigación no contempla actividades de planificación y gestión propias de todo PDS ya que no es uno de los objetivos de este trabajo y que la misma especificación original del UP proporcionada en Jacobson y cols. (2000) no las contempla, dejándolas implícitas.

## CAPÍTULO II. MARCO DE REFERENCIA

### MARCO TEÓRICO

#### Antecedentes de la investigación

La experiencia en la realización de investigación en IS, a través de la redefinición y sistematización de una estrategia metodológica, así como el dominio tecnológico y conceptual del SPEM plasmados en este informe de Trabajo de Grado, no tienen precedentes en el Programa de la Licenciatura en Informática del Núcleo Sucre de la UDO que puedan ser tomados como punto de partida para apoyar el trabajo aquí desarrollado. Sin embargo, a través del proceso de búsqueda de documentación presente en la estrategia general definida, se pudo comprobar que SPEM está siendo ampliamente utilizado en la industria y la academia para especificar PDS e incluso poder automatizarlos. Algunos trabajos en los que se ha utilizado y que se referencian como antecedentes para este trabajo particular se mencionan a continuación.

En la Universidad Nacional de Río Cuarto, Romero y Uva (2005), propusieron una alternativa para la optimización de los PDS mediante la automatización de las metodologías de desarrollo. Para lograr este objetivo utilizaron como caso de estudio una instancia del RUP, denominada *SmallRUP*, definida por Pollice (2003) en su artículo “*Using the IBM Rational Unified Process for small projects: Expanding upon eXtreme Programming*”, la cual fue especificada en SPEM, para transformar los procesos en procesos *workflow* de manera que pudieran ser ejecutados, como cualquier proceso automatizado, por un sistema *workflow*.

En la misma línea de investigación, también en la Universidad Nacional de Río Cuarto, Riesco y Romero (2004) propusieron automatizar la producción del software a través de un *workflow* que siguiera las reglas definidas en el RUP, especificadas a través del SPEM.

Otro antecedente lo representa el proyecto Vulcano o de “Promoción del desarrollo de SW libre en un entorno de calidad y confianza adaptando las metodologías, procesos, modelos de negocio y últimas tecnologías” desarrollado en España por Larrucea y cols. (2007). El objetivo del proyecto Vulcano fue la integración de herramientas para proveer un entorno de colaboración que permitiera la creación de software libre de calidad optando por SPEM como el meta-modelo para la definición de metodologías.

En un contexto geográfico más cercano, en la Universidad Central de Venezuela (UCV), Chirinos (2003) desarrolló un proceso para especificar, medir y controlar la calidad de productos de software durante su desarrollo para su tesis doctoral. Para especificar la calidad propuso un modelo de proceso expresado en SPEM, que toma en cuenta los requisitos no funcionales desde las primeras etapas del desarrollo y hace énfasis en la identificación, descripción, cuantificación, validación/verificación y gerencia de los requisitos de calidad.

## **Ingeniería de Software**

La IS es una disciplina o área de la Informática o Ciencias de la Computación, que ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo. Hoy día es cada vez más frecuente la consideración de la *Zngenierh de software* como una nueva área de la ingeniería, y comienza a ser una profesión implantada

internacionalmente, con derechos, deberes y responsabilidades que cumplir, junto a una ya reconocida consideración social en el mundo empresarial con brillante futuro.

Aunque no existe consenso, la noción de “Ingeniería de software” fue propuesta inicialmente en octubre de 1968, fecha en que se celebró la primera conferencia organizada por la OTAN (Organización del Tratado del Atlántico Norte) para tratar lo que entonces se llamó la “crisis del software” (Naur y Randell, 1969) adjudicándose a Fritz Bauer el uso del término (aun cuando previamente había sido utilizado por Edsger Dijkstra en su obra “*The Humble Programmer*”). Básicamente, la crisis del software se refiere a la dificultad de escribir grandes programas libres de defectos, fácilmente comprensibles, y que sean verificables. Las causas son, entre otras, la complejidad que supone la tarea de programar, y los cambios a los que se tiene que ver sometido un producto software para ser continuamente adaptado a las necesidades de los usuarios.

De esta manera nació formalmente la IS, como una disciplina que trata con áreas muy diversas de la informática y de las ciencias de la computación, tales como construcción de compiladores, sistemas operativos o desarrollos en Intranet/Internet, abordando todas las fases del ciclo de vida del software (CVS) para el desarrollo de cualquier tipo de producto [software] y aplicables a infinidad de áreas tales como: negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, derecho, la red de redes Internet, redes Intranet y Extranet, etc.

Definiciones del término “Ingeniería de Software”

Una definición precisa aún no ha sido contemplada en los diccionarios; tal

vez por tratarse de una ingeniería muy nueva en comparación con la trayectoria de otras ingenierías. El Diccionario de la Real Academia Española (DRAE) define ingeniería como: “1. Conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía. 2. Profesión y ejercicio del ingeniero”. De igual manera define software como “Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora”. Sin embargo, no enuncia una definición de IS; se podría exponer una definición partiendo de las dos dadas: “Conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización y construcción de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora”. No obstante, antes de adoptar esta acepción como segura, es conveniente realizar una breve recopilación de definiciones hechas por autores acreditados e instituciones de prestigio (aquí se destacan algunas, de entre las muchas propuestas).

Bauer (citado por Naur y Randall, 1969) en la conferencia de la OTAN sobre la materia expresó que la IS “es el establecimiento y uso de sólidos principios de ingeniería a fin de obtener software de modo rentable, que sea fiable y trabaje en máquinas reales”.

Para Bohem (1976), IS 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.

Sommerville (2005) define la IS como “una disciplina de la ingeniería que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de

éste después que se utiliza”.

La *IEEE Computer Society* (1993) ha elaborado una definición más comprensible de la IS al establecer que es “(1) La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software. (2) El estudio de enfoques como en (1)”.

### Áreas de conocimiento de la Ingeniería de Software

Se puede afirmar que desde 1968 se han hecho enormes progresos y que el desarrollo de esta ingeniería ha mejorado considerablemente el software. Se comprende mejor las actividades involucradas en el desarrollo de software. Se han desarrollado métodos efectivos de especificación, diseño e implantación del software. Las nuevas notaciones y herramientas de soporte reducen el esfuerzo requerido para producir sistemas grandes y complejos. A través de los años, la IS ha promovido el desarrollo de software usable y de calidad, logrando alcanzar grandes avances en todas sus áreas (análisis de requisitos, estrategias de implementación, entre otras). A medida que la importancia del software ha crecido, la IS ha desarrollado tecnologías que hacen más sencilla, rápida y menos costosa la construcción y mantenimiento del mismo. Algunas de estas tecnologías se limitan al dominio de una aplicación específica (por ejemplo, al diseño e implementación de sitios Web) y otras se enfocan en el dominio de paradigmas de desarrollo (como la Programación Orientada a Objetos, la Programación Orientada a Aspectos y la Arquitectura Dirigida por Modelos) pero todas persiguen el objetivo de mejorar los procesos de producción del software (Pressman, 2005).

Recientemente, en virtud del estatus de disciplina legítima y profesión

reconocida y la crucial importancia de un consenso hacia el estado profesional de un cuerpo de conocimientos, la *IEEE Computer Society* (2004) creó el Cuerpo de Conocimiento de la Ingeniería del Software (SWEBOK, *Software Engineering Body of Knowledge*). El SWEBOK es un documento que se define como una guía al conocimiento presente en el área de la IS y supone un paso esencial hacia el desarrollo de la profesión porque representa un amplio consenso respecto a los contenidos de la disciplina, con los siguientes objetivos:

- Promover una visión consistente de la IS en todo el mundo.
- Clarificar el ámbito, y establecer las fronteras, de la IS con respecto a otras disciplinas, tales como las Ciencias de la Computación o las Matemáticas.
- Caracterizar los contenidos de la IS como disciplina.
- Proveer acceso a través de las temáticas al conjunto de conocimientos de la IS.
- Proveer una base para su desarrollo curricular y la creación de materiales de certificación.

En el SWEBOK se recoge el conocimiento generado durante los ya más de cuarenta años de existencia de la IS (iniciada en 1968), definiendo diez (10) áreas de conocimiento (AC) para delimitar la IS (esquemáticas en las Figuras 1 y 2):

- Requisitos de software.
- Diseño de software.
- Construcción de software.
- Pruebas de software.
- Mantenimiento de software.
- Gestión de la configuración de software.

- Gestión de ingeniería de software.
- Proceso de ingeniería de software.
- Herramientas y métodos de ingeniería de software.
- Calidad de software.

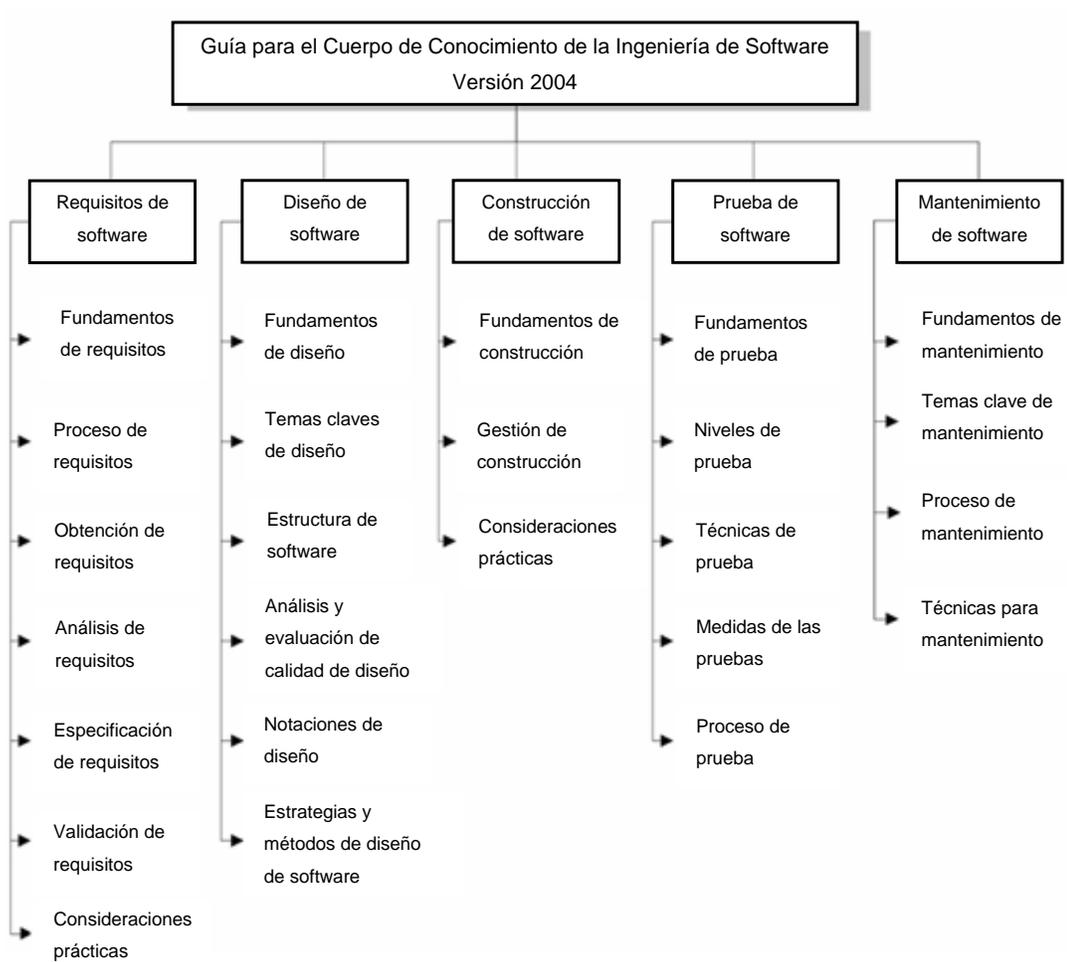


Figura 1. Primeras cinco AC del SWEBOK

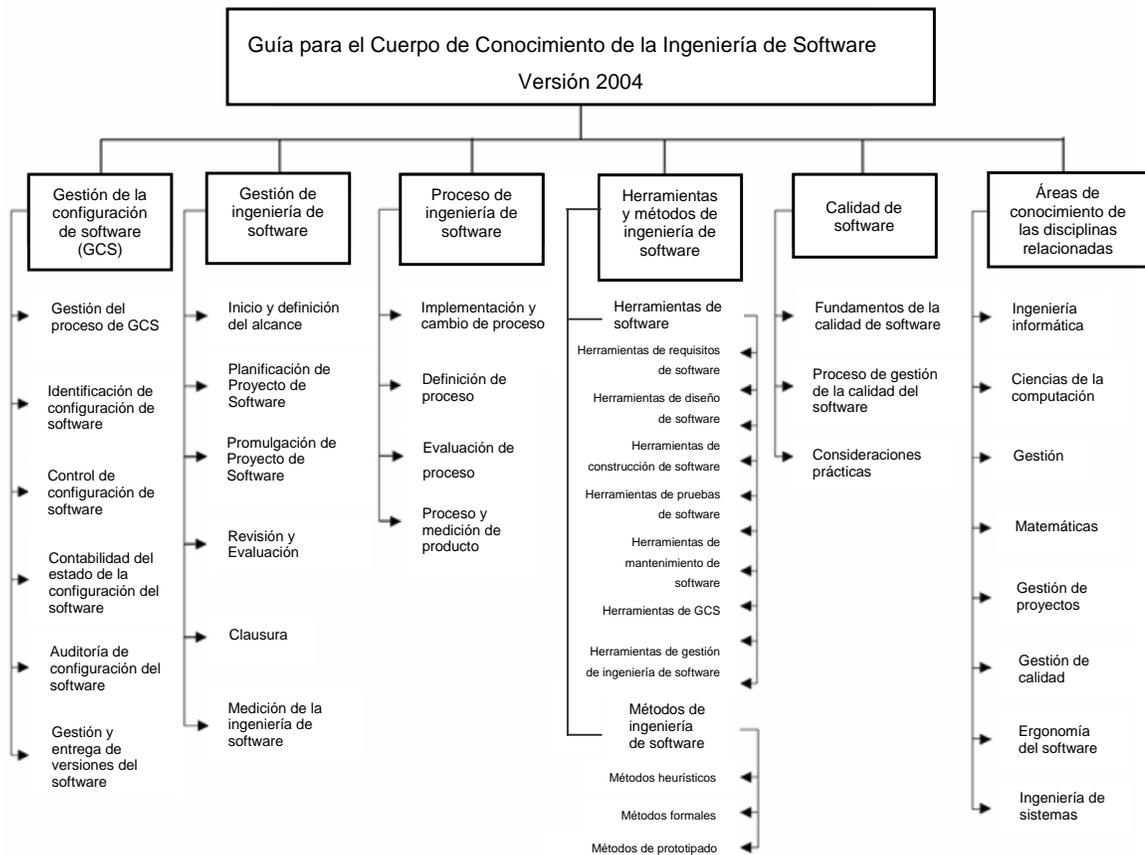


Figura 2. Últimas cinco AC del SWEBOK

Son de especial interés para el marco teórico de este trabajo el AC Proceso de Ingeniería de Software y el tópico Definición de procesos, por lo que se tratarán teóricamente en los siguientes apartados.

### Proceso de ingeniería de software

El AC Proceso de Ingeniería de Software puede examinarse en dos niveles. El primero comprende las actividades técnicas y de gestión en los procesos del CVS que se realizan durante la adquisición, desarrollo, mantenimiento y retiro del software. El segundo es el meta-nivel, que se ocupa de la

definición, implementación, evaluación, medición, gestión, cambio y mejora de los procesos del CVS. El primer nivel está cubierto por las otras AC en la Guía. Esta AC se refiere explícitamente al segundo nivel.

Como preámbulo a esta AC, se aclara que el término “proceso de ingeniería de software” puede ser interpretado de maneras diferentes; lo que podría causar confusión. Uno de los significados, donde la palabra “el” es utilizada como en “el proceso de ingeniería de software”, podría implicar que existe una única manera correcta de realizar las tareas de IS. Este significado es evitado en la guía, ya que “no existe tal proceso”. Y se añade, para aclarar, que estándares tales como el IEEE/EIA 12207 hablan de procesos de ingeniería de software, dando a entender que hay varios procesos involucrados, tales como el Proceso de Desarrollo (identificado aquí como PDS) o el Proceso de Gestión de Configuración. Un segundo significado se refiere a la discusión general de los procesos relacionados con la IS y es el que se quiere dar a entender con el título de esta AC del SWEBOK, y el más frecuentemente entendido en la descripción que se le da. Finalmente, un tercer significado podría suponer el conjunto real de actividades realizadas en una organización, que serían vistas como un proceso, especialmente desde dentro de la organización. Este significado es utilizado en esta AC en muy pocos casos.

Como AC, está relacionada con la definición, implementación, evaluación, medición, gestión, cambio, y mejora de los procesos del CVS. Está dividida en cuatro (4) subáreas, esquematizadas en la Figura 3.

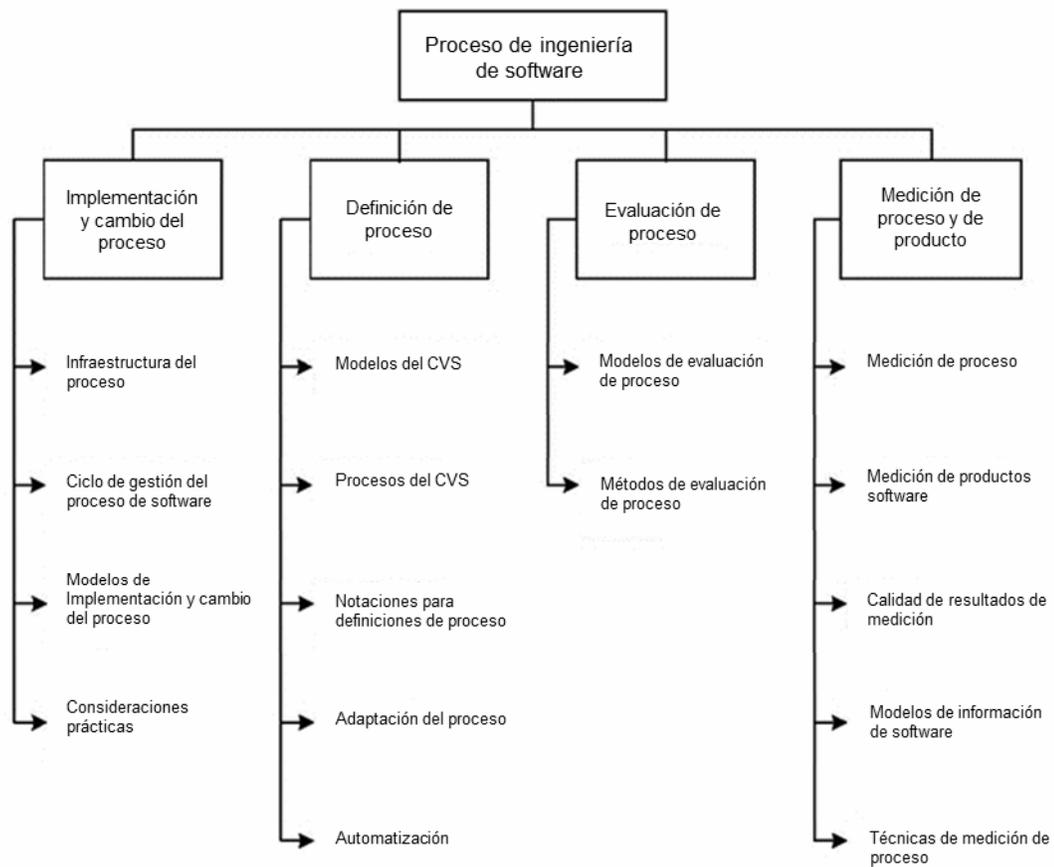


Figura 3. Desglose de temas del AC Proceso de Ingeniería de Software

La primera sub-área presenta la Implementación y el cambio del proceso. Los temas que aborda son infraestructura de proceso, el ciclo de gestión del PDS, los modelos, la implementación y cambio del proceso y consideraciones prácticas.

La segunda sub-área trata la Definición de proceso. Incluye modelos del CVS, procesos del CVS, notaciones para las definiciones de procesos, y la automatización.

La tercera sub-área es la Evaluación de proceso. Los temas que aborda

incluyen modelos de evaluación de proceso y métodos de evaluación de proceso.

La cuarta sub-área describe la Medición de proceso y producto. Los temas cubiertos son medición de proceso, medición de producto software, calidad de resultados de medición, modelos de información de software, y técnicas de medición de proceso.

A continuación, se trata la sub-área “Definición de proceso” por estar relacionada con la temática abordada en este trabajo.

#### Definición de proceso

Según el SWEBOK, una definición de proceso puede ser un procedimiento, una política o un estándar. Los procesos del CVS se definen por una serie de razones, entre ellas aumentar la calidad del producto, facilitar la comprensión y la comunicación humana, el soporte a la mejora de procesos, el soporte a la gestión de procesos, proveer guías de procesos automatizados y soporte a la ejecución automática. Los tipos de definiciones de proceso requeridos dependerán, al menos parcialmente, de la razón por la que se definen. También hay que señalar que el contexto del proyecto y la organización determinarán el tipo de definición de proceso que es más útil. Las variables importantes a tener en cuenta son la naturaleza del trabajo (por ejemplo, el mantenimiento o el desarrollo), el dominio de aplicación, el modelo de ciclo de vida, y la madurez de la organización.

Sobre este tema se tratarán sólo los tres (3) primeros tópicos por estar más estrechamente relacionados con el problema abordado en este trabajo.

## Modelos del ciclo de vida del software

Los modelos del CVS sirven como una definición de alto nivel de las fases que ocurren durante el desarrollo. Estos no tienen por objeto proporcionar definiciones detalladas pero sí resaltar las actividades clave y sus interdependencias. Ejemplos de modelos de CVS son el modelo de cascada, el modelo de prototipos, desarrollo evolutivo, iterativo/incremental, el modelo en espiral, el modelo de software reutilizable, y la síntesis automatizada de software.

## Procesos del ciclo de vida del software

Las definiciones de los procesos del CVS tienden a ser más detalladas que los modelos del CVS; sin embargo, no tratan de ordenar sus procesos en el tiempo. Esto significa que, en principio, los procesos del CVS se pueden organizar para adaptarse a cualquiera de los modelos del CVS. La referencia principal en esta área es el estándar IEEE/EIA 12207.0: Tecnologías de la Información - Procesos del Ciclo de Vida del Software<sup>1</sup>.

El estándar IEEE 1074:1997<sup>2</sup> también proporciona una lista de procesos y actividades para el desarrollo y mantenimiento de software, así como una lista de actividades del ciclo de vida que pueden ser asignadas a los procesos y organizadas de la misma forma que cualquiera de los modelos del CVS. Además, se identifican y relacionan otros estándares de software IEEE para estas actividades. En principio, el IEEE Std 1074 puede ser

---

<sup>1</sup> IEEE/EIA 12207.0: *Standard for Information Technology - Software Life Cycle Processes*

<sup>2</sup> IEEE 1074:1997 *Standard for Developing Software Life Cycle Processes*

utilizado para construir los procesos conforme a cualquiera de los modelos de ciclo de vida. Los estándares que se centran en los procesos de mantenimiento son IEEE Std 1219-1998 e ISO 14764: 1998.

Otros estándares importantes que proporcionan definiciones de procesos son:

- IEEE Std 1540: *Software Risk Management*.
- IEEE Std 1517: *Software Reuse Processes*.
- ISO/IEC 15939: *Software Measurement Process*. El AC Gestión de Ingeniería de Software proporciona una descripción detallada de este proceso.

En algunas situaciones, los procesos de ingeniería de software deben definirse teniendo en cuenta los procesos organizacionales para la gestión de la calidad. ISO 9001, establece requisitos para los procesos de gestión de la calidad e ISO/IEC 90003 interpreta los requisitos para las organizaciones de desarrollo de software.

### **Proceso de desarrollo de software**

La primera contribución importante en el área de PDS ha sido la conciencia creciente de que el desarrollo de software es un proceso complejo; los investigadores y profesionales se han dado cuenta de que el desarrollo de software es un esfuerzo colectivo, complejo y creativo. Como tal, la calidad de un producto software depende en gran medida de la gente, la organización y los procedimientos utilizados para crearlo y entregarlo. La IS tiene el objetivo de desarrollar un producto nuevo o mejorar uno existente,

por lo que es necesario un proceso que sirva como guía a todos los participantes (clientes, usuarios, desarrolladores y directores ejecutivos) conforme los productos software se hacen más complejos.

Esta visión tiene sus orígenes en los trabajos desarrollados entre los años 60 y 70. En esas dos décadas, investigadores y profesionales enfocaron sus actividades en tres tópicos principales:

- Desarrollo de lenguajes de programación estructurados (p. e. Algol, Pascal y C).
- Desarrollo de métodos y principios de diseño (p. e., ocultamiento de información, refinamiento *top-down*, descomposición funcional).
- Definición de algunos modelos de CVS (p. e. Cascada, Desarrollo incremental y Prototipado).

El tercer tópico (CVS) está directamente relacionado con la noción de PDS. Un CVS define las diferentes etapas en la vida de un producto de software – típicamente análisis y especificación de requisitos, diseño, desarrollo, verificación y validación, despliegue, operación, mantenimiento y retirada. Además, un CVS define los principios y directrices de acuerdo a lo que esas diferentes etapas han llevado a cabo. Por ejemplo, el modelo en cascada sugiere que una fase específica debe ser iniciada sólo cuando los productos de trabajo de la fase previa han sido completados. Por el contrario, el modelo en espiral considera el desarrollo de software como la repetición sistemática de un número de actividades dirigidas por el análisis del riesgo.

En general, el CVS define el “esqueleto” y la “filosofía” según la cual el PDS tiene que llevarse a cabo. Sin embargo, no prescribe un curso de acciones

precisas, una organización, herramientas y procedimientos de operación, políticas de desarrollo y restricciones. De esta manera, un CVS es claramente un punto de inicio importante para definir cómo el software debería ser desarrollado. Sin embargo, adoptar un CVS específico no es suficiente para dirigir y controlar un proyecto de software en la práctica.

La noción de PDS se construye sobre la noción de CVS y proporciona un concepto general para definir y organizar los diferentes factores y cuestiones relacionadas con las actividades del desarrollo de software. Un PDS puede ser definido como “un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software” (Fuggetta, 2000). En otras palabras, es un conjunto de actividades que proporcionan las normas que conducen al desarrollo eficiente de software de calidad y presenta las mejores prácticas que el estado actual de la tecnología permite, reduciendo así el riesgo y haciendo al proyecto más predecible; define quién está haciendo qué, cuándo, y cómo alcanzar un objetivo determinado (Jacobson y cols., 2000).

El PDS es la base para la IS. Define el marco de trabajo que debe establecerse para la entrega efectiva de la tecnología, forma la base para el control de la gestión de los proyectos de software y establece el contexto en el cual se aplican los métodos, se generan los productos de trabajo (modelos, documentos, datos, etc.), se establecen los fundamentos, se asegura la calidad, y el cambio se maneja de manera apropiada (Pressman, 2005).

De esta manera, un PDS explota un número de contribuciones y conceptos:

- Tecnología de desarrollo de software: el soporte tecnológico usado en el proceso. Claramente, para realizar las actividades de desarrollo de software se necesitan herramientas, infraestructuras y entornos. Se necesita la tecnología apropiada que haga posible y económicamente factible la creación de los productos de software complejos que la sociedad actual demanda.
- Métodos y técnicas de desarrollo de software: directrices de cómo usar la tecnología y realizar las actividades de desarrollo de software. El soporte metodológico es esencial para explotar la tecnología efectivamente.
- La conducta organizacional: la ciencia de las organizaciones y las personas. En general, el desarrollo de software es llevado a cabo por un equipo de personas que tienen que ser coordinadas y asesoradas dentro de una estructura organizacional efectiva.
- *Marketing* y economía. El desarrollo de software no es un esfuerzo autónomo. Como cualquier otro producto, el software debe satisfacer las necesidades de clientes reales en contextos específicos de mercado. Por lo tanto, las diferentes etapas del desarrollo de software (p. e., especificación de requisitos y el desarrollo/implementación) deben estar formadas de tal manera que tome en cuenta el contexto apropiado en el que el software va a ser vendido y utilizado.

En la definición de un PDS usualmente se especifican los actores ejecutando las actividades con el apoyo de las herramientas, sus roles y los artefactos producidos (Acuña y Ferré, 2001; Fuggetta, 2000). Aunque existen PDS diferentes (Sommerville, 2005), algunas actividades fundamentales son comunes para todos ellos:

- Especificación del software. Definir la funcionalidad del software y las

restricciones de su operación.

- Diseño e implementación del software. Producir software que cumpla su especificación.
- Validación del software. Validar el software para asegurar que hace lo que el cliente desea.
- Evolución del software. El software debe evolucionar para cubrir las necesidades cambiantes del cliente.

Estas actividades básicas se organizan de forma distinta en diferentes PDS. En el enfoque en cascada, están organizadas en secuencia mientras que en el desarrollo evolutivo se entrelazan. La manera de llevarlas a cabo dependerá del tipo de software, de los participantes del proceso y de la estructura organizacional implicada.

### **Modelos de procesos de software**

Los modelos de proceso del software (MPS) son representaciones abstractas de un PDS que los presentan desde una perspectiva particular, proporcionando sólo información parcial sobre ese proceso. Es por eso que es evidente el marco de trabajo del proceso, pero no los detalles de actividades específicas del mismo. Alternativamente, se usan los términos CVS y modelo de CVS (Lonchamp, 1993) como en el SWEBOK.

Los MPS se propusieron originalmente para ordenar el caos del desarrollo de software y actualmente existe una variedad de modelos de los que un ingeniero de software elige basado en los atributos del software que habrá de desarrollarse (Pressman, 2005). Según Sommerville (2005), la mayor parte de los MPS se basan en uno de los tres (3) modelos generales o paradigmas

de desarrollo de software:

- El enfoque en cascada. Considera las actividades fundamentales del PDS y las representa como fases separadas. Después de que cada fase queda definida “se firma” y el desarrollo continúa con la siguiente fase.
- Desarrollo iterativo. Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones muy abstractas. Éste se refina basándose en las peticiones del cliente para producir un sistema que satisfaga sus necesidades. El sistema puede entonces ser entregado. De forma alternativa, se puede reimplementar utilizando un enfoque más estructurado para producir un sistema más sólido y mantenible.
- Ingeniería del software basada en componentes. Este enfoque supone que las partes del sistema (un número significativo de componentes reutilizables) existen. El proceso de desarrollo del sistema se enfoca en la integración de estos componentes más que desarrollarlos desde el principio.

Estos tres (3) modelos genéricos se utilizan ampliamente en la práctica actual de la IS. No se excluyen mutuamente y a menudo se utilizan juntos, especialmente para el desarrollo de sistemas de software grandes.

### **Especificación y/o modelado de procesos de software**

El modelado de procesos de software se refiere a la definición de los procesos como modelos, además de cualquier soporte automatizado opcional disponible para el modelado y la ejecución de los modelos durante

el proceso de software. Finkelstein y cols. (citado por Acuña y Ferré, 2001) definen un modelo de proceso como la descripción de un proceso expresado en un lenguaje de modelado de procesos (PML) adecuado.

Curtis y cols. (1992) presentan algunos de los objetivos específicos y beneficios del modelado del proceso de software:

- Facilidad de comprensión y comunicación: requiere un modelo de proceso que contiene la información suficiente para su representación. Se formaliza el proceso, proporcionando así una base para el aprendizaje (capacitación).
- Soporte para la gestión y control del proceso: requiere un proyecto de proceso de software específico y el control, la gestión y coordinación.
- Orientaciones para el funcionamiento automatizado del proceso: requieren un entorno de desarrollo de software eficaz, proporcionando orientaciones al usuario, instrucciones y material de referencia.
- Provisión para soportar la ejecución automática: requieren piezas de proceso automatizadas, soporta al trabajo cooperativo, una recopilación de métricas y el aseguramiento de la integridad del proceso
- Soporte a la mejora del proceso: necesita la reutilización de procesos de software bien definidos y eficaces, la comparación de procesos alternativos y apoyar el desarrollo del proceso.

Hay diferentes enfoques de modelado de procesos; los procesos pueden ser modelados en diferentes niveles de abstracción (p. e., los modelos genéricos frente a los modelos a la medida), y también se pueden modelar con objetivos diferentes (p. e., modelos descriptivos frente a modelos prescriptivos). El tipo de información presente en un modelo de proceso

puede ser estructurada desde diferentes puntos de vista. Curtis y cols. (1992) presentan la siguiente lista que se encuentra comúnmente en la literatura:

- Funcional: representa cuáles elementos de proceso que se están aplicando y cuáles flujos de información de la entidad son importantes para dichos elementos de proceso.
- Comportamiento: representa cuándo (es decir, la secuencialidad) y bajo qué condiciones se aplican los elementos de proceso.
- Organizacional: representa dónde y por quién están siendo aplicados los elementos de proceso.
- Informativo: representa las entidades de información resultante o manipulada por un proceso, incluyendo su estructura y relaciones.

### **Lenguajes de modelado de procesos**

El énfasis puesto en la noción de PDS ha motivado una serie de iniciativas de investigación (Fuggetta, 2000). Un área de investigación primordial está relacionada con las técnicas y métodos para modelar PDS y soportar su ejecución (o publicación), por ello los investigadores han creado lenguajes y formalismos de modelado (PML) que permiten representar de forma precisa y completa las características y facetas del PDS:

- Actividades que se han de realizar para alcanzar los objetivos del proceso (p. e., desarrollar y probar un módulo).
- Roles de la gente en el proceso (p. e., el analista y el jefe de proyecto).
- Estructura y naturaleza de los artefactos [productos de trabajo] que se crearán y mantendrán (p. e., el documento de especificación de requisitos, módulos, casos de prueba).

- Herramientas que se utilizarán (p. e., herramientas CASE y compiladores).

Un PML es un formalismo capaz de representar modelos de PDS. Añade una sintaxis precisa y una semántica detallada a nivel conceptual del meta-modelo de proceso (un *framework* para desarrollar y componer MPS). Se pueden caracterizar según: sus usos previstos o su formalidad (desde notaciones gráficas hasta formalismos matemáticos) (Lonchamp, 1993).

Los PML pueden ser usados para diferentes propósitos:

- Comprensión del proceso. Un PML puede ser usado para representar en una forma precisa cómo un proceso está estructurado y organizado [5]. Esto puede ser un instrumento para eliminar incoherencias en la especificación del proceso.
- Diseño del proceso. Prácticamente, un PML puede ser utilizado para diseñar un nuevo proceso, describiendo su estructura y organización.
- La formación y la educación. Una descripción precisa del proceso puede ser útil para instruir al personal recién contratado, sobre los procedimientos y operaciones de una empresa y a los profesionales y estudiantes de la academia.
- Simulación y optimización del Proceso. Una descripción de proceso puede ser simulada para evaluar los posibles problemas, obstáculos y oportunidades de mejora.
- Soporte al proceso. Una descripción precisa del proceso puede ser interpretada y utilizada para proporcionar diferentes niveles de apoyo a las personas que trabajan en el proceso.

En general, los PML existentes se basan en una serie de paradigmas

lingüísticos que se extienden para aumentar su poder expresivo. En el Capítulo 9 del SWEBOK: Proceso de Ingeniería de Software, se resaltan una serie de notaciones que se utilizan para definir procesos; entre ellas: diagramas de flujo de datos, en términos del propósito y los resultados del proceso, como una lista de procesos descompuestos en actividades y tareas definidas en lenguaje natural, diagramas de estado, ETVX, modelado de Dependencia-Actor, notación SADT, redes de Petri, IDEFO, y basadas en reglas. Una diferencia clave entre ellas está en el tipo de información que definen, capturan y usan. Más recientemente, el OMG publicó SPEM como un estándar de modelado de procesos que pretende armonizar las notaciones de modelado existentes. En el siguiente apartado se trata de manera detallada el SPEM por ser el objeto de estudio de este trabajo de grado.

### **Meta-modelo de Ingeniería de Procesos de Software**

El SPEM (*Software and Systems Process Engineering Meta-model*) es un meta-modelo para la ingeniería de procesos así como un marco de trabajo conceptual que provee los elementos necesarios para modelar, documentar, presentar, gestionar, intercambiar, desarrollar y publicar métodos y procesos de desarrollo de software (OMG-SPEM, 2008).

SPEM puede ser una importante ayuda para que las empresas que llevan a cabo proyectos de software puedan enfrentar mejor los problemas relacionados con los procesos, entre los que cabe destacar los siguientes:

- Miembros de los equipos de desarrollo de software no tienen acceso fácil y centralizado a la información de procesos que necesitan, es decir, los distintos desarrolladores pueden contar con fuentes y

versiones diferentes de la misma información.

- Es difícil combinar e integrar contenidos y procesos que están disponibles en su propio formato propietario; cada libro, manual, herramienta utiliza un lenguaje y estilo diferente.
- Es difícil definir una aproximación de desarrollo organizado y sistemático que se adapte a las necesidades, es decir, dirija la cultura, prácticas normalizadas y los requisitos de cumplimiento.

La Figura 4 muestra un resumen del marco de trabajo general del SPEM, es decir, de los escenarios más habituales de su uso.

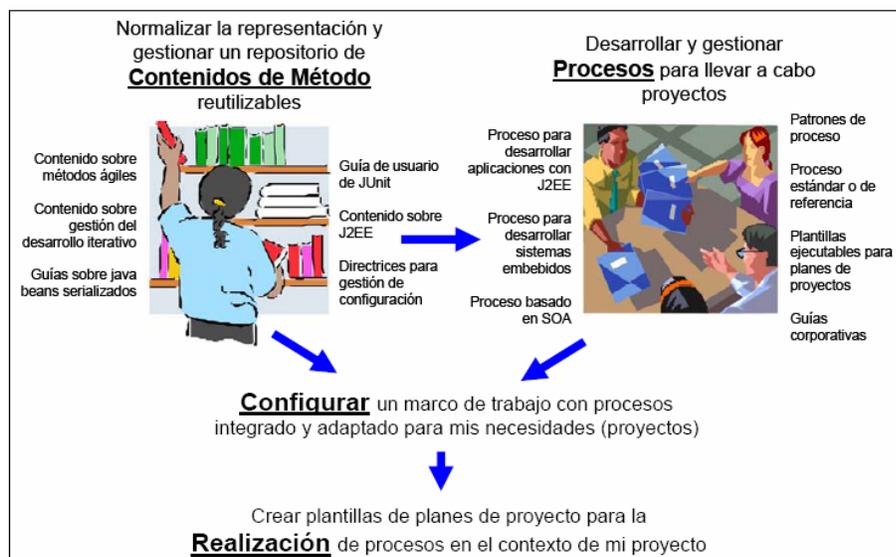


Figura 4. Marco conceptual de uso de SPEM.

Al implementar SPEM existen cuatro escenarios fundamentales:

**Proporcionar una representación normalizada y bibliotecas de contenidos de método reutilizables.** Esto en sí solo es de valor para cualquier organización de desarrollo de software, ya que supone disponer de

un repositorio de conocimiento sobre procesos en un formato estandarizado. Además, es de gran ayuda a los miembros del equipo de desarrolladores de software porque en su trabajo necesitan conocer cómo realizar las tareas de desarrollo y mantenimiento de software tales como la forma de obtener y gestionar los requisitos, la forma de hacer el análisis y el diseño, la forma de gestionar el cambio y el alcance del proyecto, y demás actividades. Así mismo, necesitan comprender los productos de trabajo que se deben crear en cada tarea, cuáles son las habilidades requeridas en cada rol, y disponer de las guías, directrices, plantillas, etc., adecuadas en cada momento. Esta base de conocimientos puede ser utilizada para la referencia y la capacitación y constituye la base para el desarrollo de procesos.

**Dar soporte al desarrollo sistemático, gestión y crecimiento de procesos de desarrollo.** Esto implica combinar, reutilizar y extender los elementos de método anteriores para configurar los procesos que sirven para guiar los proyectos. Partiendo de fragmentos de proceso elementales se puede llegar a generar todo un proceso completo o toda una metodología, incluyendo varios procesos. SPEM ayuda a los equipos de desarrollo a definir o seleccionar un proceso, incluyendo opciones para que los mismos métodos puedan ser aplicados de forma diferente en momentos distintos o en proyectos distintos, para comprender claramente cómo se relacionan unas tareas con otras, o para que los procesos puedan ser vistos como flujos de trabajo o como estructuras de desglose de trabajo (WBS, *Work Breakdown Structure*), según interese en cada momento. Para lo anterior, SPEM soporta la creación sistemática de procesos basada en la reutilización de contenidos de método. También provee el fundamento conceptual para que los ingenieros de procesos y gestores de proyectos seleccionen, adapten y rápidamente ensamblen procesos para sus proyectos concretos. El ensamblado rápido de procesos es posible gracias a la implementación de

catálogos de procesos predefinidos, “trozos” de proceso o patrones de proceso.

**Establecer un marco de trabajo general de la organización a partir de los procesos y los elementos definidos anteriormente.** Para ello, SPEM permite dar soporte al despliegue del contenido de método y proceso que se necesita en cada caso, teniendo en cuenta que ningún proyecto es exactamente como el anterior y nunca exactamente el mismo proceso software se ejecuta dos veces. En este punto es importante recordar que el nivel 3 de CMMI (proceso definido) necesita disponer de procesos estándares en la organización y de mecanismos de adaptación (*tailoring*) y SPEM provee ambas cosas. Entre otras capacidades adicionales, incorpora conceptos para:

- Reutilización de procesos o patrones de procesos.
- Variabilidad (procesos que incluyen partes alternativas configurables).
- Adaptación (los usuarios definen sus propias extensiones, omisiones y puntos de variabilidad sobre procesos estándares reutilizados).

**Generar plantillas para planes de proyecto concretos.** Esto supone que a partir de ahora los jefes de proyecto pueden contar con mucha más información y disponible de manera automática a la hora de definir los planes de los proyectos. Para darles auténtico valor, las definiciones de los procesos deben ser desplegadas en formatos que permitan su realización automática (sistemas de gestión de proyectos y recursos, motores de flujos de trabajo, entre otros). Para ello, SPEM incluye estructuras de definición de procesos que permiten expresar cómo un proceso será realizado de forma automática con estos sistemas. Ejemplos de ello son las iteraciones (una o varias definiciones de trabajo serán repetidas varias veces en un proyecto) y las

ocurrencias múltiples (varias instancias de una definición de trabajo pueden llevarse a cabo a la vez de forma paralela).

### Características del SPEM

El ámbito de aplicación del SPEM está limitado a los elementos mínimos necesarios para definir procesos, sin añadir características específicas para determinadas disciplinas o ámbitos de desarrollo (por ejemplo, la gestión de proyectos). El objetivo es dar cabida a una amplia gama de métodos y procesos de diferentes estilos, culturas, niveles de formalismo, o modelos de CVS. No es un lenguaje para el modelado de procesos en general, ya que está orientado a los PDS y tampoco provee conceptos propios para modelado del comportamiento, pero incluye mecanismos para encajar el método externo elegido para tal fin (por ejemplo, diagramas de actividad de UML).

La idea central de SPEM para representar procesos está basada en tres elementos básicos: rol, producto de trabajo y tarea (ver Figura 3). Las tareas representan el esfuerzo a realizar, los roles representan quién lo hace y los productos de trabajo representan las entradas que se utilizan en las tareas y las salidas que se producen. La idea central subyacente es que un MPS consiste, básicamente, en decir quién (rol) realiza qué (tarea) para, a partir de unas entradas (productos de trabajo), obtener unas salidas (productos de trabajo).

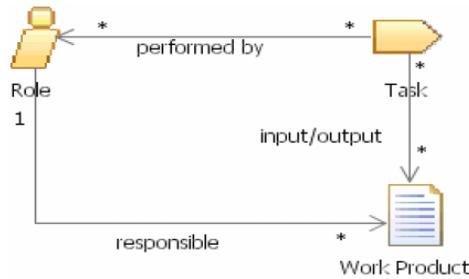


Figura 5. Idea básica de proceso en SPEM.

SPEM está basado en MOF (*Meta Object Facility*), otro estándar previo del OMG que plantea una arquitectura de modelado de cuatro niveles conceptuales (ver Tabla 1) para la definición de sus estándares, en donde cada capa se define como instancia de la anterior. Esta arquitectura de modelos representa el nivel meta más general (M3) y tiene el objetivo de permitir la incorporación de nuevos lenguajes de modelado (meta-modelos) para propósitos específicos (OMG-MOF, 2006). Un meta-modelo describe un conjunto de conceptos genéricos y sus interrelaciones, que sirven de base para la definición de modelos de un cierto dominio. Por tanto, un meta-modelo es un modelo de modelos, y mediante su uso se pueden representar modelos del correspondiente dominio.

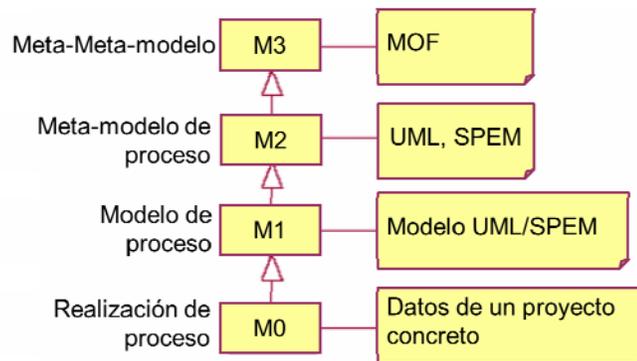


Figura 6. Niveles de modelado MOF.

Entre los distintos niveles consecutivos existen relaciones de instanciación, de forma que un elemento de un nivel  $n-1$  es una instancia de un elemento (más general, más abstracto) del nivel superior  $n$ .

A continuación se definen las diferentes capas especificadas en la arquitectura que sustenta MOF.

- Capa M3 (Meta-meta-modelo). Corresponde a MOF, es una especificación que define un lenguaje abstracto para especificar, construir y manejar elementos comunes a cualquier meta-modelo.
- Capa M2 (Meta-modelos). Especifica las entidades de un lenguaje de modelado. Los lenguajes que se han definido como instancias de MOF son entre otros: UML, SPEM y MOF en sí mismo. Adicionalmente se definen meta-modelos para otros propósitos como objetos de negocio, *workflows* y modelos de componentes (OMG-MOF, 2006).
- Capa M1 (Modelos). Se refiere a los modelos de usuarios, que suelen desarrollarse en el momento de construir un sistema de información.
- Capa M0 (Instancias). Describe instancias de las entidades propuestas en un modelo de un sistema de información. Es en este nivel en donde pueden usarse los diagramas de objetos como instancias de las clases para verificar que se cumplen las restricciones definidas en el nivel de los modelos (M1).

Aplicando estas ideas al ámbito de los PDS, los modelos se construyen mediante instanciación de los conceptos del meta-modelo de procesos genérico, es decir, de SPEM. Esta instanciación es determinada por las características propias del modelo que se quiere elaborar. En el diseño de un modelo de proceso se deben respetar las relaciones entre los diferentes

conceptos definidos en SPEM. Así, SPEM es a los PDS lo mismo que UML es a los sistemas software. UML es un meta-modelo que sirve para representar modelos de sistemas software y SPEM es un meta-modelo que sirve para representar modelos de PDS.

Recientemente se publicó una versión casi totalmente nueva y tras el desarrollo de la nueva especificación, además de conseguir lo propuesto en (OMG-SPEM, 2008), se añadieron nuevas capacidades, tales como:

- Clara separación de la definición de los contenidos de métodos de su aplicación al desarrollo de un proceso concreto.
- Mantenimiento consistente de distintas alternativas de procesos.
- Soporte para diferentes modelos de ciclo de vida.
- Mecanismo flexible para dar soporte a la variabilidad y extensibilidad de los procesos.
- Ensamblado rápido de procesos mediante el uso de patrones.

**Separación entre Contenidos de métodos y Procesos.** Tal como se deduce del marco de trabajo (mostrado en la Figura 4), en SPEM se distinguen dos grupos de conceptos separados a la hora de implementar una metodología y/o un PDS: el Contenido de método (*Method Content*) y el Proceso (*Process*). Casi todos los otros conceptos se clasifican a lo largo de esta separación, como se muestra en la Figura 7.

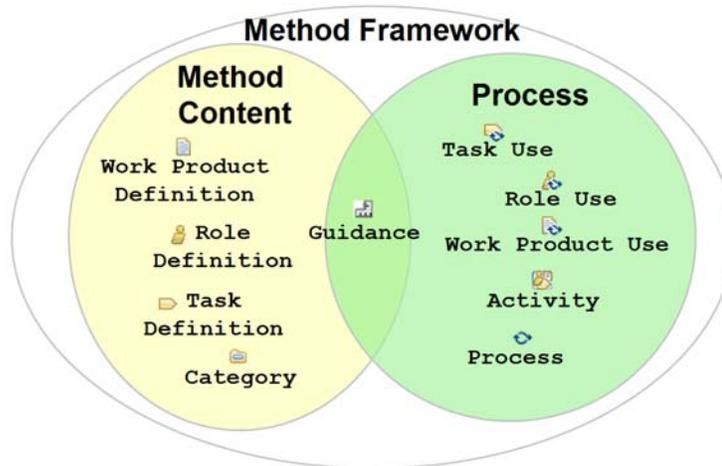


Figura 7. Aspectos principales para especificar PDS con SPEM.

El Contenido de método describe lo que será producido, los conocimientos necesarios para ello, y la explicación paso a paso que describe cómo se logran los objetivos de desarrollo específicos. Estas descripciones de Contenido de método son independientes de un ciclo de vida de desarrollo. Los Procesos describen el ciclo de vida de desarrollo tomando los elementos del Contenido de método y relacionándolos en secuencias semi-ordenadas que se adaptan a tipos específicos de proyectos.

Esta separación de Contenido de método y Proceso no es una idea nueva. Por ejemplo, ha estado presente en el UP (Jacobson y cols., 2000) desde su creación; refiriéndosele como la estructura estática frente a la estructura dinámica, y antes había sido descrita para OOSE (*Object-Oriented Software Engineering*).

La Figura 8 muestra una representación bidimensional de cómo esta separación se representa en el RUP, así como en el UP. El contenido de método, describe cómo el desarrollo del trabajo se está realizando, se categoriza por disciplinas a lo largo del eje "y". Este trabajo luego se

referencia y es secuenciado en un proceso a lo largo del eje “x” representando la línea de tiempo. Este es el ciclo de vida de un proyecto de desarrollo, que expresa el trabajo específico que realizará. Los gráficos en el centro representan un volumen (o carga) de trabajo estimado para cada disciplina. Como puede verse, por ejemplo, “nunca” se deja de trabajar sobre los requisitos en el RUP, pero sin duda hay “momentos pico” en que se lleva a cabo la mayoría del trabajo de obtención y de especificación de requisitos. También hay veces en las que se observa una tendencia a la baja donde deben ser procesados menos cambios en los requisitos para llevar el proyecto a su fin. Esto evita la “invasión de características”, en la que los requisitos de trabajo se mantienen constantes o incluso aumentan. Por lo tanto, un proceso expresa un ciclo de vida tal que las varianzas de trabajo que se realizan en las diversas disciplinas, y el trabajo en sí, están explícitamente definidos y descritos por el contenido del método.

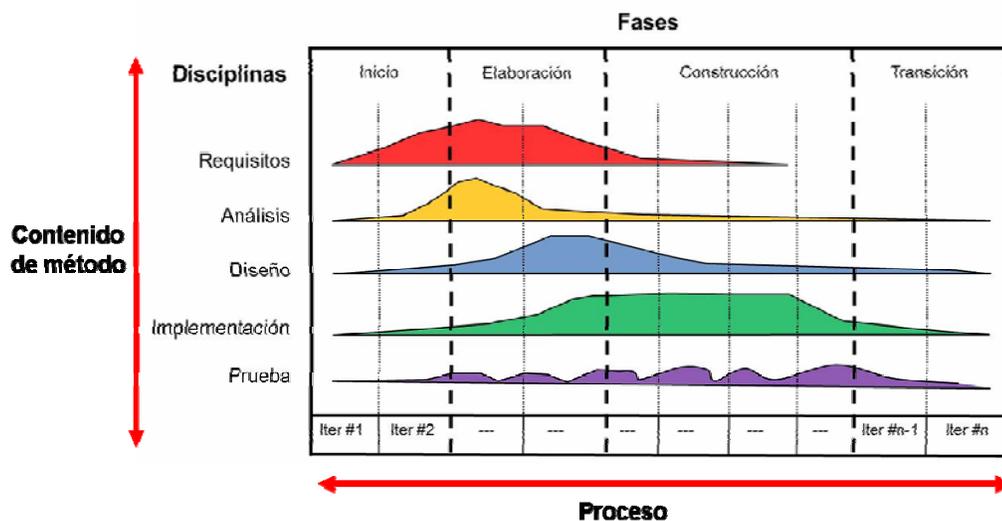


Figura 8. Separación del Contenido de método y Proceso

En SPEM, primero, se puebla el Contenido de método con los Elementos de contenido (*Content Elements*), es decir, los elementos primarios o

constructores básicos y después, se combinan y reutilizan dichos elementos para obtener Procesos. La jerarquía de desglose del trabajo, es decir, los conceptos existentes para representar el esfuerzo a realizar a distintos niveles de detalle, son los siguientes del más general al más particular (ver Figura 9):

- Proceso de despliegue (*Delivery Process*): representa un proceso tan complejo como se necesite, que será el que sirva de base para realizar cierto tipo de proyectos.
- Patrón de capacidad (*Capability Pattern*): representa un patrón de proceso, es decir, un fragmento de proceso que puede ser reutilizado más de una vez en un *Delivery Process*.
- Actividad (*Activity*): es el elemento central para definir procesos ya que permite organizar los elementos básicos de proceso (roles, productos de trabajo y tareas).
- Tarea (*Task*): es la porción más pequeña de trabajo en un modelo de proceso en SPEM.

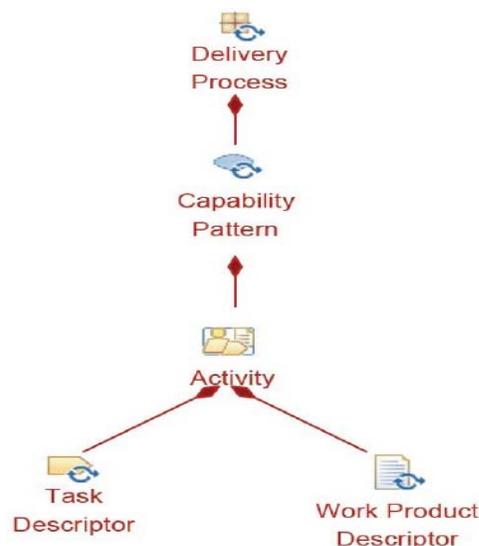


Figura 9. Conceptos para representar la jerarquía de desglose de trabajo.

**Mantenimiento consistente de muchos procesos alternativos.** Para ello, SPEM incluye: i) un conjunto extendido de interrelaciones de reutilización y variabilidad con semántica de herencia y orientación a aspectos; ii) conceptos de patrones de proceso, y iii) *plug-ins* de métodos. Estas opciones permiten tener diferentes variantes de procesos específicos, basados en los mismos contenidos de método y estructuras de procesos, pero aplicados con diferente detalle y escala.

**Muchos ciclos de vida diferentes.** SPEM permite trabajar con distintos tipos de CVS: Cascada, Iterativo, Incremental, Evolutivo, etc. Para ello incluye un conjunto de atributos que permiten especificar aspectos temporales para los elementos de proceso que luego pueden ser asociados a los planes de proyectos. Un ejemplo de atributo para clases de ciclos de vida es la propiedad Iteración, que permite representar que la ejecución de una o varias descripciones de trabajo se pueden repetir más de una vez.

**Variabilidad y extensibilidad.** Para lo cual SPEM incluye un mecanismo de *plug-ins* de dos tipos (ver Figura 10):

- *Method Plug-ins* para particularizar y adaptar contenidos de método sin modificar el original.
- *Process Plug-ins* para procesos, pudiendo añadir o sustituir elementos de trabajo en la WBS sin afectar al original.

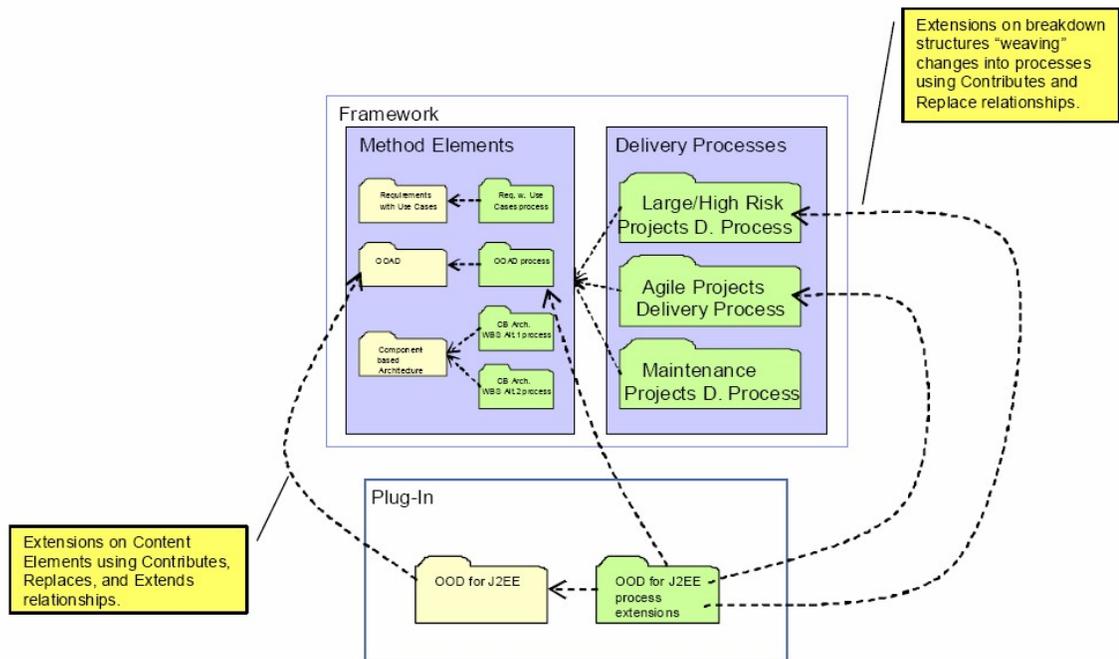


Figura 10. Ejemplo del mecanismo de *plug-ins* en SPEM.

**Patrones de proceso.** Son bloques (trozos de proceso) reutilizables para crear nuevos procesos. La selección y aplicación de un patrón de proceso puede ser hecha de dos formas: a) puede ser copiado y modificado, permitiendo individualizar el contenido del patrón según las necesidades de cada momento; o b) puede ser aplicado por medio del mecanismo de Actividad en Uso (*Use Activity*), que es una forma avanzada de reutilizar estructuras de proceso. Una Actividad en Uso define tipos de interrelaciones para que cuando el patrón esté siendo revisado o modificado, todos los cambios se reflejen automáticamente en todos los procesos en que se aplica el patrón.

**Componentes de proceso.** Son piezas de proceso sustituibles y reutilizables basadas en los principios de encapsulamiento y caja negra. No se especifica la descripción de trabajo interna del componente sino que sólo

se especifican los productos de trabajo de entrada y salida que habrá mediante los llamados puertos de productos de trabajo (ver Figura 10). Esta opción de SPEM permite manejar las situaciones en que un proyecto requiere que partes del proceso no sean decididas hasta la ejecución o nunca (caso típico: *outsourcing*).

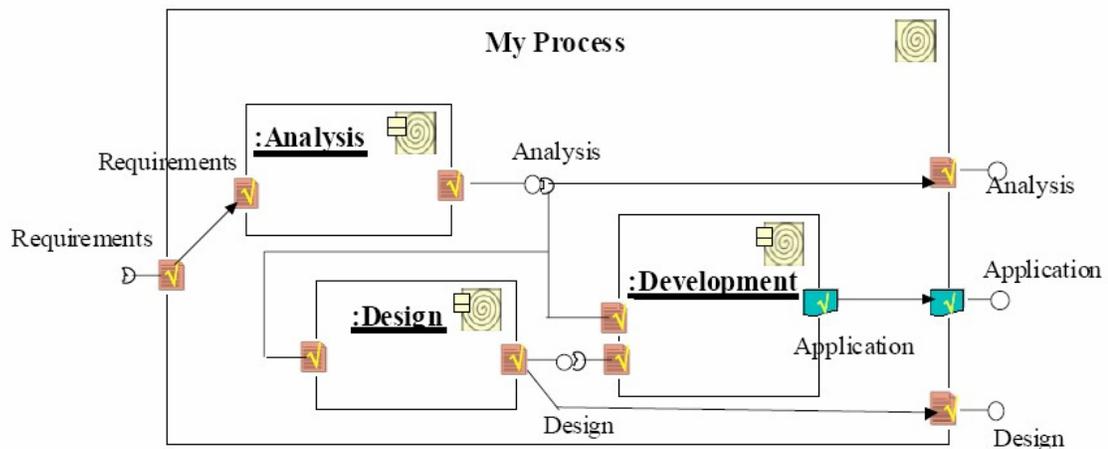


Figura 11. Ejemplo de componente de proceso en SPEM.

### Mecanismos de trabajo con SPEM

El SPEM se describe de dos maneras (ver parte izquierda de la Figura 12) en la especificación proporcionada por el OMG (OMG-SPEM, 2005):

Como un **Meta-modelo MOF-compliant**, es decir, un meta-modelo MOF del nivel M2 de forma que todos sus elementos están definidos mediante instanciación de elementos del meta-meta-modelo universal (del nivel M3). Este meta-modelo define todas las estructuras y reglas de estructuración para representar contenidos de métodos y procesos. Es completo en sí mismo, es decir, no necesita de otro meta-modelo, aunque en la práctica reutiliza algunas clases de UML por razones de ahorro.

Como **Perfil UML**, define un conjunto de estereotipos UML que permiten representar métodos y procesos usando UML.

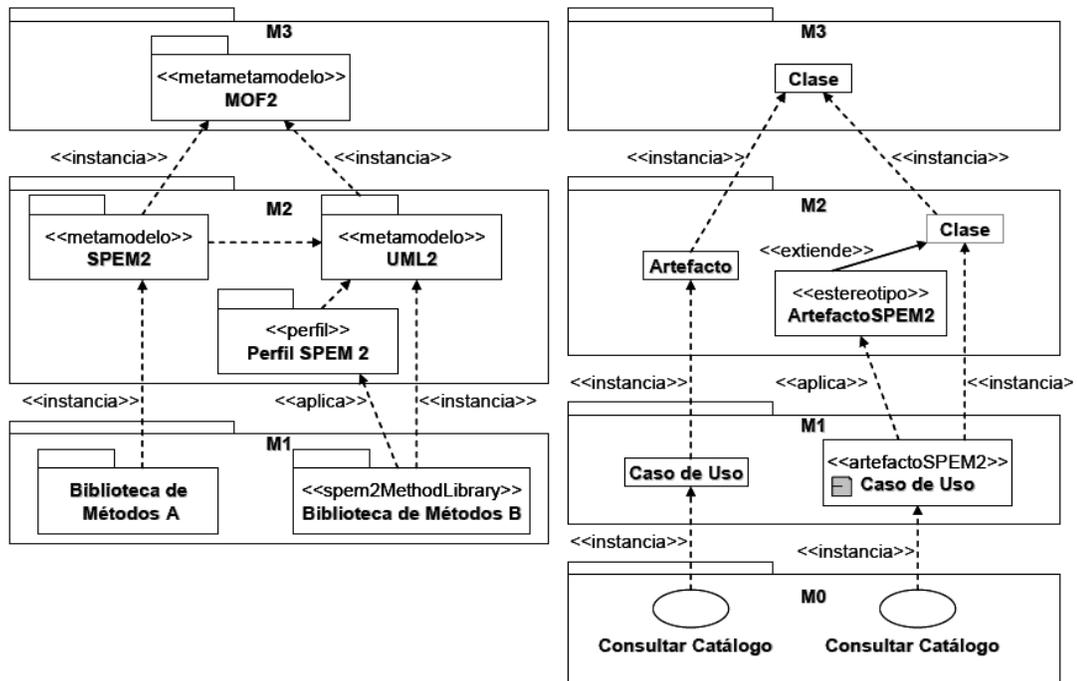


Figura 12. Niveles de modelado y ejemplos de instanciaciones.

Es posible crear bibliotecas de métodos, es decir, colecciones de elementos y fragmentos de procesos, con ambos mecanismos. Así, la “Biblioteca de Métodos A” en el nivel M1 de la Figura 12 (lado izquierdo) es un ejemplo de instancia concreta (modelo) del meta-modelo SPEM utilizándolo como un esquema para representar su contenido (primer mecanismo). Esto implica que, mientras SPEM define los conceptos de rol, tarea y artefacto y las relaciones entre ellos, en la “Biblioteca de Métodos A” se incluyen instancias concretas de dichos conceptos, tales como “Analista de Sistema” (instancia de Rol) y “Caso de Uso” (instancia de Artefacto). En el lado derecho de la Figura 5 puede comprobarse que “Caso de Uso” (nivel M1) es una instancia

directa de la meta-clase “Artefacto” (nivel M2) de SPEM, que a su vez es instancia de la meta-meta-clase “Clase” del nivel M3. Por último, una instancia de “Caso de Uso” que se podría crear durante un proyecto concreto de desarrollo de un sistema software sería “Consultar Catálogo” (nivel M0).

Los estereotipos incluidos en el perfil UML de SPEM incluyen íconos para la representación visual. Esto permite utilizar diagramas UML con dichos estereotipos para representar los métodos y procesos. La Figura 13 muestra el contexto de la tarea “*Use Case Analysis*”, indicando los roles que intervienen (*System Analyst* y *Designer*) y los productos de entrada (*Analysis Model* y *Use Case*) y de salida (*Analysis Model* y *Use Case Realization*).

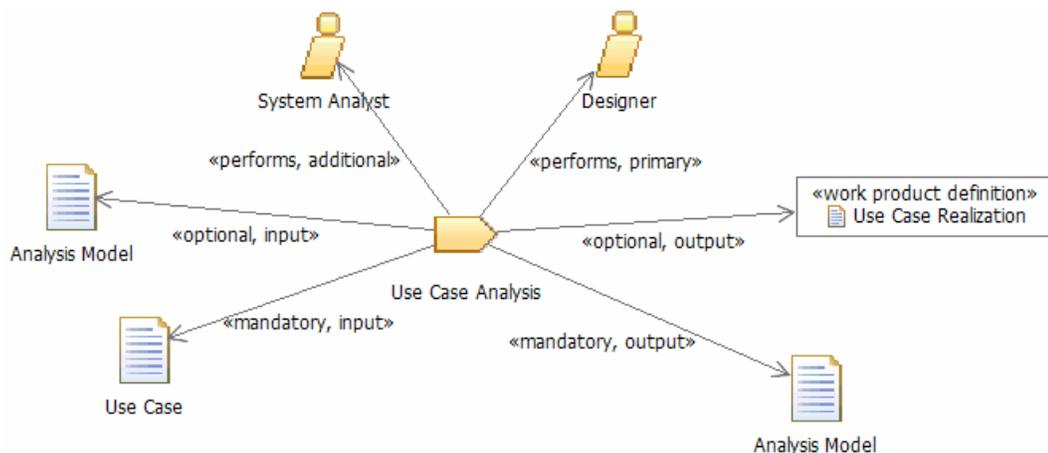


Figura 13. Representación gráfica de procesos con el perfil UML de SPEM.

### Meta-modelo SPEM

De aquí en adelante se hace referente teórico únicamente al meta-modelo basado en MOF; el mecanismo de uso más potente de SPEM.

## Arquitectura general del Meta-modelo SPEM

Al igual que UML, con fines de organización y reutilización, SPEM está organizado en siete (7) paquetes, que se muestran en la Figura 14. Cada paquete es una unidad lógica que extiende los paquetes de los que depende, proveyendo estructuras y capacidades adicionales. Los paquetes definidos en un nivel inferior pueden ser realizados en una implementación parcial de SPEM sin necesidad de los paquetes de los niveles superiores. Como regla general, cada clase del meta-modelo (constructor) se incluye en el paquete del nivel más inferior posible. En algunos casos, las clases se extienden (especializan) en paquetes de nivel superior vía el mecanismo “merge” (idéntico al de UML 2) para incluir más propiedades e interrelaciones. Un ingeniero de procesos puede elegir utilizar diferentes niveles de capacidades, conjuntos de conceptos, y niveles de formalismo para expresar sus procesos utilizando unos u otros paquetes.

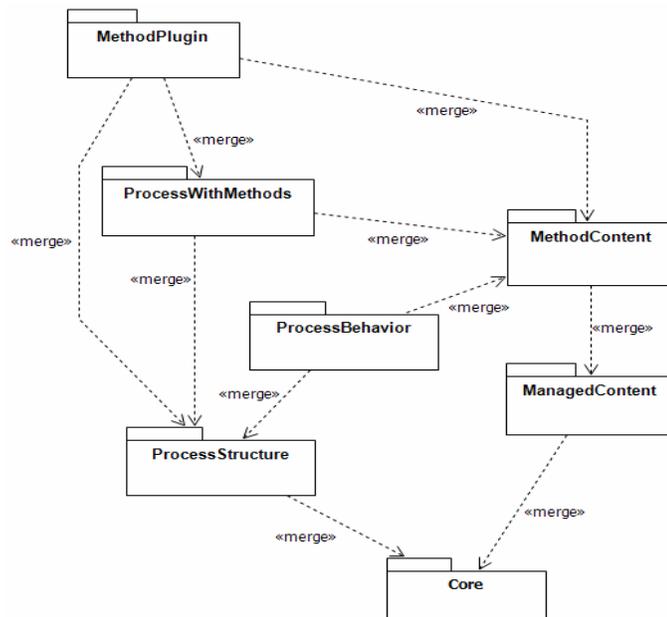


Figura 14. Estructura del Meta-modelo SPEM.

De forma resumida, los paquetes que se muestran en la figura anterior proporcionan las capacidades siguientes.

*Core.* Contiene las clases y abstracciones que sirven de base para las clases de los demás paquetes del meta-modelo. Provee dos (2) capacidades básicas: (a) crear cualificaciones definidas por el usuario (*kinds*) que permiten establecer tipos diferentes entre las instancias de una clase; y (b) una colección de clases abstractas para definir el trabajo.

*Process Structure.* Define la base para la creación de modelos de procesos flexibles y sencillos, es decir, define la WBS estática mediante anidamiento de actividades y dependencias de precedencia entre ellas. Dicha estructura también incluye referencias a la lista de Roles que realizan cada actividad y a los Productos de Trabajo que son entradas y/o salidas. También provee capacidades para reutilización mediante ensamblado de procesos usando conjuntos de actividades enlazadas de forma dinámica.

*Process Behavior.* Permite extender las estructuras del paquete anterior con modelos de comportamiento externos: diagramas de actividad de UML 2 (comportamiento de proceso), máquina de estados (ciclo de vida de un producto de trabajo), etc. En vez de incluir un mecanismo propio para representar el comportamiento, se opta por reutilizar los ya existentes (de UML o de otro tipo).

*Managed Content.* Permite incorporar y gestionar descripciones en lenguaje natural, documentos y otras informaciones útiles para la comprensión por humanos. Esta posibilidad es muy interesante y útil porque ciertos valores y culturas no pueden ser formalizados con modelos, sólo pueden ser capturados con documentación en lenguaje natural. Hay libertad total para

combinar modelos estructurales de proceso con contenidos en lenguaje natural. Así, un proceso puede estar formado sólo por una colección de guías definiendo buenas prácticas (esto es especialmente apto en el caso de métodos ágiles poco estructurados); sólo una estructura de actividades sin ningún tipo de documento textual; o una combinación interrelacionada de ambas cosas.

*Method Content.* Incluye los conceptos para crear elementos de método, que son los ítems elementales que sirven de base para el ensamblado de procesos, metodologías, ciclos de vida, etc. Los principales elementos de método se derivan del patrón básico de SPEM: alguien (rol) hace algo (tarea) para obtener algo (producto de trabajo) basándose o ayudándose en algo (guía). Por tanto, los elementos de método permiten describir, con el detalle necesario, cómo se alcanzan los objetivos del proceso haciendo qué tareas por qué roles usando qué recursos y obteniendo qué resultados. Los procesos reutilizan y relacionan entre sí los elementos de método de distintas maneras para diferentes tipos de proyectos. Los procesos *ad-hoc* tienen estructura de descomposición pero no reutilizan elementos de método, al no estar previamente definidos.

*Process with Methods.* Incorpora nuevas estructuras para poder integrar los procesos definidos con el paquete *Process Structure* con los elementos de método (instancias del paquete *Method Content*). Al asociar elementos de método a partes específicas de procesos, se crean nuevas clases (tarea en uso, rol en uso, etc.) que heredan de los elementos de método pero pueden tener cambios individualizados.

*Method Plug-in.* Incorpora conceptos para diseñar y gestionar bibliotecas o repositorios de contenidos de método y de procesos, que sean mantenibles a

gran escala, reutilizables y configurables. Con estos conceptos los ingenieros de procesos pueden definir una o varias Configuraciones de Método de cada proceso. Esto permite tener vistas diferentes de un mismo proceso adaptadas a distintas audiencias o tipos de usuarios.

Utilizando unos u otros paquetes, se puede disponer de diferentes capacidades, conjuntos de conceptos y niveles de formalismo para expresar PDS. Los escenarios más habituales son los siguientes:

- *Core + Managed Content + Method Content*. En este caso no hay modelos de proceso formales definidos, sino tan solo un repositorio para gestionar la documentación de descripciones de métodos de desarrollo y mantenimiento, técnicas y mejores prácticas.
- *Anterior + Process Structure + Process Behavior*. Ahora ya existe un proceso definido pero no son necesarios mecanismos avanzados para organizar o gestionar un repositorio de métodos, o para permitir diferentes vistas de un mismo proceso.
- Los siete (7) paquetes completos se emplean cuando se quiere disponer de toda la potencia y funcionalidad de SPEM.

### Organización de un Repositorio SPEM

Un repositorio o biblioteca de métodos y procesos (*Method Library*) en SPEM es una colección de uno o más *plug-ins* y una o varias configuraciones (*Configuration*). Cada *plug-in* se almacena en un directorio de disco diferente e incluye contenido de método (*Method Content*) y procesos (*Processes*), cada uno de los cuales se refiere a las dos partes, izquierda y derecha, mostradas en la Figura 12. A su vez, el contenido de método está formado por paquetes de contenido (*Content Package*), categorías estándar

(*Standard Category*) y categorías personalizadas (*Custom Category*); y el apartado de Procesos contiene patrones de proceso (*Capability Pattern*) y procesos para despliegue (*Delivery Process*). Un ejemplo de toda esta estructura se muestra en la Figura 15.

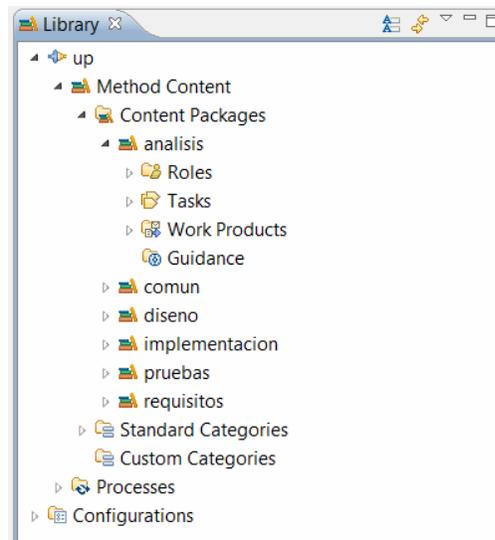


Figura 15. Ejemplo de jerarquía para organizar un repositorio SPEM

## Contenido de Método

A continuación se presentan los conceptos del Contenido de Método (*Method Content*) de SPEM. No es una revisión exhaustiva, ya que no se incluyen aquellos conceptos que quedan “ocultos” al usuario cuando modela procesos con algún editor de SPEM. Los conceptos que se presentan corresponden principalmente con los paquetes *Managed Content*, *Method Content*, *Process with Methods* (en parte) y con el *Base Plug-in*, que es un *plug-in* pre-construido incluyendo diversas especializaciones predefinidas en SPEM para actividad, categoría, guía y producto de trabajo.

El contenido de método puede ser organizado a voluntad del usuario mediante una jerarquía de paquetes de contenido (*Content Package*), cada

uno de los cuales puede incluir roles, tareas, productos de trabajo y guías.

En la Figura 16 se muestra la jerarquía completa de conceptos empleados en el *Method Content*, incluidos los elementos de contenido (*Content Element*).

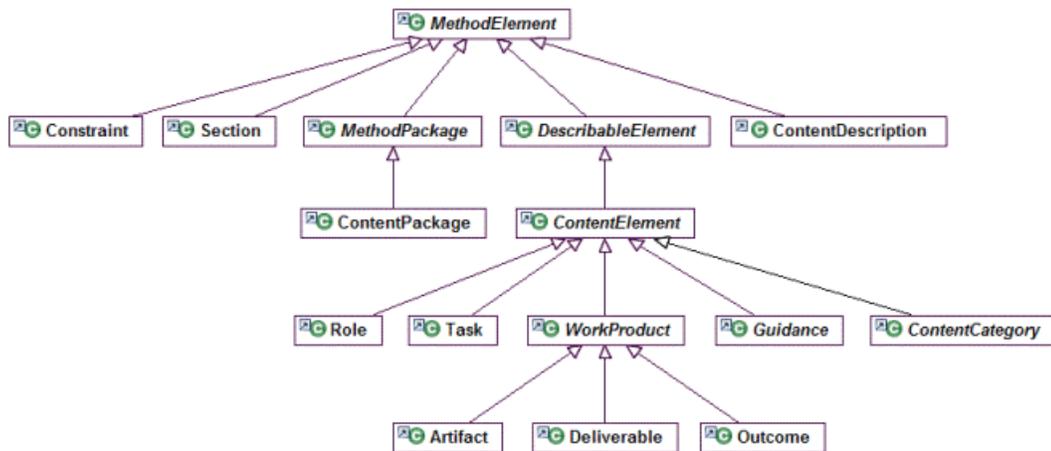


Figura 16. Jerarquía de conceptos del *Method Content*

Los elementos descriptibles (*Describable Element*) son elementos de método (o de proceso) que pueden tener descripciones textuales mediante una Descripción de Contenido (*Content Description*) opcional, que incluye nombre de presentación, descripción breve, descripción principal (que permite texto enriquecido), y propósito. Un elemento descriptible puede tener 0..\* Secciones (*Section*), que a su vez pueden estar anidadas. Además, puede estar asociado con 0..\* Guías (*Guidance*) y con 0..\* Categorías (*Category*). Éstas últimas sirven para poder clasificarlos y agruparlos de múltiples formas.

#### Elementos de Contenido (*Content Element*)

Son los constructores básicos y se derivan del patrón primitivo de trabajo:

alguien (rol) hace algo (tarea) para obtener algo (producto de trabajo) basándose o ayudándose en algo (guía). En consecuencia, los cuatro tipos de elementos de contenido son: Tarea, Rol, Producto de Trabajo y Guía. Adicionalmente, existe un quinto tipo de elemento de contenido incluido con fines de clasificación y agrupación, llamado Categorías. A continuación se presenta cada uno de ellos.

*Tareas (Task Definition):* Una Tarea describe una unidad de trabajo asignable y gestionable, es decir, es la unidad atómica de trabajo para definir procesos. Su granularidad es de unas pocas horas a unos pocos días, afectando a unos pocos productos de trabajo y vinculando a unos pocos roles. Es un Elemento de Método que define el trabajo realizado por roles, pero también es una Definición de Trabajo (en procesos). Una Tarea está asociada con (ver ejemplo en Figura 15):

- 1..\* Roles, distinguiendo entre:
  - 1 realizador principal obligatorio [responsable]
  - 0..\* realizadores adicionales opcionales
- 1..\* Productos de Trabajo como:
  - Entradas obligatorias
  - Entradas opcionales
  - Salidas
- 0..\* Herramientas, que se recomienda usar.
- 0..\* Pasos, que describen de forma secuencial el trabajo a realizar.
- 0..\* Habilidades, que se requieren habitualmente para llevar a cabo la tarea.

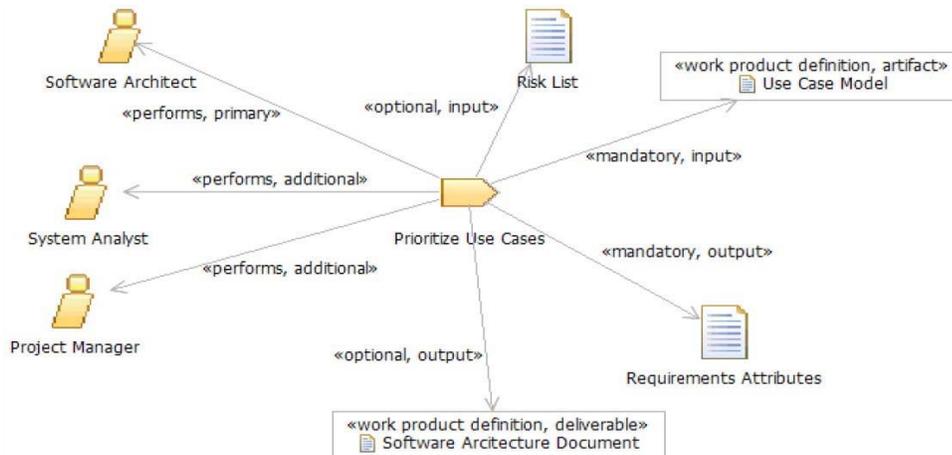


Figura 17. Ejemplo de asociaciones de una Tarea

**Rol (Role Definition):** Un Rol define un conjunto de habilidades, competencias y responsabilidades relacionadas, de un individuo o de un grupo. No se deben confundir roles con personas, ya que la vinculación entre personas y roles se realiza durante la planificación del proyecto y puede ocurrir que un individuo desempeñe varios roles o que un rol sea desempeñado por varios individuos. Un rol es un Elemento de Método usado en las Definiciones de Tareas para señalar quiénes las realizan. Un Rol está asociado con:

- 0..\* Productos de Trabajo, de los que es responsable.
- 0..\* Habilidades, que el rol típicamente provee.

**Productos de Trabajo (Work Product Definition):** Un Producto de Trabajo es consumido, producido o modificado por Tareas. Un Producto de Trabajo puede estar asociado con otros Productos de Trabajo mediante asociaciones de los siguientes tipos:

- **Composición (Composition),** cuando las instancias de un producto de trabajo sirven para componer instancias de otro producto de trabajo. Ejemplo: “Actores” se emplean para componer “Casos de uso”.

- Agregación (*Aggregation*), si un producto de trabajo está formado por agregación de otros. Ejemplo: el “Manual de usuario” incluye el “Manual de instalación”.
- Es impactado por (*Impacted by*), cuando un producto de trabajo impacta en otro, es decir, si los cambios del primero obligan a cambiar el segundo. Ejemplo: si cambia el “Modelo de casos de uso”, es necesario adaptar a dicho cambio la “Realización de casos de uso”.

Existen tres (3) tipos predefinidos (especializaciones) de Productos de Trabajo:

- Artefacto (*Artifact*), de naturaleza tangible (modelo, documento, código, archivos). Un artefacto puede estar formado por otros artefactos más simples.
- Entregable (*Deliverable*): provee una descripción y definición para empaquetar otros productos de trabajo con fines de entrega a un cliente interno o externo. Representa una salida de un proceso que tiene valor para un usuario, cliente u otro participante (*stakeholder*). Está asociado con 0.\* componentes de entregable (*Deliverable Component*), que son los productos de trabajo, habitualmente artefactos, que lo forman. La versión en castellano de *EPF Composer* le denomina “producto final”. Aquí se emplea el término “Entregable”, que es el utilizado por los diversos estándares de IS y de gestión de proyectos.
- Resultado (*Outcome*): un producto de trabajo de naturaleza intangible (resultado o estado), o que no está formalmente definido. Por ejemplo, una herramienta de software o un servidor instalados.

Guías (*Guidance*): Una guía o instrucción es un elemento de método (o de

proceso) que provee información adicional relacionada con otros elementos. Por ejemplo: ayuda o información sobre cómo trabaja un rol, cómo crear un producto de trabajo, cómo usar una herramienta o cómo realizar una tarea. SPEM tiene predefinidos bastantes tipos de guías:

- Activo Reutilizable (*Reusable Asset*): Provee una solución a un problema para un contexto dado. Incluye reglas o instrucciones sobre cómo utilizarlo.
- Concepto (*Concept*): Resumen de ideas clave asociadas con principios básicos subyacentes. Refieren a tópicos más generales que las directrices y abarcan varios productos de trabajo y/o actividades. Ejemplo: “iterativo e incremental”.
- Definición de Término (*Term Definition*): Definición de un término, concepto o idea relevante. Sirve para generar una especie de glosario automático. Se relacionan con Elementos de Contenido mediante su aparición en las descripciones textuales.
- Directriz (*Guideline*): También llamada Instrucción en *EPF Composer*. Provee detalle adicional sobre cómo realizar una tarea o grupo de tareas, o provee información adicional, reglas y recomendaciones sobre productos de trabajo. Entre otros, puede incluir detalles sobre: las mejores prácticas y aproximaciones diferentes para hacer un trabajo; cómo usar cierto tipo de producto de trabajo; los subtipos y variantes de un artefacto y su evolución a lo largo del tiempo; habilidades que los roles deben adquirir o mejorar; medidas del progreso o madurez, etc.
- Documentación (*Whitepaper*): Versión especial de Concepto que ha sido revisada o publicada externamente y que puede ser leída y comprendida de forma aislada.
- Ejemplo (*Example*): Ejemplo de una instancia típica, parcialmente

completada, de uno o más productos de trabajo o descripción del escenario en que una tarea debe ser realizada.

- Guía de Herramienta (*Tool Mentor*): Explica el uso de una cierta herramienta en el contexto de cierto trabajo, o de forma independiente.
- Guía para la Estimación (*Estimation Considerations*): Indicaciones para estimar el esfuerzo asociado con cierto trabajo, incluyendo consideraciones sobre cómo hacer la estimación y las métricas a utilizar. La versión en castellano de *EPF Composer* le llama “Consideraciones para el Cálculo”, pero es más correcto denominarle “Guía para la estimación”, ya que no es lo mismo estimar (aproximación a un valor futuro, por fuerza inexacto) que calcular (con resultado exacto salvo error).
- Hoja de Ruta (*Roadmap*): Hoja de ruta que describe, en forma de camino lineal, cómo suele llevarse a cabo una actividad o proceso complejos. Provee información sobre como las actividades y tareas se relacionan entre sí a lo largo del tiempo. Sólo pueden estar asociados a Actividades y Procesos. La versión en castellano de *EPF Composer* le denomina “mapa”. Aquí se emplea el término “hoja de ruta”, que es como se le conoce habitualmente y cómo figura en el diccionario de la RAE.
- Informe (*Report*): Plantilla predefinida de un resultado que se obtiene de forma automática mediante alguna herramienta.
- Lista de Comprobación (*Checklist*): Identifica una serie de ítems que deben ser completados o verificados.
- Material de Soporte (*Supporting Material*): Comodín para utilizar cuando se está en un caso que no encaja en ninguno de los demás tipos de guías.

- Plantilla (*Template*): Establece la tabla de contenidos, secciones, cabeceras y formato estandarizado predefinido de un artefacto (documentos, modelos). Puede incluir descripciones sobre cómo usar y completar cada parte.
- Práctica (*Practice*): Manera o estrategia predefinida de hacer un trabajo que tiene un impacto positivo sobre la calidad de un producto de trabajo o de un proceso. Son ortogonales a los métodos y procesos, de forma que una práctica resume aspectos que pueden impactar en diferentes partes de un método o proceso. Ejemplos: “gestionar riesgos”, “verificación continua de la calidad”, “desarrollo centrado en la arquitectura”, “desarrollo basado en componentes”.

Categorías (*Category*): Una Categoría es un elemento de contenido, o de proceso, usado para categorizar, es decir, clasificar o agrupar dichos elementos en base a los criterios que desee el ingeniero de procesos. Una categoría puede tener 0..\* subcategorías. Esto permite establecer cualquier tipo de jerarquía de agrupamiento de elementos. SPEM distingue dos clases de categorías:

- Estándar (*Standard Category*): Vienen predefinidas en SPEM. Se comentan después.
- Personalizada (*Custom Category*): Sirven para que el ingeniero de procesos pueda definir otras categorías nuevas.

En SPEM se incluyen cinco (5) tipos predefinidos de categorías:

- Conjunto de Roles (*Role Set*): Sirven para agrupar roles que tienen algo en común (usan técnicas similares, requieren habilidades parecidas). Ejemplo: Analista englobando a Analista de Sistemas e Ingeniero de Requisitos.

- Disciplina (*Discipline*): Permiten categorizar el trabajo (tareas). Una disciplina es una colección de tareas que están relacionadas con un área principal de esfuerzo dentro de un proyecto completo. Suelen estar basadas en una perspectiva tradicional de proyectos en cascada: requisitos, análisis, diseño, construcción, pruebas, mantenimiento, gestión del proyecto, aseguramiento de calidad, etc.
- Dominio (*Domain*): Permiten establecer una jerarquía de dominios, para clasificar productos de trabajo, con tantos niveles como se desee. El nivel inferior son productos de trabajo y el resto de niveles son dominios y subdominios. Al ser una jerarquía, un producto de trabajo sólo puede estar asociado con un único dominio. Ejemplos: “modelo”, “código”.
- Herramienta (*Tool*): A pesar del nombre, no sirve para categorizar herramientas sino guías de herramientas. Por tanto, para asociar herramientas con tareas, roles o productos de trabajo deberá emplearse una categoría personalizada.
- Clase de Producto de Trabajo (*Work Product Kind*): Se distingue de Dominio en que un producto de trabajo puede pertenecer a varias clases de producto de trabajo distintas. Ejemplo: El mismo artefacto puede incluirse dentro de “Documento de Análisis” y dentro de “Producto Software”.

### Propiedades principales

Todos los *Content Elements* de todos los tipos anteriores tienen las propiedades ya comentadas de los elementos describibles, incluyendo: *name* (nombre); *presentation name* (nombre de presentación); *brief description* (descripción breve); *main description* (descripción principal). La última

permite utilizar un editor de texto enriquecido (*rich text editor*), pudiendo incluir texto con formatos variados, imágenes, etc., al estilo del HTML.

Otras propiedades específicas de cada clase de *Content Element* son:

- *Task*: *purpose* (objetivo); *key considerations* (factores clave); *alternatives* (alternativas); *steps* (lista de pasos), que detalla el trabajo a realizar de forma ordenada.
- *Role*: *key considerations* (factores clave); *skills* (habilidades); *assignment approaches* (propuestas de asignación); *synonyms* (sinónimos).
- *Work Product*: *unique ID* (ID exclusivo); *purpose* (objetivo); *key considerations* (factores clave); *impact of not having* (impacto de no tener); *reason for not needing* (motivos para no necesitar). Además, los *Artifacts* tienen *brief outline* (esquematización breve), y *representation options* (opciones de representación). Los *Deliverables* también tienen *external description* (descripción externa), *packaging guidance* (guía de empaquetado), y *deliverable parts* (componentes de entregable), que es una lista de otros *work products* que forman parte del *deliverable*.
- *Guidance*: algunos tipos de *guidances* tienen otros atributos. Así, *Checklist* tiene una lista de *check items* (elementos de comprobación), y *Practice* tiene *additional information* (información adicional), *goals* (objetivos), *application* (aplicación), *problem* (problema), *background* (fondo), *level of adoption* (nivel de adopción), *referentes* (referencias), que es una lista de *content elements* referenciados. Por otro lado, un *Template* tiene una lista de *template files* (archivos de plantilla). Cada clase de *content element* tiene un icono predefinido por el perfil UML de SPEM, que lo caracteriza en los diagramas y documentación

utilizados o generados también por las diversas herramientas (como *EPF Composer*). Con fines de personalización, a cada tipo de *work product* y *guidance* se le puede asociar otro ícono específico.

## Asociaciones

De forma resumida, las relaciones que se pueden representar con SPEM, son las siguientes:

- *Task-Steps*: lista ordenadas de pasos que se llevan a cabo en una tarea.
- *Task-Roles*: *primary performer* (realizador principal, obligatorio); *additional performers* (realizadores adicionales).
- *Task-Work Products*: *mandatory inputs* (entradas obligatorias); *optional inputs* (entradas opcionales); *outputs* (salidas).
- *Task-Guidances*: *guidances* relacionadas (sólo tipos *checklist*, *concept*, *estimating guideline*, *example*, *guideline*, *reusable asset*, *supporting material*, y *tool mentor*).
- *Task-Categories*: *disciplines* y *custom categories* a las que pertenece la *task*.
- *Role-Work Products*: *responsible for* (responsable de); *work products that are output of tasks that this role performs* (productos de trabajo que son salida de tareas que realiza este rol).
- *Role-Guidances*: *guidances* relacionadas (sólo tipos *checklist*, *concept*, *example*, *guideline*, *reusable asset*, y *supporting material*).
- *Role-Categories*: *role sets* y *custom categories* a las que pertenece el *role*.
- *Work Product-Guidances*: *guidances* relacionadas (sólo tipos *checklist*, *concept*, *estimating guideline*, *example*, *guideline*, *report*, *reusable*

*asset, supporting material, template y tool mentor*).

- *Work Product-Categories: domains, work product kinds y custom categories* a las que pertenece el *work product*.
- *Guidance-Guidances: lista de guidances* (de tipo *checklist, concept, example, guideline, reusable asset y supporting material*) que pertenecen al *guidance* principal. Todos los tipos de *guidance* pueden incluir otros *guidance*, salvo el tipo *practice* que en lugar de incluir otros *guidances* está relacionado con diversos *content elements*.

## Procesos

Tal como se mostró en la Figura 7, con SPEM se distinguen dos etapas a la hora de implementar un proceso o metodología: primero se puebla el Contenido de Método, cuyos conceptos han sido presentados anteriormente, y en segundo lugar, se combinan y reutilizan dichos elementos para ensamblar actividades y procesos.

A continuación se presentan los conceptos de SPEM que se manejan para ensamblar patrones de proceso y procesos completos (ver Figura 18). No es una revisión exhaustiva ya que, por motivos de sencillez, se ha optado por no incluir los conceptos de SPEM que quedan “ocultos” al usuario cuando modela procesos con algún editor de SPEM. En los sub-partados siguientes se presentan conceptos que corresponden, principalmente, con los paquetes *Core, Process Structure, Process with Methods* (en parte) y con el *Base Plug-in* (que incluye las especializaciones predefinidas de actividad).

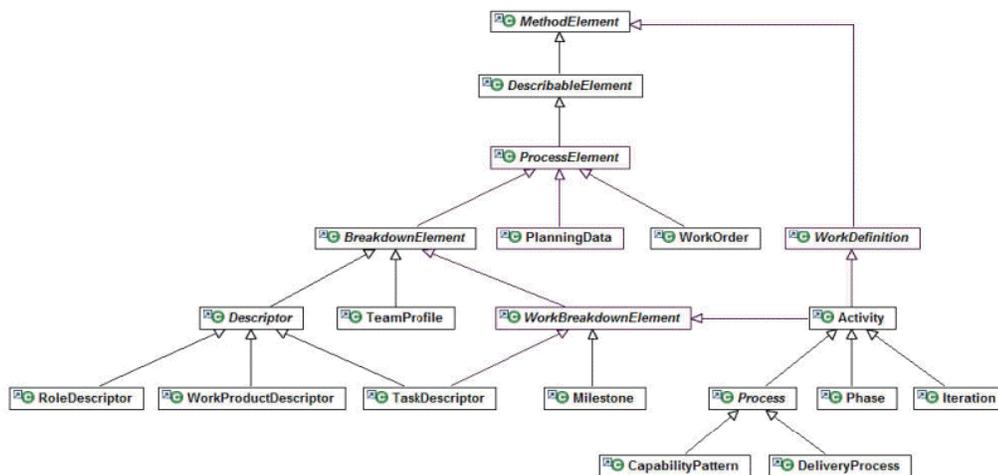


Figura 18. Jerarquía de conceptos de Proceso

### Aspectos Generales

Una Definición de Trabajo (*Work Definition*) es un concepto abstracto que generaliza todos los tipos de definiciones de trabajo en SPEM. Una Definición de Trabajo puede estar asociada con 0..\* precondiciones (restricciones que deben cumplirse para que el trabajo pueda comenzar) y con 0..\* post-condiciones (restricciones que deben cumplirse para que el trabajo pueda considerarse concluido). Las definiciones de trabajo pueden ser modeladas de forma gráfica mediante diagramas de actividad de UML.

Un Elemento de Desglose (*Breakdown Element*) es una generalización abstracta para cualquier tipo de elemento que aparece en un proceso y es parte de una estructura de desglose. Aporta tres propiedades importantes:

- Admite Varias Apariciones (*Has Multiple Occurrences*): al realizar el proceso puede haber más de una instancia del elemento.
- Es Opcional (*Is Optional*): no es obligatoria su inclusión cuando se lleva a cabo el proyecto.
- Planeado (*Is Planned*): El elemento es incluido al generar los planes

de proyecto que se exportan a las herramientas de gestión de proyectos.

Los tipos de Elementos de Desglose son:

- Parámetros de Procesos, para asociar productos de entrada/salida.
- Realizadores de Procesos, para asociar roles.
- Secuencias de Trabajo, para las relaciones de precedencia.
- Elementos de Desglose de Trabajo.
- Roles en Uso.
- Productos de Trabajo en Uso.
- Relaciones entre Productos de Trabajo en Uso.
- Asignaciones de Responsabilidad en Procesos.

Cada elemento de desglose tiene asociada una WBS que representa su estructura interna y, opcionalmente, un flujo de trabajo (*Work Flow*).

### Elementos de Desglose de Trabajo

Los Elementos de Desglose de Trabajo (WBE, *Work Breakdown Element*) son el principal tipo de elemento de desglose, ya que representan la descomposición del trabajo. Como se muestra en la Figura 19, existen dos tipos: Actividad e Hito. Las propiedades comunes a ambos son:

- Se Puede Repetir (*Is Repeatable*): puede haber varias iteraciones o repeticiones.
- Continuo (*Is Ongoing*): es un trabajo sin duración fija o estado final. Ejemplo: trabajo de un gestor de proyecto que 1 hora al día se dedica a revisar el estado de avance de las tareas.
- Condicionado por Sucesos (*Is Event Driven*): su inicio no está

determinado por eventos normales (cuando acaba el trabajo que lo precede o cuando se concluye algún producto de trabajo), sino por otro evento especial.

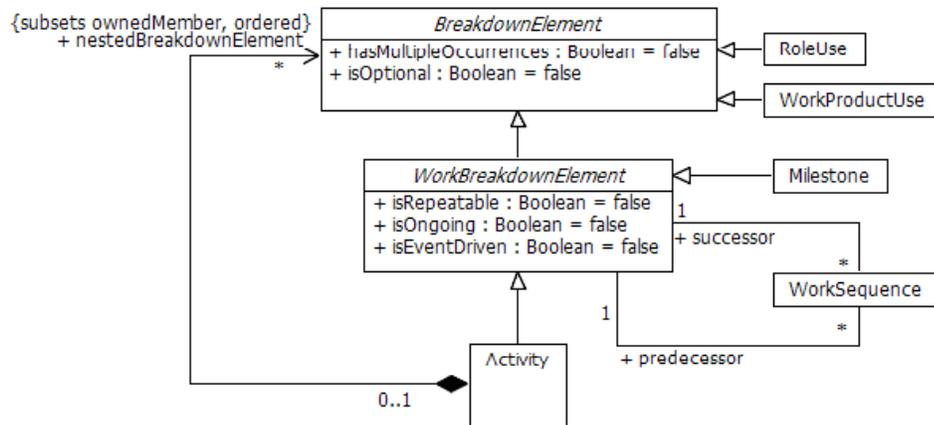


Figura 19. Flujo de trabajo y anidamiento de actividades entre WBE.

Como muestra la Figura 19, el flujo entre los WBE se representa por medio de Secuencias de Trabajo (*Work Sequence*). Cada Secuencia de Trabajo conecta dos (2) WBE, designados como predecesor y sucesor, mediante alguna de las siguientes clases (P representa al predecesor y S al sucesor):

- Acabar para Empezar: S no puede empezar hasta que no concluye P.
- Acabar para Acabar: S no puedo acabar mientras no esté acabado P.
- Empezar para Empezar: S no puede comenzar hasta que no lo ha hecho P.
- Empezar para Acabar: S no puede concluir hasta que no se inicia P.  
Se emplea en *Justin-Time*.

Ortogonalmente al flujo de trabajo, en SPEM la descomposición del trabajo en distintos niveles de detalle se realiza mediante el concepto de Actividad (*Activity*), de forma que cualquier elemento de desglose de trabajo con

estructura interna (es decir, que incluye elementos de desglose) recibe el nombre de Actividad, independiente del nivel de desglose, es decir, del tamaño del fragmento de trabajo representado. En consecuencia, una WBS se representa (Figura 19) mediante una recursividad de agregaciones: una actividad está formada por agregación de diversos elementos de desglose, entre los que se encuentran actividades más pequeñas que a su vez pueden estar formadas de igual manera.

Actividades (*Activity*): Una Actividad representa una unidad de trabajo general en un proceso. Una actividad puede tener estructura interna formada por agregación de elementos de desglose, que pueden ser de varios tipos, no solo de trabajo: actividades más simples (anidamiento), elementos de método en uso (roles en uso, productos de trabajo en uso), e hitos. La complejidad de la estructura de desglose de una actividad puede variar entre 0 tareas (pueden estar formadas sólo por productos de trabajo en uso) o todo un proceso completo. Una Actividad puede estar asociada con varias Actividades en Uso, que reutilizan la primera. Para ello, existen varias maneras de reutilización (herencia), que se presentan más adelante.

Una actividad representa una unidad de trabajo general asignable a realizadores específicos, representados por Roles en Uso. Dicha asignación se hace a través de Realizadores de Proceso (*Process Performer*). Una actividad puede tener entradas y producir salidas, representadas por Productos de Trabajo en Uso. Para ésta última asociación se utilizan los llamados Parámetros de Proceso (*Process Parameter*). La Figura 20 muestra estos vínculos entre actividades y otros elementos de proceso.

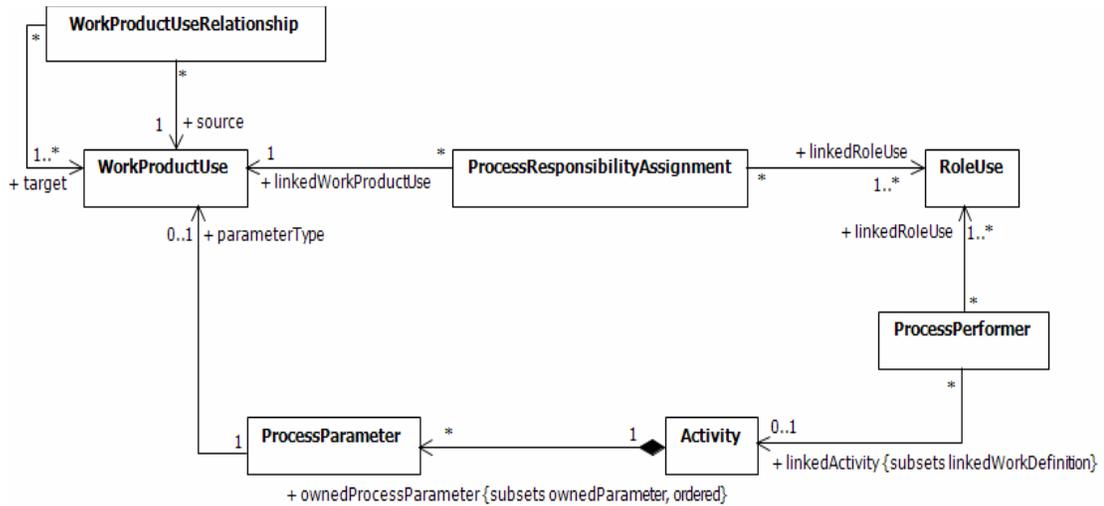


Figura 20. Relación entre Actividades y Roles y Productos de Trabajo

En la Figura 21 se muestra un ejemplo de una actividad (centro de la figura) y sus asociaciones con dos (2) Artefactos en Uso (izquierda y derecha), dos Tareas en Uso (abajo, a la izquierda y derecha), un Role en Uso (abajo en el centro) y una Secuencia de Trabajo precediendo a otra Actividad (arriba a la izquierda).

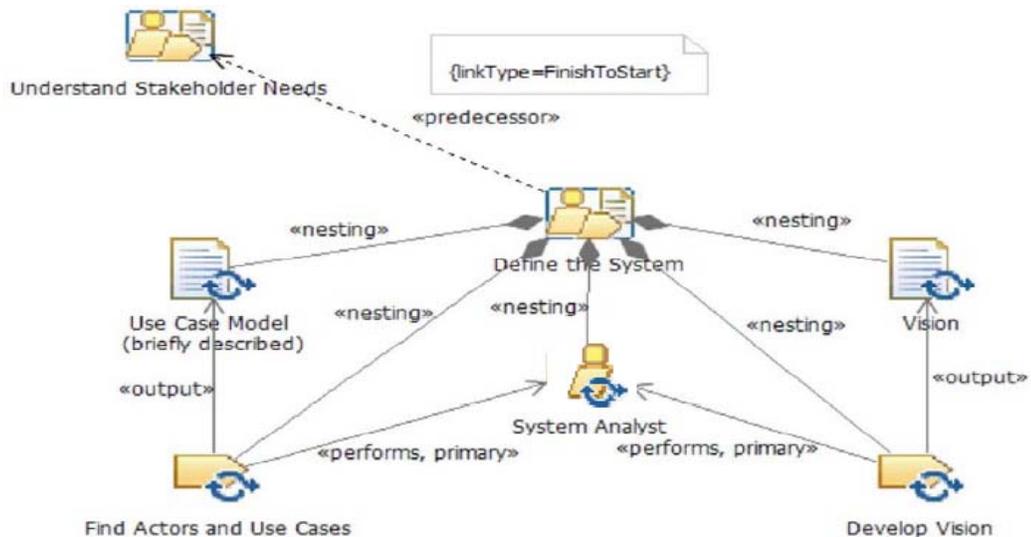


Figura 21. Ejemplo de una actividad y sus asociaciones

SPEM tiene predefinidos varios tipos especiales de Actividades: Iteración, Fase y Proceso. Puesto que Actividad es un concepto genérico, que incluye a proceso como especialización, en SPEM no se respeta la nomenclatura tradicional de los estándares ISO de IS. En especial, debe tenerse en cuenta que SPEM no sigue los tres niveles estáticos de descomposición utilizados en el modelo de procesos de ISO/IEC 12207: procesos formados por actividades formadas por tareas.

Contenido de Método en Uso: Como ya se comentó, los elementos del contenido de método (tareas, roles y productos de trabajo) reutilizados en los procesos reciben el nombre de elementos en uso. Un elemento en uso es una instancia de un elemento de método particularizada para un contexto de proceso determinado. Por ello, los elementos en uso no son reutilizables (para reuso ya están las descripciones de los elementos en el contenido de método). Así, una Tarea en Uso (*Task Use*) representa la ocurrencia de una Definición de Tarea (Contenido de Método) en el contexto de una Actividad. En ella se pueden precisar y modificar, respecto de la tarea original, su documentación, pasos, roles y productos de entrada y de salida. Además, se pueden definir dos asociaciones que no aparecen en las tareas como elementos de contenido de método: los roles que asisten en la tarea y las entradas externas. Existen dos maneras de utilizar tareas en actividades:

- Reutilizar una descripción de tarea del contenido de método, que pasa a ser una tarea en uso, o
- Crear directamente en la actividad una instancia de tarea en uso. En este caso no se tendrá descripción asociada, sino sólo el nombre.

Un Producto de Trabajo en Uso (*Work Product Use*) representa la ocurrencia de un Producto de Trabajo real en el contexto de una Actividad, pudiendo

modificar su documentación. Además, se pueden definir dos atributos que no aparecen en la definición de producto de trabajo en el contenido de método: los estados de entrada y de salida del producto en la actividad. Por ejemplo, un producto “Requisitos” puede entrar en una actividad en el estado “identificados” y salir en el estado “refinados”. También existen las dos (2) maneras de utilizar productos de trabajo en actividades equivalentes a las dos indicadas para las tareas:

- Reutilizar una descripción de producto de trabajo del contenido de método, que pasa a ser un producto de trabajo en uso.
- Crear directamente en la actividad una instancia de producto de trabajo en uso, que sólo tendrá nombre pero no descripción asociada.

De forma similar a las dos anteriores, un Rol en Uso (*Role Use*) representa la ocurrencia de un Rol real en el contexto de una Actividad, pudiendo particularizar la documentación, productos de los que es responsable o que modifica, y equipos (roles compuestos) a los que pertenece. Igualmente, existen dos maneras de utilizar roles en actividades:

- Reutilizar una descripción de rol del contenido de método, que pasa a ser un rol en uso. SPEM le llama Realizador (*Performer*).
- Crear directamente en la actividad una instancia de rol en uso. En este caso no se tendrá descripción asociada, sino sólo el nombre. SPEM le llama Participante (*Participant*).

Un Rol Compuesto (*Composite Role*) es un rol en uso especial, que se corresponde con más de una Definición de Rol del Contenido de Método.

En la Figura 19 se muestra un ejemplo de de Contenido de Método en Uso (parte derecha) referenciando a Contenido de Método (parte izquierda). La

Figura 22 muestra otro ejemplo de una Tarea en Uso dentro de una Estructura de Desglose de Trabajo.

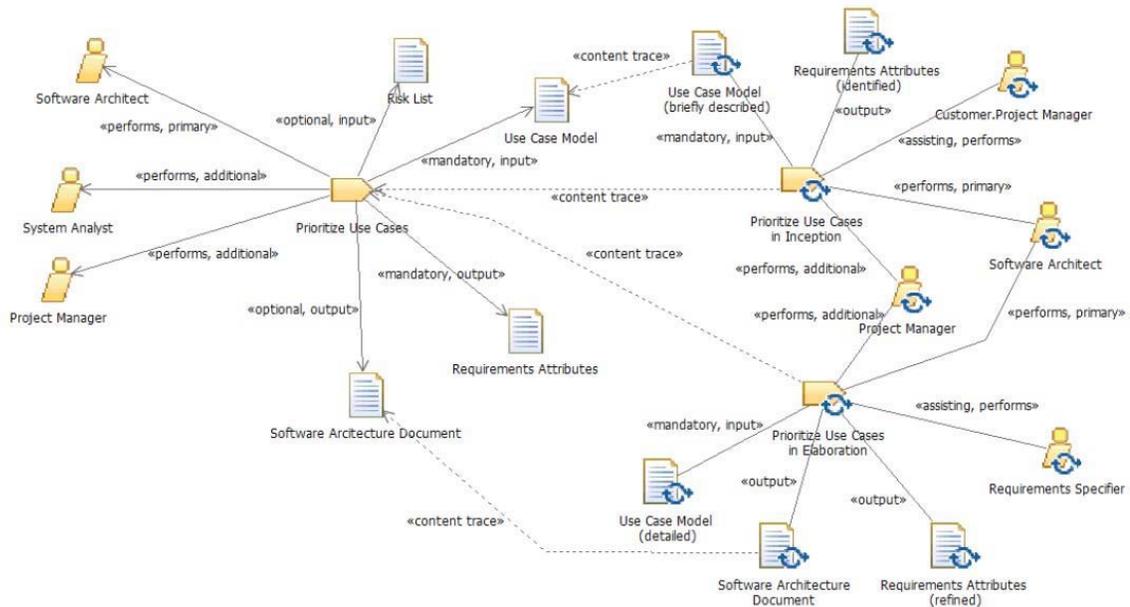


Figura 22. Ejemplo de Contenido de Método en Uso.

Fases, Iteraciones e Hitos: Como ya se ha indicado, las fases e iteraciones son dos tipos especiales de actividades (adicionales a los procesos), mientras que los hitos son un tipo de elemento de desglose de trabajo, distinto de los otros dos, actividades y tareas.

Una Fase (*Phase*) representa un periodo de tiempo que es significativo para un proyecto, y que acaba con un punto de control de gestión importante, un hito o un conjunto de entregables concluidos. En la práctica, en SPEM una fase es una actividad que tiene el valor falso en la propiedad "Es repetible".

Por el contrario, una Iteración (*Iteration*) representa un conjunto de actividades anidadas que se repiten más de una vez. Sirve para organizar

ciclos repetitivos de trabajo. En la práctica, en SPEM una iteración es una actividad que tiene el valor verdadero en la propiedad “Es repetible”.

Un Hito (*Milestone*) representa un evento significativo para el desarrollo de un proyecto:

- Una decisión importante,
- La conclusión de un entregable,
- La conclusión de una fase, etc.

Según se muestra en la Figura 19, los hitos son elementos de desglose del trabajo y, por tanto, aparecen en la WBS y pueden tener relaciones de precedencia.

#### Tipos de Procesos

En SPEM, un Proceso (*Process*) es un tipo de Actividad que describe una estructura para tipos particulares de proyectos o partes de ellos. Mediante un proceso de SPEM se pueden representar distintos tipos de métodos de IS: un proceso, un modelo de procesos completo, un ciclo de vida con sus diversos procesos, o una metodología completa.

Los PDS definen cómo serán ejecutados los proyectos software. Por ello, para ayudar a cumplir los requisitos del nivel 3 de CMMI (*Capability Maturity Model Integration*): proceso definido, SPEM provee mecanismos de adaptación y ensamblado dinámico, basados en patrones de proceso reutilizables:

- Reutilización (Actividad en Uso).
- Particularización (Elementos con Clases - *Kinds*).

- Especificación descriptiva (Elementos de Desglose).
- Variabilidad (*Plug-ins*).

En SPEM existen dos clases principales de procesos:

#### Patrón de Proceso (*Capability Pattern*)

Es un “fragmento de proceso” que describe un grupo de actividades reutilizable como solución a algún tipo de problema o situación habitual. Se definen para poder ser empleados más de una vez en uno o varios procesos o con fines de organización. Se pueden almacenar en una jerarquía de Paquetes de Proceso (*Process Package*). Algunos escenarios de uso de patrones de proceso son:

- Servir como bloques para construir Procesos para Despliegue o Patrones de Proceso más complejos.
- Ayudar a la ejecución de proyectos que no siguen un proceso bien definido, sino que trabajan en base a fragmentos de proceso (buenas prácticas) de una manera flexible (métodos ágiles).
- En formación, para describir el conocimiento de una cierta área clave, buena práctica, disciplina, etc.

#### Proceso para Despliegue (*Delivery Process*)

Describe una aproximación completa e integrada para realizar un tipo específico de proyecto, abarcando un ciclo de vida completo de desarrollo o mantenimiento. Sirven como plantillas para planificar y ejecutar los proyectos. En un Proceso de Despliegue se ensamblan Patrones de Proceso y Elementos en Uso (tareas, roles y productos de trabajo en uso). Ejemplos:

RUP, XP, METRICA 3, etc.

Existe un tercer tipo de proceso llamado Plantilla para Planificación de Procesos (*Process Planning Template*), que contiene la misma información que los otros tipos pero ampliada con ciertas decisiones que se refieren a un plan de proyecto concreto, por ejemplo, se incluyen asignaciones de personal o fechas. Por tanto, este tipo de proceso no es realmente un nuevo tipo de proceso sino una adaptación de los anteriores de cara a la exportación a una herramienta de gestión de proyectos, es decir, a poder generar un archivo que sirva de plantilla para un plan de proyecto concreto.

Todos los procesos, independientemente del tipo que sean, tienen en común las siguientes propiedades: *name* (nombre), *presentation name* (nombre de presentación), *brief description* (descripción breve), *external ID* (ID externo), *purpose* (objetivo), *main description* (descripción principal), *scope* (ámbito), *usage notes* (notas de utilización), *alternatives* (alternativas), *how to staff* (cómo proveer de personal), *key considerations* (factores clave). Adicionalmente, un proceso para despliegue tiene éstas otras propiedades: *scale* (escala), *project characteristics* (características del proyecto), *risk level* (nivel de riesgo), *estimating techniques* (técnicas de estimación), *project member expertise* (especialidad de miembros del proyecto), *type of contract* (tipo de contrato).

Los patrones de proceso pueden ser incorporados como elementos de desglose a una WBS. Esto implica la reutilización automática de todos sus elementos y estructura (actividades, tareas, roles, productos de trabajo, WBS, etc.).

Reutilización y Variabilidad

La organización en *plug-ins* permite que se puedan reutilizar los elementos de contenido y los procesos definidos en una Biblioteca (*Library*). Dicha reutilización se puede realizar de dos maneras:

- Al crear un *plug-in* nuevo se puede referenciar a otros *plug-ins*.
- Usar de forma directa el contenido de un *plug-in* desde otro *plug-in* diferente.

A veces interesa reutilizar el contenido de un *plug-in* con ciertas modificaciones. Para ello existe el mecanismo de variabilidad (*Variability*), que permite modificar elementos de método o de proceso sin modificar directamente el original. La variabilidad de un elemento de método o de proceso permite definir diferencias (adiciones, cambios, omisiones) con el elemento original. Dichas diferencias afectan a las propiedades, es decir, a los atributos y a las asociaciones con otros elementos.

#### Variabilidad de Elementos de Contenido

SPEM contempla cinco (5) tipos de variabilidad entre elementos de contenido: No asignado, Contribuye, Amplía, Reemplaza, y Amplía y Reemplaza. A continuación se describen sus características.

##### No asignado (*not assigned*)

No existe relación de variabilidad entre el elemento en cuestión y otros. Es el valor por defecto.

### Contribuye (*contributes*)

Un elemento E que contribuye a un elemento base EB añade sus valores de atributos e instancias de asociación a EB sin modificar directamente las propiedades que ya tiene dicho elemento base (es una adición). Al generar una vista (por ejemplo, publicar en la web), el elemento base EB se muestra combinado con los atributos y relaciones de E, mientras que E queda oculto. Las reglas que rigen dicha combinación son (ver ejemplos en la Figura 23 y resultado en la Figura 24):

- El valor de los atributos (campos de texto) de E se concatena al final del contenido de los correspondientes atributos de EB.
- Las instancias de asociaciones de entrada o de salida con cardinalidad “muchos” de E se añaden a EB.
- Las instancias de asociaciones de entrada y de salida con cardinalidad “1” (como la relación “realizador principal” entre una tarea y un rol) definidas en E se ignoran si ya existen en EB. En caso contrario, se añaden.
- Un elemento base puede recibir varias contribuciones.
- La contribución es transitiva.



## Reemplaza (*replaces*)

Mediante este tipo de variabilidad un elemento E puede reemplazar (sustituir) los atributos y las asociaciones de salida de un elemento base EB sin modificar directamente ninguna propiedad de EB. El elemento E que reemplaza se incluirá en las vistas generadas mientras que el reemplazado EB no aparecerá. Las reglas que regulan el reemplazo son (un ejemplo se muestra en la Figura 25 y el resultado en la Figura 26):

- Los valores de los atributos de EB se reemplazan por los respectivos valores de los atributos no vacíos de E, incluyendo los identificadores.
- Las instancias de asociación de salida de EB se reemplazan por las de E.
- Las instancias de asociación de entrada con cardinalidad “muchos” de EB se amplían con las instancias de dichas asociaciones existentes para E.
- Las instancias de asociación de entrada con cardinalidad “uno” de EB se dejan intactas. Las instancias incluidas en E que no están en EB se añaden a EB.
- Un elemento base sólo puede ser reemplazado por otro único elemento en una misma configuración. Si se define más de un reemplazo para un mismo elemento base, entonces no se realizará ningún reemplazo.
- El reemplazo es transitivo.

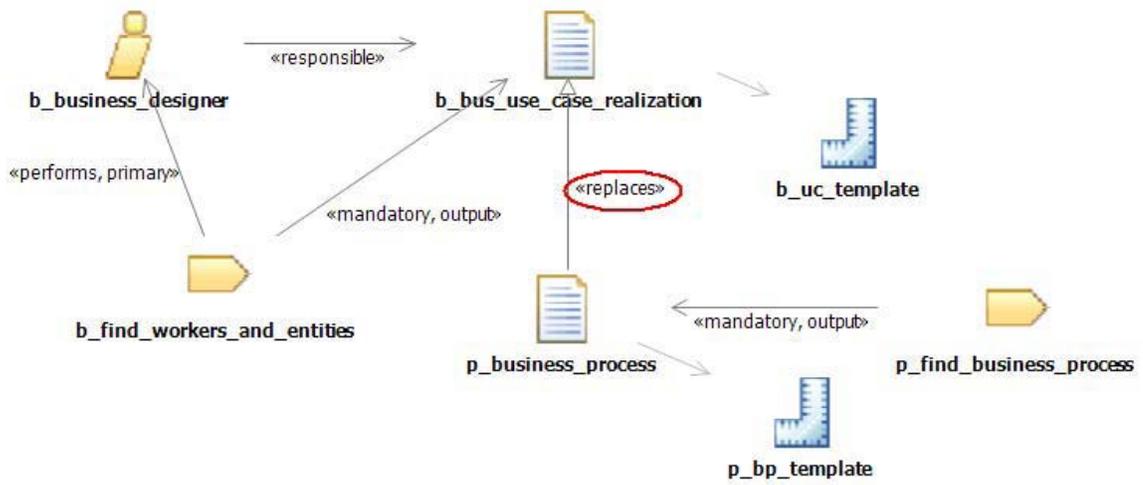


Figura 25. Ejemplo de variabilidad de tipo “reemplaza”

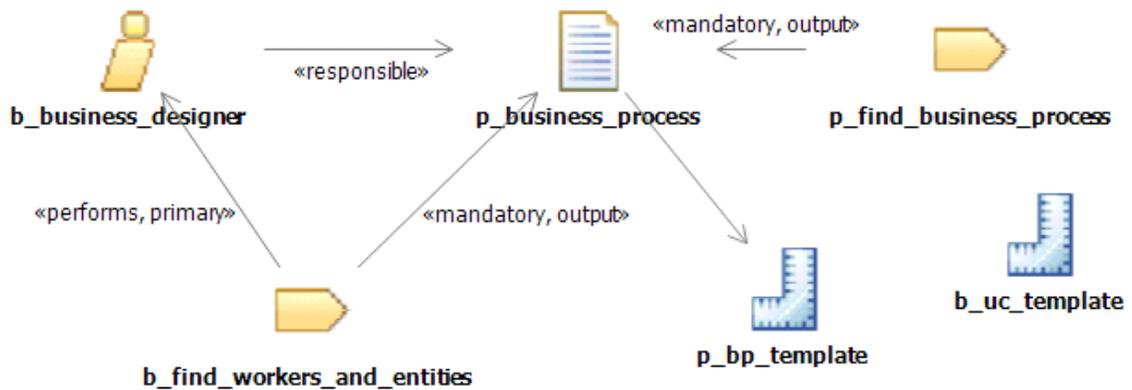


Figura 26. Resultado de la variabilidad de la Figura 25

### Amplía (*extends*)

En este caso, el elemento E hereda las propiedades del elemento base EB que amplía, dejando EB totalmente intacto. El elemento E incluye las propiedades heredadas de EB más las propias (por ejemplo, nuevas asociaciones). Tanto EB como E aparecen en las vistas generadas. La Figura 27 muestra un ejemplo de esta clase variabilidad y la Figura 28

permite comprobar el resultado. Las reglas que la regulan son:

- Los valores de los atributos de EB son heredados por E en caso de que los equivalentes suyos estén vacíos. Si se definen campos en E, los respectivos campos de EB se ignoran.
- Las instancias de asociación de salida con cardinalidad “muchos” de EB se añaden a E.
- En las instancias de asociación de salida con cardinalidad “1” se ignora la instancia de EB si E ha definido la suya propia. En caso contrario, E hereda la de EB.
- Las asociaciones de entrada de EB se ignoran (no se añaden a E).
- Las asociaciones de ampliación son transitivas.

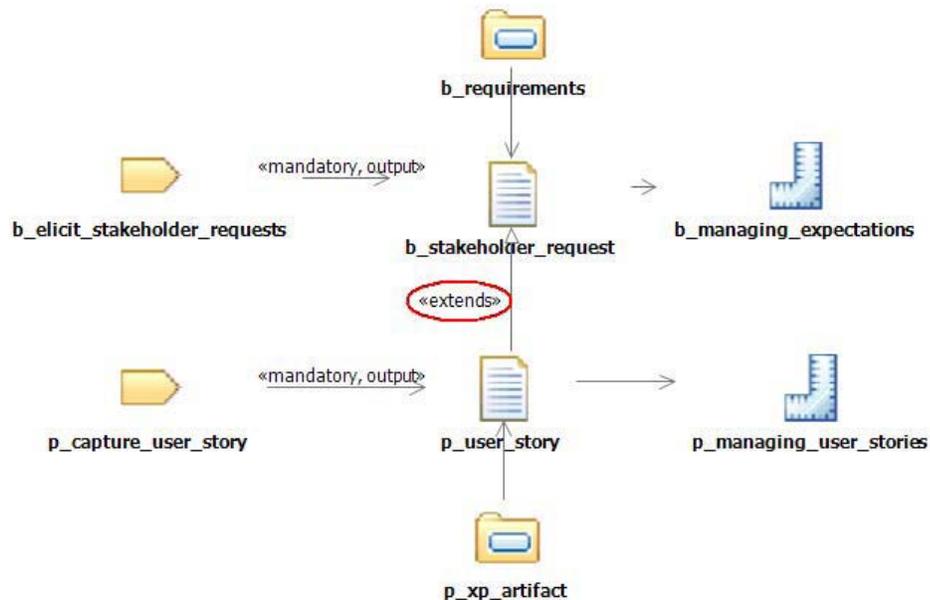


Figura 27. Ejemplo de variabilidad de tipo “amplía”

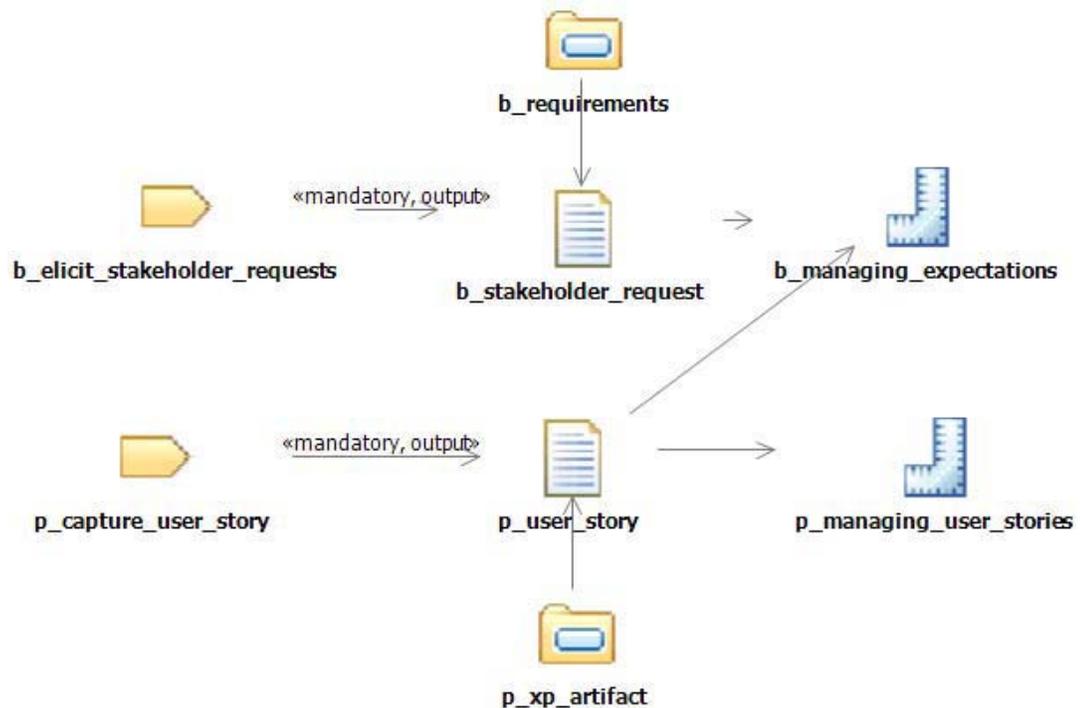


Figura 28. Resultado de la variabilidad de la Figura 27

### Amplía y Reemplaza (*extends-replaces*)

Este tipo de variabilidad combina los efectos de las variabilidades de tipo Amplía y de tipo Reemplaza, ya comentadas. El resultado es que las propiedades que no se han definido en el elemento E que amplía y reemplaza se heredan del elemento base EB. En cambio, las propiedades definidas en E sustituyen a las de EB. E aparece en las vistas que se generan, pero EB no. Este tipo de variabilidad se utiliza al generar *plug-ins* que renombran elementos, reemplazan descripciones, etc., sin remodelar completamente todas las relaciones y atributos del *plug-in* base. Las reglas que rigen esta clase de variabilidad son (ver ejemplo en Figura 29 y su resultado en Figura 30):

- Si el elemento E que amplía y sustituye define sus propias

asociaciones de salida, reemplazarán a las del elemento base EB. Las instancias de asociaciones de salida no definidas en E se heredan de EB.

- Las instancias de asociaciones de entrada de EB se añaden a E.
- Los valores de los atributos no vacíos de E reemplazan a los valores de los mismos atributos de EB. del elemento base. Si E no define valor para un atributo, el valor se hereda de EB.
- La relación amplía y reemplaza es transitiva.
- Un elemento base sólo puede ser reemplazado o ampliado y reemplazado por otro único elemento en una misma configuración. En caso contrario no se realizará ningún reemplazo.

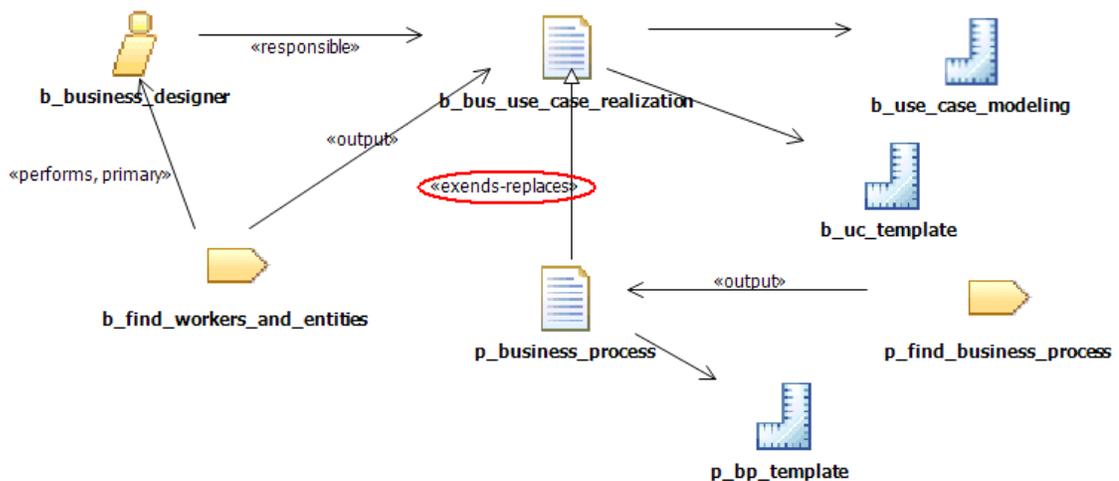


Figura 29. Ejemplo de variabilidad de tipo “amplía y reemplaza”.

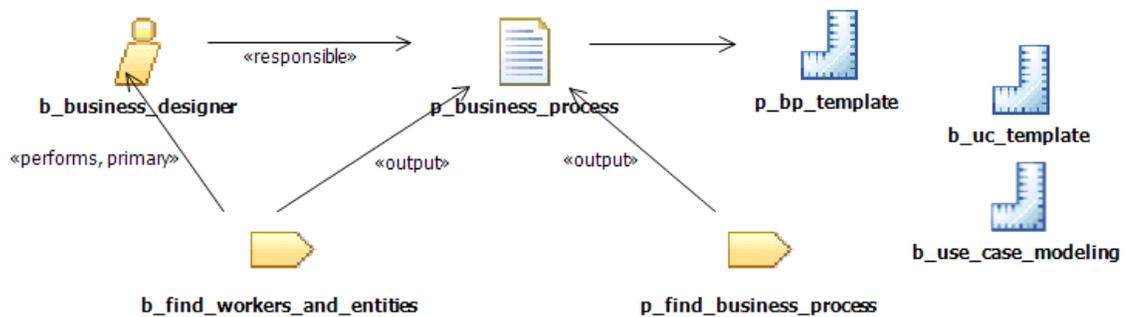


Figura 30. Resultado de la variabilidad de la Figura 29.

Cuando se resuelven varias variabilidades, las reglas de prioridad son las siguientes:

- Primero se realizan las asociaciones de variabilidad de tipo “Contribuye”, seguidas de las de tipo “Reemplaza”, luego de las de tipo “Amplía” y, por último, las de tipo “Amplía y Reemplaza”.
- Entre dos variabilidades del mismo tipo, primero se realiza la que está más arriba en la jerarquía.

### Reutilización y Variabilidad de Elementos de Proceso

Una manera de definir procesos es aplicar una estrategia *bottom-up* desde lo pequeño hacia lo grande. Al seguirla, primero se definen patrones de proceso sencillos, que no incluyen jerarquía de actividades, sino simplemente descriptores de tarea, de rol y de producto). Después se crean patrones más complejos en los que se reutilizan los patrones más sencillos definidos anteriormente. De esta forma se van componiendo los procesos reutilizando bloques de proceso cada vez más complejos, hasta llegar a los procesos para despliegue.

En general, los elementos de proceso de tipo actividad (fase, iteración,

actividad, patrón de proceso) se pueden reutilizar en otros elementos de proceso más grandes. En este caso reciben el nombre genérico de Actividad en Uso (*Activity Use*). Una Actividad en Uso en SPEM define la habilidad para reutilizar las estructuras definidas en los elementos de desglose de una actividad en una segunda actividad, sin necesidad de copiar físicamente dichas estructuras. Por tanto, se trata de una forma para heredar dinámicamente dichas estructuras desde la actividad referenciada.

SPEM establece varios mecanismos de reutilización de una actividad (*Activity Use Kind*):

No asignado (*not assigned*)

Cuando una actividad no tiene asociaciones de Actividad en Uso.

Extensión (*extension*)

Una actividad extendida A hereda la estructura interna (subestructuras e instancias de asociación) de una actividad base AB. Se pueden añadir nuevos elementos de desglose a A, pero no se pueden modificar los elementos de desglose heredados desde AB. Los cambios realizados en A no afectan a AB. En cambio, los cambios realizados en AB con posterioridad se reflejan automáticamente en A y en las demás actividades en uso que extienden AB.

Contribución local (*local contribution*)

Sirve para definir adiciones locales específicas (llamadas contribuciones) a los elementos de desglose heredados vía la extensión del tipo anterior. Por

tanto, para que existe una asociación de este tipo entre una actividad A y otra actividad base AB es necesario que existe previamente la asociación de tipo extensión entre ambas.

#### Reemplazo local (*local replacement*)

De forma similar al anterior, sirve para definir sustituciones locales específicas (llamadas reemplazos) a los elementos de desglose heredados vía la extensión. Igualmente, para que existe una asociación de este tipo entre una actividad A y otra actividad base AB es necesario que existe previamente la asociación de tipo extensión entre ambas. Las instancias de asociación de este tipo sustituyen subelementos de AB con elementos propios de A.

No existe un tipo “Eliminación local”, es decir, no es posible eliminar en una Actividad en Uso elementos de desglose heredados de otra actividad mediante el tipo “Extensión”. Por tanto, al reutilizar una actividad se debe tener en cuenta que se van a reutilizar todos los elementos contenidos en la actividad. Esta restricción en el meta-modelo SPEM es debida a razones de integridad y consistencia, pero no supone un problema en la práctica porque existen opciones para no incluir ciertos elementos heredados en las vistas generadas de un proceso.

Debido a lo anterior, el escenario habitual de reutilización de elementos de proceso consiste en reutilizar primero una cierta actividad mediante una asociación de extensión, seguida después de un refinamiento, mediante contribuciones y reemplazos locales, de la actividad en uso resultante. Por otro lado, los elementos de proceso de tipo actividad (fase, iteración, actividad, patrón de proceso) permiten variabilidad de los tipos siguientes:

Amplía, Contribuye y Reemplaza.

Las características de estos tipos son las mismas que las explicadas para la variabilidad en los elementos de contenido de método.

### Configuraciones de Método

Una Configuración de método (*Method Configuration*) es una selección de contenidos de los *plug-ins* de una Biblioteca de métodos (*Method Library*) de forma que se limita el espacio visual de la biblioteca al subconjunto seleccionado. En consecuencia, es el mecanismo ofrecido en SPEM para poder tener diferentes vistas de una misma biblioteca de *plug-ins* o de un único *plug-in*. La definición de una configuración puede estar basada en las definiciones de otras configuraciones. Por ejemplo, una configuración A podría definirse como el superconjunto de las configuraciones B, C y D, más ciertos *plug-ins* y paquetes adicionales. En este caso, se dice que B, C y D son las configuraciones base de A. Si alguna de estas configuraciones base cambian, los cambios serán automáticamente válidos también para A, reduciendo de esta manera el esfuerzo de mantenimiento.

Una configuración de método define un subconjunto lógico de una biblioteca de métodos mediante la selección (filtro) de los paquetes deseados, tanto de contenido de método como de procesos. La Figura 28 muestra un ejemplo, indicando con color rojo (fondo resaltado) las partes de una biblioteca de *plug-ins* que se incluye en una cierta configuración de método. La selección de elementos incluidos en una configuración se puede refinar añadiendo o sustrayendo de la configuración todos los elementos asociados con una cierta categoría.

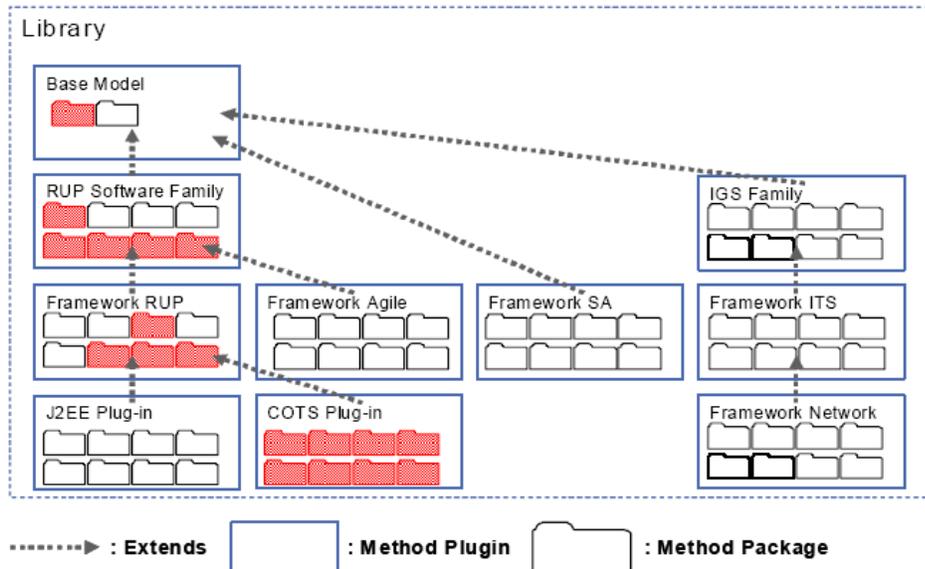


Figura 31. Ejemplo de una configuración de método (elementos en rojo)

Una actividad representando un proceso puede ser válida para diferentes configuraciones de método, pero en cada configuración se incluye o excluye contenido particular para situaciones específicas. Por ejemplo, un proceso definido para desarrollar esquemas de bases de datos puede tener dos configuraciones, una con la vista particular para SGBD (Sistemas de Gestión de Bases de Datos) relacionales y otra para el caso de sistemas OO. Por tanto, este mecanismo ofrece una alternativa en la creación de procesos, ya que puede definirse un proceso general que incluya contenidos para varios tipos de proyectos más específicos. Mediante el uso de configuraciones se seleccionan los contenidos de cada uno de esos tipos de proyectos específicos. Crear vistas para los diversos tipos de proyecto es sencillo, ya que basta con seleccionar la configuración adecuada.

## MARCO METODOLÓGICO

Comprender una disciplina implica aprendizaje, es decir, observación,

reflexión y encapsulación de conocimiento, construcción de modelos (del dominio de aplicación, de los procesos para resolver problemas, etc.), experimentación y evolución de dichos modelos con el tiempo (Basili, 2000). Este paradigma ha sido utilizado en campos tan diversos como la medicina, la física, o la industria manufacturera; pero las únicas diferencias entre unos campos y otros estriban en cómo se construyen y analizan los modelos, y cómo se lleva a cabo la experimentación.

Desde un punto de vista científico, la investigación en IS es de carácter experimental. Se centra en conocer la naturaleza de los procesos, productos e interrelaciones entre ellos en el contexto de un sistema software o de un sistema organizacional. Gran parte de la investigación llevada a cabo en esta área es de tipo cuantitativo y está basada en técnicas estadísticas. Sin embargo, en los últimos años los métodos de investigación cualitativos han merecido la atención y aceptación de la comunidad investigadora.

El presente trabajo parte de un enfoque de investigación cualitativo, según una clasificación ampliamente aceptada de los métodos de investigación hecha por Hernández y cols. (2003), porque se basa en documentos, impresiones y reacciones del investigador, entre otros, sin medición numérica (evitando la cuantificación). Así mismo, según la visión de investigación científica expresada por Arias (2006) y Sabino (1978; 1994; 2000), el nivel de investigación es exploratorio-descriptivo, porque se propone alcanzar una visión general del objeto de estudio, por ser de interés para elaborar una descripción sistemática que pueda ser utilizada en trabajos de investigación posteriores. El diseño de investigación es de campo, porque el conocimiento es obtenido directamente de la experiencia empírica; mediante la incorporación de un caso de estudio.

Para la resolución del problema planteado se estudiaron diversos métodos de investigación, debido a que la naturaleza del problema no responde directamente a la aplicación de un método o metodología de desarrollo de software sino a la búsqueda del conocimiento sistemático y tecnológico del objeto de estudio. De esta manera, la estrategia general de investigación sigue los principales lineamientos del método propuesto por Marcos (2002), que se adaptó para efectos de esta investigación al añadir tareas necesarias a las etapas originalmente propuestas e incorporar otros métodos con el propósito de encarar algunas de las fases y actividades de la investigación. Entre los métodos incorporados, se utilizó la propuesta metodológica de Caro y cols. (2005) que plantea una serie de actividades que guían la realización de revisiones documentales en proyectos de fin de carrera; así mismo se incorporó un caso de estudio con la idea de obtener conocimiento y dominio tecnológico del objeto de estudio (Sabino, 1978). Es importante resaltar que la adaptación de estos métodos de investigación está relacionada con un importante componente de creatividad, ineludible para la configuración del entorno tecnológico de trabajo, la selección de la herramienta de software para el soporte del SPEM y para el ensamblaje de la estrategia general para la resolución del problema.

A continuación se presentan tanto la estrategia general de investigación como cada uno de los métodos utilizados para configurar la misma, de modo que pudiera ser adaptada a este trabajo específico.

### **Métodos de investigación en Ingeniería de Software**

Existen variadas clasificaciones de ciencias (Bunge, 1976), basadas en diferentes criterios. Dependiendo del tipo de ciencia, se utilizan unos u otros métodos de investigación (Chalmers, 1984). Sin embargo, ninguno de estos

métodos parece ser totalmente apropiado para la investigación en IS. La distinta naturaleza del saber de las ingenierías, con respecto al saber de las ciencias empíricas y formales, hace que los métodos de investigación de estas últimas ciencias no sean directamente aplicables a la investigación en el ámbito de la IS. Por este motivo, la búsqueda de un método apropiado se está convirtiendo en un tema de investigación en sí mismo.

Quizá, una de las clasificaciones de Ciencia más comúnmente aceptada es aquella que divide las ciencias en formales (Lógica y Matemáticas) y empíricas (entre las que se encuentran la Biología, la Química, etc.). Mientras las primeras emplean métodos de investigación deductivos, las segundas se basan en métodos de investigación empíricos (inductivo e hipotético-deductivo). Para poder llegar al tipo de método de investigación apropiado para la IS, es preciso determinar previamente cuál es la naturaleza del conocimiento en la IS y cuál su objeto de estudio.

Dependiendo pues del objeto que trate cada Ciencia, encontraremos un método más o menos adecuado para su estudio. Pero, ¿cuál es el objeto de estudio de la IS? Las ciencias empíricas y formales, a fin de encontrar respuesta a numerosos interrogantes sin resolver, se centran en el estudio de objetos existentes para obtener respuestas a dichos interrogantes a través de la creación de hipótesis y modelos, la observación y la experimentación.

Si la investigación se centra en el cómo construir nuevos objetos, necesitaremos nuevos métodos; si la investigación se centra en estudiar dichos objetos (o los objetos que permiten construir) quizá nos valga con adaptar alguno de los métodos usados por las ciencias tradicionales, ya que éstas estudian fenómenos y objetos del mundo de manera independiente a

cómo fueron creados. La investigación en IS trata, por tanto, distintos problemas que deberán ser abordados con distintos métodos.

Una vez determinado el objeto de estudio, podemos ver qué tipos de métodos se adecúan más a cada tipo de investigación. Lógicamente, y como ocurre con cualquier clasificación, habrá problemas que caigan en más de una de estas clases y que requieran, por tanto, la combinación de diferentes métodos.

Aunque los métodos de investigación pueden clasificarse de diversos modos, una clasificación ampliamente aceptada en la actualidad y suficiente para ubicar este Trabajo de Grado en el contexto científico, es la que divide a los métodos en cuantitativos y cualitativos. Los métodos, deductivos y empíricos, podrían encuadrarse dentro de lo que se denominan métodos de investigación cuantitativos y son especialmente apropiados para el estudio de fenómenos u objetos naturales. Sin embargo, el estudio de fenómenos culturales y sociales requiere otro tipo de métodos, que no se basen en experimentos ni teorías formales, sino en entrevistas, cuestionarios, documentos, impresiones y reacciones del investigador, etc. Reciben el nombre de métodos cualitativos y entre ellos se encuentran la investigación en acción, los casos de estudio, la etnografía, etc.

En una importante aportación realizada por Marcos (2002) con el fin de encontrar un método apropiado para la investigación en IS se hace referencia a un tercer tipo de método de investigación al que se le denomina métodos de investigación creativos y se plantean como aquellos que utilizan mayoritariamente las artes, si bien creatividad y ciencia están siendo cada día más relacionadas (Standler, 1998). Aunque la creatividad podría verse como una característica de la investigación, independiente del método, hay

ciencias donde la investigación requiere de un alto grado de creatividad en oposición a la observación o la experimentación. Tal es el caso de las artes y de las ingenierías en cuanto al fuerte componente artístico de las mismas. Cuando la creatividad marca el proceso de investigación, se habla de métodos creativos. Estos métodos se basan en características como la imaginación, premonición, visualización, entre otros y en ellos interviene la inteligencia creativa del investigador por encima de la racional.

### **Metodología para la investigación en Ingeniería de Software**

Para realizar investigación en IS, Esperanza Marcos (2002), miembro del grupo Kybele, desarrolló una propuesta metodológica para llevar a cabo su tesis doctoral en la Universidad Rey Juan Carlos de España. Lo interesante y novedoso de la misma es que resalta el paralelismo existente entre el método de investigación en IS y el método de desarrollo de software con el fin de encontrar un método apropiado para la investigación en IS. Esta propuesta está constituida por siete (7) pasos que por su generalidad son aplicables, con ciertas modificaciones, a cualquier tipo de investigación. A continuación, se muestra cada uno de los pasos en la Figura 32 y seguidamente se describen tal como se proponen originalmente.

#### **Búsqueda de documentación**

La búsqueda de documentación aparece, generalmente, como la primera etapa a realizar en toda investigación. Sin embargo, se debe buscar y analizar documentación durante todo el proceso de investigación, intensificándose durante la determinación del problema y la etapa de resolución y validación; incluye documentación acerca del problema a resolver y documentación relacionada con el método de resolución y

validación. La labor de búsqueda de documentación deberá prolongarse hasta que se finalice el proceso de investigación, a fin de mantenerse al día sobre otros trabajos relacionados que pudieran estarse realizando.

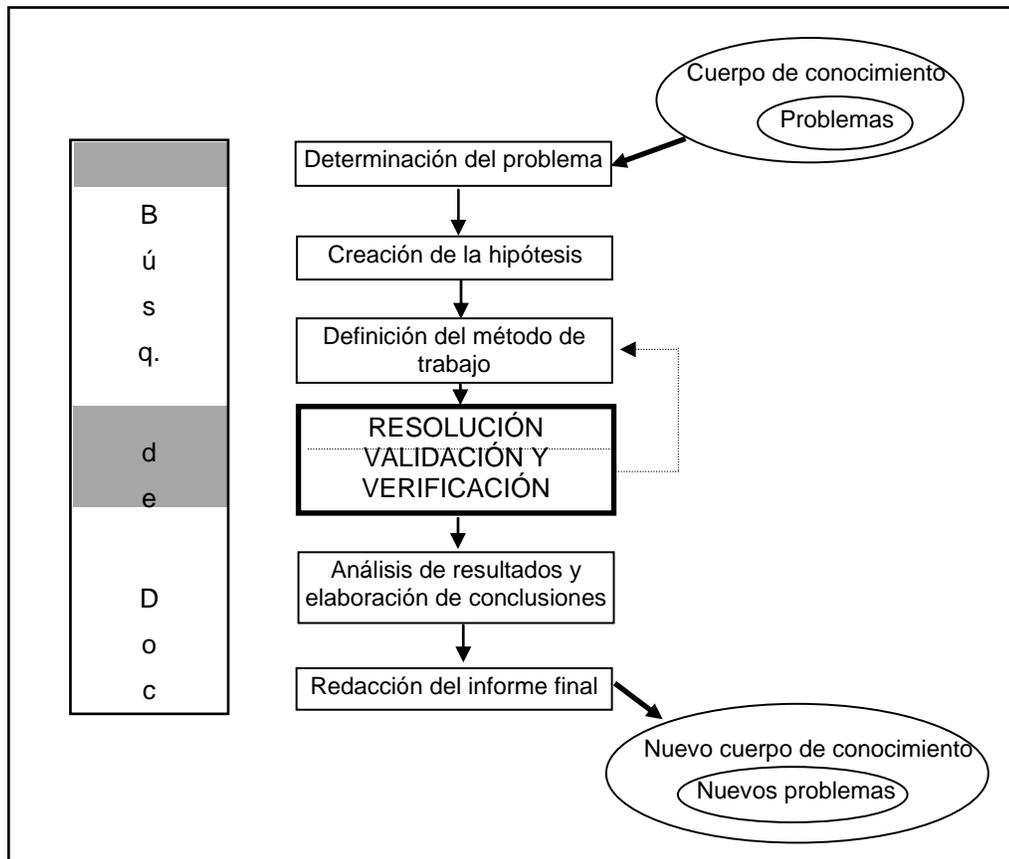


Figura 32. Propuesta metodológica de Marcos (2002)

### Determinación del problema

En esta etapa se trata de determinar y definir claramente, partiendo de los problemas sin resolver dentro del campo de conocimiento (en el que el investigador se desenvuelve), el problema que se va a abordar. Esta etapa tiene grandes similitudes con la captura de requisitos en el desarrollo de software. La captura de requisitos permite realizar un análisis del problema a

abordar, así como delimitar los aspectos concretos que se tendrán en cuenta para el futuro sistema software.

### Creación de la hipótesis

En los métodos tradicionales de investigación científica la hipótesis se formula en términos causales (sí ocurre A entonces ocurre B). Estas hipótesis son conjeturas de hechos que el método científico deberá contrastar y verificar. No es este el caso de la investigación aquí realizada. La hipótesis se formulará como la descripción del nuevo objeto que se desea construir por lo que se corresponde con la especificación de requisitos de software, producto de trabajo (artefacto) obtenido como resultado de la disciplina de captura de requisitos. La hipótesis será pues, la especificación de requisitos del nuevo objeto a construir.

### Definición del método de trabajo

Aunque, en general, los métodos de investigación no plantean la definición del método a seguir como una tarea a realizar para la resolución y verificación del problema, es consideración del autor del método que ésta es necesaria ya que no existe un método universal para la resolución de problemas sino que cada problema requiere su propio método. Esta etapa parte de la elección del paradigma metodológico a seguir, así como el método concreto a utilizar; del mismo modo que al iniciar un desarrollo de software se decide el paradigma metodológico (estructurado, orientado a objetos, etc.) y la metodología concreta a seguir (XP, METRICA, RUP, etc.). La retroalimentación entre la etapa de resolución y verificación, y la definición del método de trabajo (señalada con trazo continuo en la Figura 32) muestra cómo el método se va haciendo y refinando a medida que se avanza en la

solución del problema. Así, se puede decir que la definición del método de trabajo no concluye hasta que se finaliza la fase de resolución y verificación. La fase de resolución mostrará cómo resolver el problema y, al igual que ocurre con la fase de definición del método de investigación, el método definido en ella, se irá haciendo y refinando en aproximaciones sucesivas. De igual modo, cuando se lleva a cabo un desarrollo software, el producto no está totalmente terminado hasta que se realizan las últimas pruebas y, generalmente, se modifica el software de manera continua.

#### Resolución, validación y verificación

Para la resolución, validación y verificación del problema se podrían emplear diferentes métodos de investigación. La fase de resolución podría corresponder a la creación de la hipótesis en un método empírico o la elaboración de la teoría en un método deductivo. Visto de este modo, la verificación del método empírico equivaldría a la validación del método utilizado para la resolución del problema, mientras que la verificación de la teoría en un método deductivo equivaldría a la comprobación de la consistencia en dicho método, es decir a la tarea de verificación propiamente dicha.

#### Análisis de resultados y elaboración de conclusiones

Se trata de contrastar la hipótesis planteada al comienzo de la investigación con los resultados obtenidos. Se debe comprobar hasta qué punto se cumplieron los objetivos y en qué medida se ha resuelto el problema. En esta fase es muy importante delimitar los aspectos que no se han podido resolver y otros nuevos problemas que hayan surgido como consecuencia de la investigación y que pasarán a ser puntos de partida de nuevas

investigaciones. En métodos de naturaleza cuantitativa, parece que este aspecto es claro. Cuando la investigación es de naturaleza cualitativa y la hipótesis formulada no responde a una expresión de causa efecto, contrastar de la hipótesis consiste básicamente en comprobar hasta qué punto se han cumplido los requisitos impuestos al principio de la investigación. Equivaldría, en un desarrollo software, a contrastar con el usuario hasta qué punto se han cumplido sus expectativas y analizar si es posible realizar alguna mejora, aunque incluso no hubiera sido tomada en cuenta al inicio del proyecto.

### **Metodología para el análisis y revisión de la literatura**

Toda investigación emprendida debe considerar una revisión de la literatura, aunque este no sea el objetivo final sino sólo una manera de imponerse acerca del estado del arte del tema que se desea abordar. En las disciplinas de computación, el desarrollo de la investigación, por ejemplo en IS, tiene una trayectoria reciente, lo que las transforma en un campo sin demasiados antecedentes históricos al respecto. No obstante, existe un esfuerzo por mejorar dicha situación. La aplicación de diversos métodos de investigación entre las que se incluyen análisis conceptual, casos de estudio, análisis de datos, experimentación de campo, experimentación de laboratorio y simulación (Glass y cols., citado por Caro y cols., 2005) da cuenta de ello.

Caro y cols. (2005) argumentan que no existen metodologías que guíen el desarrollo de revisiones sistemáticas de la literatura en las disciplinas de la computación y, consecuentemente con esto, Kitchenham (citado por Caro y cols., 2005) propone un método para realizar esta tarea. Una revisión sistemática se define como una manera de evaluar e interpretar toda la investigación relevante disponible, respecto de una interrogante de investigación particular, en un área temática o fenómeno de interés. Los

estudios individuales que contribuyen a una revisión sistemática se denominan estudios primarios; una revisión sistemática se considera un estudio secundario.

En el caso particular de un proyecto de fin de carrera, Caro y cols. (2005) consideran que “el trabajo será realizado por un solo investigador-alumno (a lo más dos), en que la supervisión está a cargo de un tutor o guía y que tiene un tiempo límite para su realización, estipulado en el plan de estudios de cada caso en particular”. Estas condiciones han sido consideradas en la propuesta que realizan y que modifica la propuesta original, manteniendo la esencia de ésta. En particular este método propone tres etapas fundamentales que son: planificación de la revisión, desarrollo de la revisión y publicación de los resultados de la revisión, las que a su vez se encuentran divididas en otras etapas que detallan la forma en que se deben desarrollar. A continuación, se muestra la propuesta en la Figura 33 y seguidamente se explica en qué consiste cada etapa.

<b>Etapa 1: Planificación de la revisión</b>
Identificación de la necesidad de revisión
Definición de un protocolo de búsqueda
Definición de un protocolo de revisión
Evaluación de la planificación
<b>Etapa 2: Desarrollo de la revisión</b>
Búsqueda de resultados primarios
Selección de estudios primarios
Extracción y gestión de datos
Síntesis de datos
<b>Etapa 3: Publicación de los resultados</b>

Figura 33. Método de Caro y cols. (2005)

### Planificación de la revisión

Esta etapa tiene como propósito específico definir los parámetros más importantes que serán tomados en cuenta cuando se lleve a cabo la revisión. Se debe establecer las razones que justifican llevarla a cabo, la manera en que se hará la búsqueda de trabajos y la forma en que éstos serán revisados, finalmente, se evaluará la planificación realizada. Comprende las siguientes sub-etapas: identificación de la necesidad de revisión, definición de un protocolo de búsqueda, definición de un protocolo de revisión y evaluación de la planificación.

### Desarrollo de la revisión

En esta etapa se lleva a cabo la revisión propiamente tal. Su desarrollo está guiado por la planificación de la revisión. Sin embargo, y ya que es un proceso flexible, es posible incluir cambios que mejoren su desempeño. Las sub-etapas que contempla el desarrollo de la revisión son: búsqueda de resultados primarios, selección de estudios primarios, extracción y gestión de datos y síntesis de datos.

### Publicación de los resultados

Esta etapa corresponde a la utilización de los resultados una vez que disponemos de ellos. Es muy importante la difusión de los resultados obtenidos producto de una revisión sistemática. En este caso, junto con la obtención del documento exigido en cada currículo [programa de estudios] en particular, sería conveniente comunicar los resultados a través de la participación en conferencias, publicación de un artículo o de un informe técnico.

## **Estudio de caso**

El estudio de casos es, según Sabino (1978), un tipo de “la gran variedad de métodos posibles” de diseños de campo estándares más frecuentes. La nota peculiar de este diseño de investigación la constituye el estudio profundizado y exhaustivo de uno o muy pocos objetos de investigación, lo que permite obtener un conocimiento amplio y detallado del mismo, casi imposible de alcanzar mediante los otros diseños considerados por este autor. Su ventaja principal estriba en su relativa simplicidad y en la economía que supone, ya que puede ser realizado por un investigador individual o por un grupo pequeño, y porque no requiere de técnicas masivas de recolección [de información] como las encuestas y otros métodos.

Sin embargo, desde el punto de vista de Hernández y cols. (2003) “el estudio de caso no es una elección de método, sino del “objeto” o la “muestra” que se va a estudiar”. El caso es la unidad básica de la investigación y puede tratarse de una persona, una pareja, una familia, un objeto, un sistema, una organización, una comunidad, un municipio, un departamento o estado, una nación, etc. El estudio de caso es tanto de corte cuantitativo como de corte cualitativo o incluso mixto (cuantitativo-cualitativo) y se realizan bajo cualquier diseño de investigación.

En este trabajo, basado en la perspectiva de los autores citados, se considera el UP como caso de estudio que permitirá al investigador obtener el conocimiento suficiente sobre el SPEM como objeto de estudio para sistematizar la experiencia de su implementación, por considerarse una manera práctica de lograr el objetivo propuesto.

## **La creatividad como método de investigación**

La creatividad, denominada también inventiva, pensamiento original, imaginación constructiva, pensamiento divergente, pensamiento creativo, es la generación de nuevas ideas o conceptos, o de nuevas asociaciones entre ideas y conceptos conocidos, que habitualmente producen soluciones originales. Algunos sentidos del concepto en la actualidad son:

- Acto de inventar cualquier cosa nueva (ingenio).
- Capacidad de encontrar soluciones originales.
- Voluntad de modificar o transformar el mundo.
- La redisposición en orden de experiencias del pasado.

Esta última definición es probablemente, según Bocchino (1990), la más interesante y específica. No obstante, si se considera a la creatividad con los términos concretos de la redisposición de experiencias pasadas, ¿cómo puede obtenerse una base rica y variada de experiencias para tener la materia prima para la creatividad? Las investigaciones indican que eso se hace frecuentemente por medio de:

- El descubrimiento de las ideas interesantes de otros, mediante la lectura, los viajes, conferencias y conversaciones.
- Experiencias personales en actividades profesionales.
- Educación formal para proporcionar una ampliación organizada y sistematizada de las experiencias.
- Investigaciones en el campo por mediación de fuentes profesionales, conferencias, literatura y experimentación.

De esta manera, el desarrollo de este trabajo estuvo cargado de un fuerte componente creativo, reflejado en gran parte, en la disposición y manejo de recursos, incorporación y configuración de diferentes métodos de

investigación científica y técnicas de IS y la configuración del entorno de trabajo adecuado para llevar a cabo la investigación misma.

### **Justificación del uso del Proceso Unificado como caso de estudio**

El Proceso Unificado de Desarrollo Software o simplemente Proceso Unificado es un marco de desarrollo de software que se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental. En palabras de sus creadores el UP “no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos”.

Debido al carácter relativamente investigador de este Trabajo de Grado, y a la necesidad de modificar los requisitos que surgirían según se fueran evaluando y probando las distintas posibilidades para desarrollarlo, se ha optado por el UP propuesto por Rumbaugh, Booch y Jacobson por su reconocida reputación en el campo de la IS, la variedad de instanciaciones que han sido desarrolladas partiendo de las bases fundamentales propuestas en el marco de trabajo original y fundamentalmente porque está descrito en lenguaje natural en libros de texto; lo que lo hace un candidato ideal como caso de estudio para lograr el objetivo que se plantea en este Trabajo de Grado.

## **CAPÍTULO III. DESARROLLO**

En este capítulo se presentan de manera general las distintas fases por las que transitó el proceso de investigación, desde la creación del proyecto de investigación previo, que supone un proceso de revisión de fuentes documentales relativas al tema abordado y el estudio de paradigmas y métodos de investigación, hasta la redacción de este informe. Se describe la manera en que encajan cada uno de los métodos de investigación abordados con la estrategia general adoptada.

### **BÚSQUEDA DE LA DOCUMENTACIÓN**

La búsqueda de documentación estuvo caracterizada por la presencia de nuevos conceptos por asimilar y la gran cantidad de literatura a revisar por lo que se optó por la incorporación de un método de apoyo pensado para aquellos que, como el autor de este trabajo, carecen de resultados de investigaciones previas y necesitan realizar un esfuerzo importante para comenzar a encauzar su investigación mediante un análisis de la bibliografía. En consecuencia, las actividades correspondientes se llevaron a cabo desde el inicio de la investigación, siendo su ejecución más intensa en la fase de determinación del problema y la fase de resolución, verificación y validación. Se revisaron y analizaron diversas fuentes de información bibliográfica, además de realizar la búsqueda de herramientas de software que dieran soporte a la notación SPEM y de información acerca del método para la resolución del problema planteado y del mismo método que da apoyo a esta fase.

Para su cumplimiento, dada la importancia de los elementos conceptuales en

el desarrollo de la investigación, el proceso se apoyó en el método de Caro y cols. (2005) para el análisis y revisión de literatura en el contexto de proyectos de fin de carrera. Las etapas del método se renombraron para efectos de esta investigación como actividades. A continuación, se presentan los resultados de cada una de las actividades realizadas.

### **Planificación la revisión**

Esta actividad se inició con el propósito de definir los parámetros más importantes que serían tomados en cuenta para el desarrollo de la revisión. Se identificó la necesidad de emprender una revisión sistemática, con el objetivo de sintetizar la información concerniente a los principales elementos conceptuales que constituyen el universo del discurso del tema abordado, para proveer un marco de trabajo y/o los antecedentes necesarios. Para esta actividad se realizaron las siguientes tareas: identificación de la necesidad de revisión, definición de un protocolo de búsqueda, definición de un protocolo de revisión y finalmente, la evaluación de la revisión.

#### Identificación de la necesidad de la revisión

Al momento de iniciar la investigación no existía alguna revisión sistemática previa en el Programa de la Licenciatura en Informática del Núcleo de Sucre de la UDO acerca del fenómeno de interés de este trabajo; es decir, no habían antecedentes de trabajos realizados en la línea de investigación concerniente a este trabajo particular por lo que la necesidad de recopilar información era imperante.

Junto con las razones antes mencionadas, que justificaron la necesidad de la revisión emprendida, se identificaron los recursos con que inicialmente se

contó para llevarla a cabo, presentados en la Tabla 1.

Tabla 1. Identificación de la necesidad de revisión.

<b>Objetivo</b>
Resumir la evidencia existente en relación a cómo se ha abordado el tema de la especificación de PDS, los lenguajes y notaciones empleadas para tal fin así como las herramientas de soporte. Además de todos los elementos conceptuales alrededor del SPEM.
<b>Interrogantes</b>
Se consideraron las siguientes interrogantes de investigación: <ul style="list-style-type: none"><li>• ¿Qué es la especificación de PDS?</li><li>• ¿Existen lenguajes notacionales para este dominio?</li><li>• ¿Hay herramientas disponibles para soportar las notaciones empleadas?</li><li>• ¿Es SPEM una notación estándar?</li><li>• ¿Se dispone de herramientas software que implementen el SPEM?</li></ul>
<b>Recursos</b>
Libros de IS. Autores: Roger Pressman y Ian Sommerville.  Internet, a través del uso de motores de búsqueda, revistas electrónicas y bibliotecas digitales: <ul style="list-style-type: none"><li>• <i>Google Scholar</i> versión beta en <a href="http://scholar.google.com/">http://scholar.google.com/</a></li><li>• DOCIS: <i>Documents in Computing and Information Science</i> en <a href="http://wotan.liu.edu.docis/">http://wotan.liu.edu.docis/</a></li><li>• CiteSeerX versión beta: <i>Scientific Literature Digital Library and Search Engine</i> en <a href="http://citeseerx.ist.psu.edu/">http://citeseerx.ist.psu.edu/</a></li><li>• The ACM Digital Library en: <a href="http://portal.acm.org/portal.cfm">http://portal.acm.org/portal.cfm</a></li><li>• Novatica en: <a href="http://novatica.ati.es/">http://novatica.ati.es/</a>.</li></ul> Actas de conferencias internacionales.  Sitios web de organismos de normalización (OMG, ISO, IEEE, entre otros) que provean estudios primarios.

### Definición de un protocolo de búsqueda

Una vez identificada la necesidad de revisión, establecido su objetivo e identificado los recursos para llevarla a cabo, se definió un protocolo de búsqueda para precisar las normas que seguiría la investigación respecto al proceso de búsqueda en las fuentes de información definidas anteriormente.

Dicho protocolo consistió en establecer los términos claves de las búsquedas, las combinaciones de éstos, la estrategia de búsqueda según cada fuente y la manera en que se registraron los resultados; refinándose cada uno en la medida que se avanzaba en la investigación.

El protocolo de búsqueda establecido se encuentra descrito en la Tabla 2. Las palabras del renglón “Términos” fueron derivadas principalmente del objetivo general del proyecto de investigación que gira alrededor de: especificación de procesos y SPEM, la identificación de las alternativas ortográficas y sinónimos para la mayoría de los términos, uso del OR booleano para incorporar las alternativas ortográficas y sinónimos y el uso del AND booleano para vincular los términos.

En la medida en que avanzó, el proceso de revisión general sufrió sucesivas refinaciones en las que se incluyeron nuevos términos que se consideraron pertinentes, hasta obtener el producto de trabajo presentado en la Tabla 2. Para registrar los resultados se estableció un sistema de catalogación caracterizado por la clasificación del tipo de documento obtenido (formato HTML y PDF, principalmente) y el tema o asunto tratado (de acuerdo al nivel de interés).

Tabla 2. Protocolo de búsqueda.

<b>Protocolo de búsqueda</b>	
Términos	Procesos, Modelos, Métodos, Software, Especificación, Notación, Lenguajes, SPEM.
Combinaciones	<p>“Modelo de procesos de software”, “Software process model”.</p> <p>“Proceso de software” OR “Proceso de desarrollo de software”, “Software process” OR “Software development process”.</p> <p>(Especificación OR modelado OR definición OR descripción) AND (procesos de software OR procesos de desarrollo de software).</p> <p>“Especificación” OR “Modelado” AND “Procesos de software” AND SPEM.</p>
Estrategias de búsqueda	<p>Libros: revisión manual de textos reconocidos.</p> <p>Internet: acceso a documentos mediante motores de búsqueda, ingresando los términos y/o combinaciones de ellos. Cuando los documentos no sean accesibles, buscar en las páginas personales de los autores o en sitios de publicación alternativos.</p> <p>Autores: identificar a los autores relevantes; acceder directamente a sus páginas personales para la búsqueda de material.</p> <p>Artículos: detectar referencias bibliográficas de utilidad; en base a ellas buscar directamente el documento citado usando los antecedentes que aparecen en la referencia (autor, título, conferencia, etc.)</p>
Registro de los resultados	Registrar los resultados de búsqueda mediante tablas.

## Definición de un protocolo de revisión

Los resultados obtenidos por medio del protocolo de búsqueda fueron sometidos a revisión parcial, bajo un protocolo de revisión que especifica los métodos empleados para emprender la revisión sistemática.

Dado que un alto número de los estudios revisados estaban escritos en formato de artículo científico se siguieron las normas tomadas en cuenta cuando se trata de estudios en formato de artículo científico; una breve lectura del resumen y de la introducción de dichos artículos bastó para decidir la inclusión del estudio en los documentos a revisar detalladamente o en los documentos para una revisión posterior; almacenándolos en directorios para documentos “interesantes” o “no interesantes” en el repositorio de trabajo creado en el disco duro del computador asignado para tal fin. En la Tabla 3 se muestra el protocolo de revisión establecido.

Tabla 3. Protocolo de revisión

<b>Protocolo de revisión</b>	
Normas de revisión	<p>Cada estudio encontrado será revisado de acuerdo al criterio del investigador y registrado según se considere un documento “relevante” o un documento para “cuarentena”.</p> <p>Si no se dispone del trabajo completo, conseguirlo para asegurar su relevancia para la revisión.</p> <p>Se debe leer al menos el resumen e introducción y con ello se decidirá la inclusión o exclusión del estudio, basándose en los criterios de este protocolo.</p> <p>Se sugiere registrar comentarios acerca del estudio, que en una etapa posterior permita, por ejemplo, recordar su relevancia para la investigación, el motivo de su exclusión, o cualquier otro antecedente que pueda ser útil.</p>
Criterios de inclusión	<p>Se incluirán todos aquellos trabajos o estudios que aborden el tema de la especificación de PDS con SPEM u otro PML y las herramientas de soporte.</p>

Tabla 3. Protocolo de revisión (Continuación)

<b>Protocolo de revisión</b>	
Criterios de exclusión	Se excluirán aquellos estudios que a pesar de contener los términos de búsqueda o combinaciones de ellos, no contengan información relevante sobre el tema y/o no abordan tópicos de interés para la investigación.
Estrategia de extracción de datos	<p>Cada estudio seleccionado, debe ser leído con el objeto de extraer datos para este trabajo, considerando dos (2) etapas:</p> <p>Etapa 1: Se deberá leer las secciones del resumen, introducción, trabajos relacionados, conclusión y referencias. Con ello será posible obtener la siguiente información:</p> <ul style="list-style-type: none"> <li>• La comunidad a la que está orientado el artículo. [Introducción, Trabajos relacionados, Referencias]</li> <li>• Las principales contribuciones (según los autores). [Resumen, Introducción, Conclusión]</li> <li>• Posibles consecuencias de las contribuciones (Aplicaciones directas, nuevas técnicas, nuevas áreas de investigación, etc.). [Introducción]</li> </ul> <p>Etapa 2: Se leerán los preliminares y el cuerpo del artículo, lo que permitirá:</p> <ul style="list-style-type: none"> <li>• Incluir en forma detallada la información que es necesaria para la revisión.</li> <li>• Comprender un experimento o los fundamentos de un marco de trabajo, identificar las características de un modelo, etc.</li> </ul>
Estrategia de síntesis de datos	<p>Los datos serán sintetizados de acuerdo a los siguientes temas:</p> <ul style="list-style-type: none"> <li>• Procesos de desarrollo de software.</li> <li>• Lenguajes de modelado de procesos de desarrollo de software.</li> <li>• Uso de SPEM en el modelado y/o especificación de PDS.</li> </ul>

## Evaluación de la planificación

La evaluación de la planificación fue realizada por el asesor de este trabajo, quien determinó la relevancia de la información obtenida.

## Desarrollo de la revisión

El desarrollo de la revisión consistió en la exploración de los resultados de la búsqueda como tal. Se revisó cada uno de los resultados catalogados para hallar estudios primarios potencialmente útiles, manteniéndolos accesibles, en formato electrónico y/o físico con intención de realizar la selección de los

estudios primarios, guiado por criterios de inclusión y exclusión inherentes a la investigación. Para su cumplimiento, esta actividad consistió en las tareas de búsqueda y selección de estudios primarios y extracción y síntesis de datos.

#### Búsqueda de estudios primarios

La búsqueda de estudios primarios se realizó en base al protocolo de búsqueda definido y sobre los recursos de información disponibles, mejorando a su vez el protocolo en la medida que avanzó la revisión según se iban reconociendo algunos términos empleados como sinónimos o alternativas ortográficas de los inicialmente planteados como términos de búsqueda.

#### Selección de estudios primarios

La selección de los estudios primarios se hizo en base al protocolo de revisión definido, manteniéndose su ejecución a lo largo de la investigación. Con los estudios seleccionados se procedió a la extracción de datos y en los no seleccionados se registró la información relativa al motivo del rechazo. Este registro se hizo directamente en el documento impreso.

#### Extracción de datos

La extracción de los datos se realizó de manera continua a lo largo de la investigación; extrayendo los datos en la medida que eran necesarios para desarrollar los resultados de la revisión sistemática. Se registraron en base a un formato de año-nombre del trabajo-interés.

## Síntesis de datos

La síntesis de los datos corresponde con los resultados bibliográficos distribuidos entre el marco teórico y el marco metodológico de este trabajo de investigación, abordado desde los siguientes temas de interés: Ingeniería del software, Procesos de ingeniería del software, Procesos de desarrollo de software, Modelos de procesos de software, Modelado de procesos de software y soporte y SPEM.

Cada uno de estos temas se introdujo en el marco teórico y el marco metodológico permitiendo ubicar de esta manera el trabajo en un contexto teórico concreto así como presentar los antecedentes que la preceden.

## **Publicación de los resultados**

Esta actividad consistió en la publicación de los resultados de la revisión sistemática, los cuales fueron utilizados para el desarrollo del estado del arte de la investigación que se presentan enmarcados dentro de los marcos teórico y metodológico del Capítulo II de este informe de Trabajo de Grado.

## **DETERMINACIÓN DEL PROBLEMA**

Esta etapa permitió refinar el problema que se enuncia en el apartado destinado para tal fin en este informe de Trabajo de Grado a través de la creación de un modelo de dominio aprovechando la notación UML (Jacobson y cols., 2000) para diagramas de clases que permitió capturar, describir y comprender los elementos más importantes en el contexto del problema, así como establecer un vocabulario y terminología común al producir algunas convenciones o acuerdos sobre la terminología inmersa en el desarrollo del

trabajo para compartir el conocimiento asociado y delimitar los aspectos concretos de la investigación.

El modelo de dominio se elaboró utilizando el *plug-in* UML2 para Eclipse 3.3. La Figura 34 muestra el modelo de dominio producto de sucesivas refinaciones, según se iba avanzando en la comprensión del tema de investigación como resultado de las actividades de la etapa de búsqueda de la documentación, de manera similar a como sucede en un desarrollo de software iterativo e incremental.

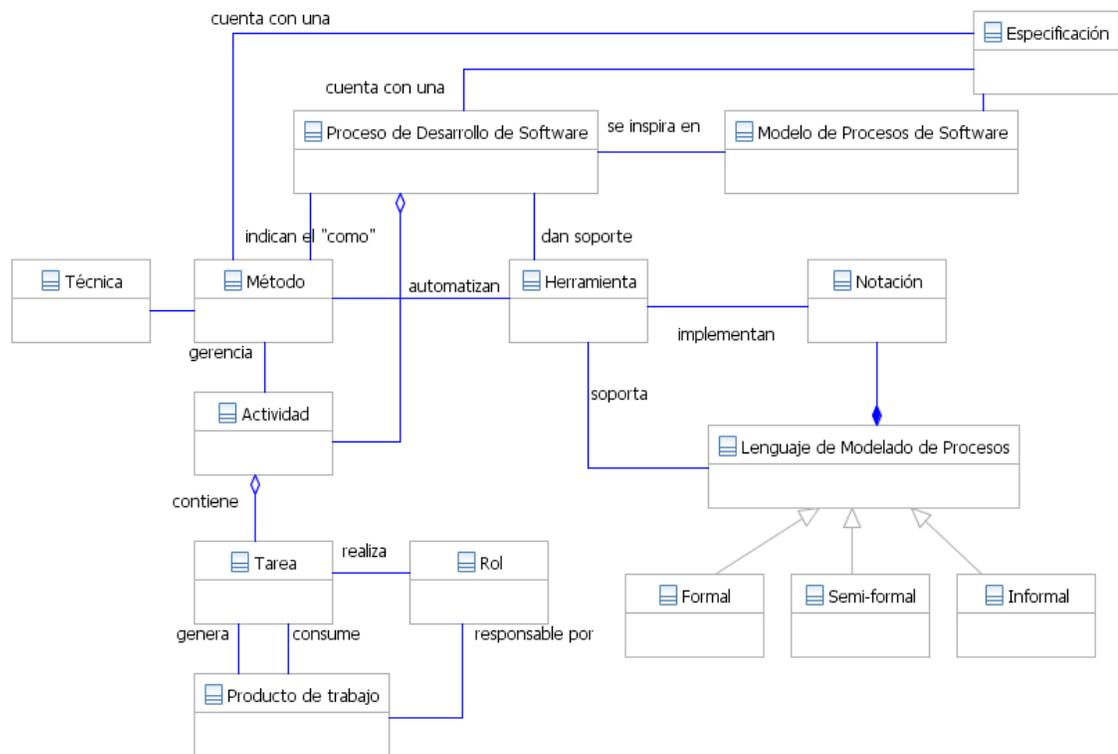


Figura 34. Modelo de dominio para la determinación del problema

En el modelo de dominio desarrollado se presenta al PDS como una entidad compuesta de un conjunto de Actividades que describen las Tareas

necesarias para obtener (salidas) y/o consumir (recibir como entradas) los Productos de trabajo sobre los que un determinado Rol tiene responsabilidad. Las Actividades del proceso son gestionadas por los Métodos que aplican diferentes Técnicas para lograr los objetivos del Método y del proceso general a través del soporte automatizado o semi-automatizado de las Herramientas de software, las cuales implementan la Notación de un Lenguaje de Modelado de Procesos a través del que se puede describir de manera formal, semi-formal o informal Técnicas, Métodos y Actividades del Proceso para obtener así una Especificación que permita entender claramente lo que se pretende (objetivos) a través de la aplicación de los mismos.

## **CREACIÓN DE LA HIPÓTESIS**

La hipótesis se formuló como la descripción del nuevo objeto a construir que, en este caso corresponde a la especificación en lenguaje natural de los requisitos y características deseables para caso de estudio descrito en la notación del SPEM.

Se espera que la especificación en la notación del SPEM sea capaz de describir el caso de estudio identificando su nombre, versión (es), autor (es), fecha de creación, la cual debe poder guardar trazabilidad con versiones anteriores, identificando mejoras y los responsables de las mismas, así como una descripción del propósito, los principios y lineamientos en que se basa. Se debe especificar utilizando un lenguaje de modelado visual, que posea sintaxis y semántica estándar a fin de poder ser utilizado por distintas personas del área. Debe ser claro el (los) camino (s) a seguir, es decir el “cómo” llevar a cabo cada una de las actividades y los responsables de su ejecución, inclusive poder sugerir plantillas de ser necesario. La

especificación del caso de estudio constituirá una guía que lleve de la mano al lector partiendo de lo general a lo particular. Debe poseer la característica de poder navegar entre las actividades, artefactos, responsables, ente otros elementos del mismo. De la misma manera se debe poder incluir descripciones detalladas (contenido) sobre cada elemento del caso de estudio que presenten información relevante sobre los mismos al lector.

## **DEFINICIÓN DEL MÉTODO DE TRABAJO**

Para la resolución, validación y verificación, se planteó obtener dominio conceptual y tecnológico del objeto de estudio a través de un proceso de creatividad, a fin de comprender las ramificaciones técnicas entre la teoría y la aplicación tecnológica, incorporando un caso de estudio que permitiera desarrollar la experiencia y comprobar así los resultados.

El método de trabajo consistió en la revisión y comprensión de la especificación del SPEM desarrollada por el OMG, para caracterizar sus elementos y conceptos de modelado; la selección del caso de estudio, su revisión y comprensión; selección de una herramienta de soporte y configuración del ambiente de trabajo; la especificación del caso de estudio y realización de pruebas para la validación de la hipótesis.

La Figura 35 muestra un esquema gráfico o modelo de cada una de las actividades establecidas en el método de trabajo definido.

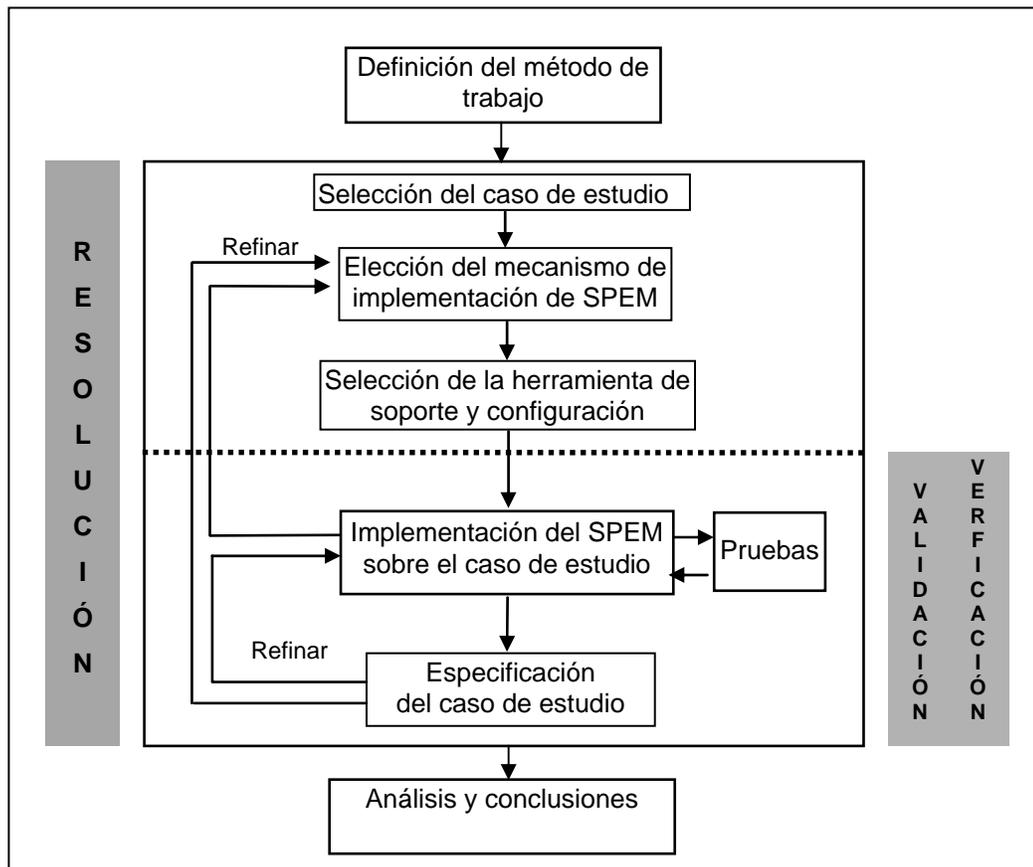


Figura 35. Método para la resolución del problema

## RESOLUCIÓN, VERIFICACIÓN Y VALIDACIÓN

En esta etapa se elaboró un método para la solución del problema. Se puede observar que éste se acerca más a cualquiera de los métodos tradicionalmente utilizados en la IS (refinamientos sucesivos) que a los métodos de investigación científica (la fase de resolución y validación equivale a las fases de creación y realización del experimento en el método experimental). En este caso concreto, se utilizó caso de estudio e investigación en acción, permitiendo definir y refinar las relaciones de los elementos involucrados en la especificación en la medida que ésta se fue configurando. De manera general, se llevaron a cabo las siguientes

actividades:

### **Revisión y comprensión de la especificación del SPEM**

Esta actividad consistió en el estudio de los principales elementos conceptuales presentes en la especificación del SPEM proporcionada por el OMG en su sitio web. De la lectura de este documento, así como de otros trabajos relacionados, se obtuvo la comprensión del meta-modelo y la formalización de sus elementos teóricos presentes en este Trabajo de Grado que se pueden ubicar como parte del marco teórico que lo comprende. Así mismo, fue necesario el descubrimiento de las ideas interesantes de autores e investigadores de IS mediante el análisis de los resultados de la revisión sistemática realizada en la búsqueda de documentación y a través de conversaciones y contacto mediante correo electrónico con investigadores del área.

### **Selección del caso de estudio**

Esta actividad consistió en una revisión general de modelos y propuestas existentes para el desarrollo de software que permitiera seleccionar un proceso de desarrollo entre la variedad existente en la actualidad. Finalmente, el caso de estudio elegido fue el UP.

### **Elección del mecanismo de implementación de SPEM**

Se decidió implementar el SPEM a través del mecanismo de meta-modelado. Para ello, se debió utilizar una herramienta que soportara este mecanismo para obtener una especificación en la que mediante hipervínculos se pudiera navegar a través de todo el proceso de manera natural, que proporcionara

una descripción general del mismo así como su propósito, principios, lineamientos, productos de trabajo consumidos y generados en cada actividad, roles desempeñados por los participantes, entre otros, y se debía permitir documentar cada uno de los elementos de modelado utilizados e incorporar plantillas para guiar el desarrollo de productos de trabajo que así lo requieran.

### **Selección de herramientas y configuración del ambiente de trabajo**

Una vez entendidos los elementos conceptuales propios del SPEM y establecido el mecanismo de implementación, la siguiente actividad consistió en la búsqueda y elección de una herramienta de software que diera soporte al SPEM para poder desarrollar en ella la especificación del caso de estudio y la configuración del entorno de trabajo. Las actividades de la fase de búsqueda de la documentación fueron de vital importancia en esta fase de la investigación, que como ya se ha mencionado se realizaron de manera solapada y no siguiendo un enfoque lineal-secuencial.

La herramienta software elegida fue *EPF Composer*, desarrollada sobre la plataforma de código abierto *Eclipse*; empleando la versión 1.2, tanto en inglés como en español; haciendo uso del paquete *NLPack-epf-composer-1.2.0* que permite traducir la herramienta al español.

*EPF Composer* es una herramienta que sirve para editar fragmentos de método, procesos y/o metodologías y generar automáticamente la documentación adecuada en formato para la web. Dichos fragmentos se almacenan en formato XMI (*XML Meta-data Interchange*) que al estar basados en el estándar SPEM 2 pueden ser reutilizados por cada vez más herramientas CASE (*Computer Aided Software Engineering*).

La elección de *EPF Composer* obedeció principalmente a que implementa SPEM bajo el mecanismo de meta-modelado que permitió disponer de la sintaxis y semántica para especificar, de manera clara y precisa, cada uno de los elementos constitutivos del caso de estudio. Además de lo anterior, con el uso de *EPF Composer* no existen gastos derivados de la inversión que representa el costo de adquisición de una herramienta de software.

La Figura 36 muestra un esquema del entorno de trabajo. En ella se puede observar que la herramienta se ejecuta sobre el entorno de ejecución Java (JRE, *Java Runtime Environment*) que debe ser instalado previamente en el computador en el cual funcionará el *EPF Composer*.

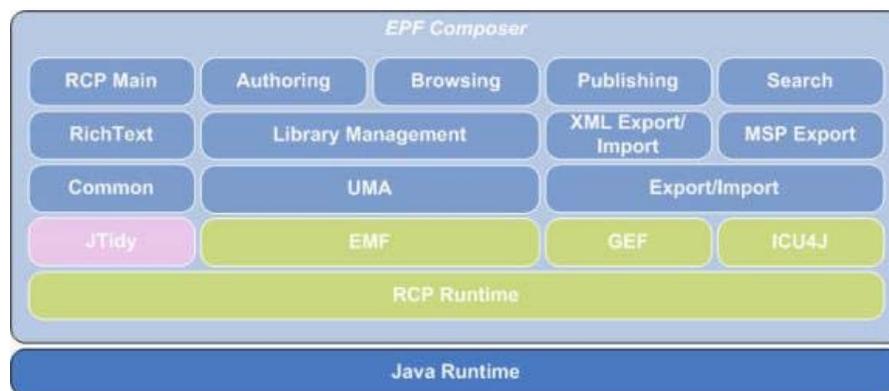


Figura 36. Ambiente de trabajo.

Otra de las herramientas utilizadas para probar la notación del SPEM fue *Enterprise Architect* (EA). La versión utilizada para este trabajo fue la 6.5. EA tiene un mecanismo de perfil UML genérico para cargar y trabajar con diferentes perfiles. Los perfiles UML proveen un mecanismo de extensión genérico para construir modelos UML en dominios particulares. Están basados en estereotipos y valores etiquetados adicionales que se aplican a

elementos, atributos, métodos, vínculos y más. Un perfil es una colección de tales extensiones que describen conjuntamente algún problema de modelado en particular y facilitan construcciones de modelado en ese dominio.

Los perfiles UML para EA se especifican en archivos XML, con un formato específico. Estos archivos se pueden importar en EA (ver Anexo B) en la página de Recursos (*Resource page*) del navegador de proyecto. Una vez importados, se pueden arrastrar y soltar los elementos del perfil en el diagrama que se esté trabajando. EA agregará el estereotipo, los valores etiquetados y los valores por defecto, las notas y además un meta-archivo, si es que se especificó uno, al nuevo elemento. También se pueden arrastrar y soltar atributos y operaciones en las clases existentes y tenerlas inmediatamente extendidas con el estereotipo, los valores, entre otros. A través del perfil UML para SPEM v1.0 (beta) se define un conjunto de estereotipos para definir procesos y sus componentes usando la especificación SPEM. La Figura 37 muestra la página de recursos (a la derecha) del EA, en la que se puede visualizar los iconos que la herramienta pone a disposición a través del mencionado perfil UML para SPEM.

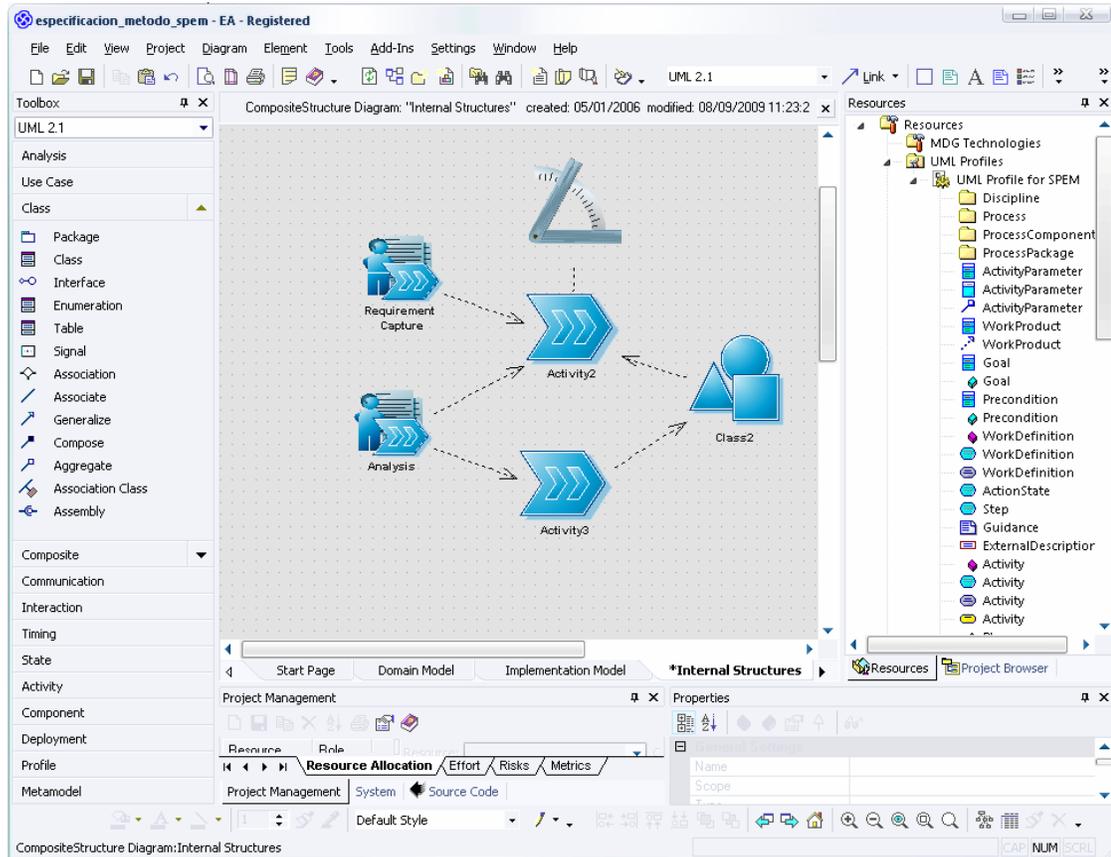


Figura 37. UML Profile for SPEM para Enterprise Architect 6.5.

## Implementación del SPEM sobre el caso de estudio

La implementación del SPEM sobre el caso de estudio permitió reducir las ambigüedades presentes en la descripción en lenguaje natural del UP al aplicar la notación estándar que proporciona el lenguaje, enriqueciéndolo y verificando su corrección.

Para esta actividad se tomaron en cuenta cada uno de los aspectos que presenta el UP, tales como los Flujos de trabajo fundamentales, los Trabajadores y Actividades y las Fases del ciclo de vida para ser implementados en SPEM a través de la herramienta *EPF Composer*.

Finalmente el producto resultante es la especificación en SPEM del UP. En el Capítulo IV se describe detalladamente la manera en que se desarrolló esta actividad.

### **Realizar pruebas**

La ejecución de pruebas consistió en verificar que cada uno de los elementos y aspectos del proceso implementados en SPEM estaban fielmente representados y que las relaciones entre ellos fueran correctas, así mismo verificar que no se obviaron elementos y realizar revisiones ortográficas sobre las descripciones textuales de cada uno. Finalmente se realizaron pruebas de navegación a través de la perspectiva de navegación (*Browsing*) de *EPF Composer* para corregir enlaces rotos o vínculos incorrectos entre los elementos del Paquete de contenido establecidos para el UP.

### **ANÁLISIS DE RESULTADOS Y ELABORACIÓN DE CONCLUSIONES**

En esta etapa se contrastó la hipótesis planteada con los resultados obtenidos, comprobando hasta qué punto se cumplieron los requisitos impuestos a la especificación del caso de estudio y en qué medida se resolvió el problema que se planteó al principio de la investigación. Los resultados de esta fase están concretados en el apartado: Resultados y discusión, de este informe de Trabajo de Grado.

## **CAPÍTULO IV. ESPECIFICACIÓN DEL CASO DE ESTUDIO**

Como ya se ha escrito, el caso de estudio elegido para implementar el SPEM es el UP. En este apartado se introducen las reglas establecidas para representar cada aspecto del UP con *EPF Composer*.

### **ASPECTOS GENERALES**

En esta parte de la especificación se establecieron las reglas sobre los conceptos y aspectos del UP descritos en lenguaje natural en Jacobson y cols. (2000) para establecer la organización del contenido de método (*Method Content*) y del proceso (*Process*). Se contempló la Parte II del libro que introduce cada uno de los flujos de trabajo fundamentales (disciplinas en la terminología del SPEM) en capítulos separados: los Capítulos 6 y 7 tratan los requisitos; el Capítulo 8, el análisis; el Capítulo 9, el diseño; el Capítulo 10, la implementación; y el Capítulo 11, la prueba. Las reglas son las siguientes.

#### **Disciplinas del proceso**

Una disciplina representa una colección de tareas relacionadas para definir un "área de interés" mayor y en la IS, las disciplinas incluyen: Requisitos, Análisis, Diseño, Implementación y Prueba. En el UP se denominan "Flujos de trabajo fundamentales". Para establecer este aspecto se consideraron cada una de las disciplinas del UP (Requisitos, Análisis, Diseño, Implementación y Pruebas), las cuales se encapsularon y distribuyeron en tres (3) paquetes de contenido (*Content Package*) dentro del contenido de método (*Method Content*). Cada uno de estos paquetes de contenido

representa las actividades generales comunes a cualquier tipo de PDS (Especificación del software, Diseño e Implementación del software y Validación y Evolución del software) según Pressman (2002). De esta manera, la estructura interna de cada paquete de contenido se organiza en sucesivos paquetes (*Packages*); uno por cada disciplina, de la siguiente manera.

- Especificación del software
  - Requisitos
  - Análisis
- Diseño e implementación del software
  - Diseño
  - Implementación
- Validación y evolución del software
  - Pruebas

Adicionalmente, se creó un paquete “comun” cuyo contenido se explica en los apartados siguientes. Por consiguiente, el contenido de método del UP aquí establecido tiene una estructura de paquetes basada en la jerarquía proceso-subproceso-actividad-tarea. La Figura 38 muestra la estructura del Contenido de método resultante.

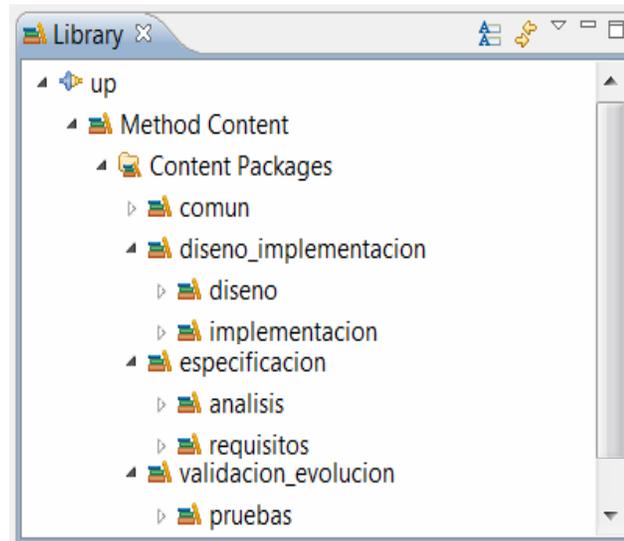


Figura 38. Estructura del Contenido de método para el UP.

### **Roles en el proceso**

En el UP se denominan “Trabajadores” a los distintos roles que puede desempeñar un persona en un proyecto de desarrollo de software. En SPEM a éstos se les representa como un *Role*.

Los roles son generales y comunes a todo el UP, por tanto se definen en el paquete “común” aquellos roles que participen en más de una disciplina del proceso. Es decir, se crean roles abstractos dentro del paquete “común” y sus “especializaciones” se crean en los paquetes de las disciplinas correspondientes estableciendo relaciones de contribución entre éstos y la definición de rol del paquete “comun”. Por ejemplo, los roles “Arquitecto (requisitos)”, “Arquitecto (análisis)”, “Arquitecto (diseño)” y “Arquitecto (implementación)” de los paquetes para las disciplinas de “Requisitos”, “Análisis”, “Diseño” e “Implementación”, respectivamente, establecen una relación de contribución con el rol “Arquitecto” en el paquete “común”. Esto para representar la participación de dicho rol en las distintas disciplinas del

UP. La Figura 39 muestra los tres roles abstractos “Arquitecto”, “Ingeniero de componentes” e “Ingeniero de casos de uso” y en la Figura 40 se muestran todos los roles en sus respectivos Paquetes de contenido.

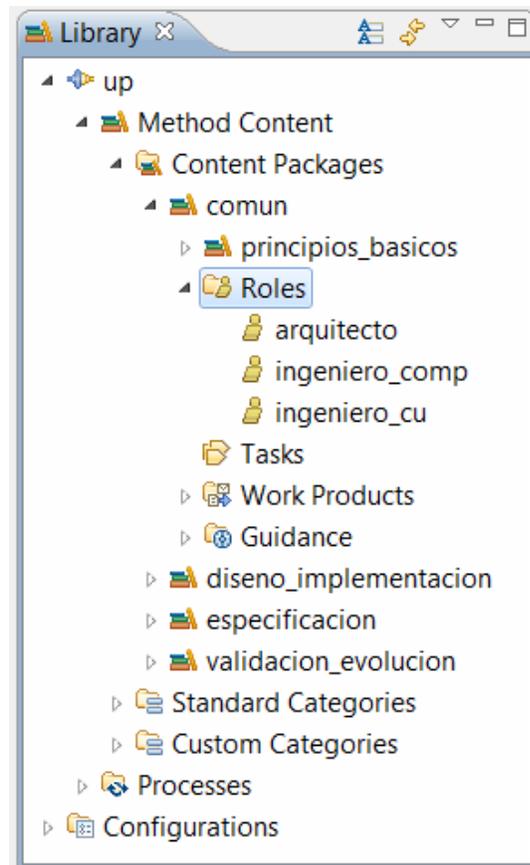


Figura 39. Roles abstractos en el *Content Package* “común”.



Figura 40. Roles del UP en sus respectivos Paquetes de contenido

## Productos del proceso

Se procede igual que con los roles, utilizando el paquete “común” para el caso de los productos de trabajo que pasan de una disciplina a otra adoptando formas distintas o cambiando su nivel de abstracción.

Al contrario de SPEM, en Jacoson y cols. (2000) no se incluye algún capítulo o apartado que especifique el tipo de los productos generados en cada actividad. Por ello, la única información sobre los productos de trabajo que

se puede extraer es:

- El nombre de los productos se obtiene de los productos de entrada y de salida de las tareas.
- En algunas técnicas, se dan detalles sobre el contenido y forma de algún producto (especificación de un caso de uso, modelos, etc.)

En consecuencia, cada producto de trabajo que se emplea como entrada o salida en alguna tarea se implementa como un *work product* dentro del *content package* correspondiente a cada una de las disciplinas y por defecto, el tipo del *work product* será *artifact* salvo en los casos siguientes:

- Si es un producto formado por otros productos tangibles más simples, será un *deliverable*, cuyas *deliverable parts* son dichos componentes más simples.
- Si el producto es intangible o es un entorno de trabajo formado por herramientas varias instaladas y configuradas, entonces será un *outcome*.

La Figura 41 muestra los productos de trabajo abstractos, agrupados en el paquete “común”.

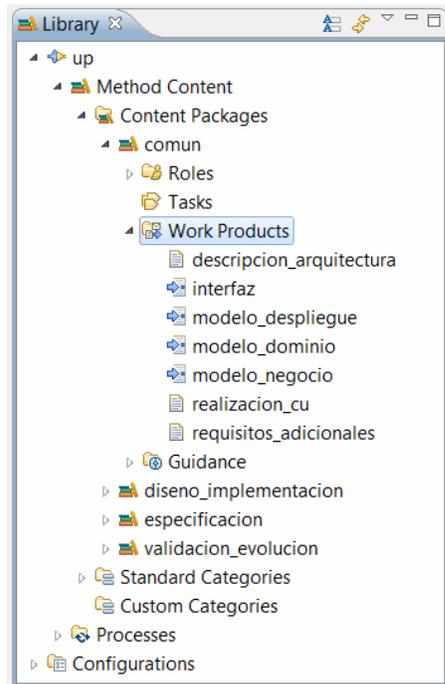


Figura 41. Productos de trabajo abstractos del *Content Package* “común”.

## Instrucciones

Se procede igual que con los roles y productos de trabajo, utilizando el paquete de contenido “común” y asignando uno de los distintos tipos de instrucciones de SPEM según sea el caso. La Figura 42 muestra parte de las *Guidances* para el UP.

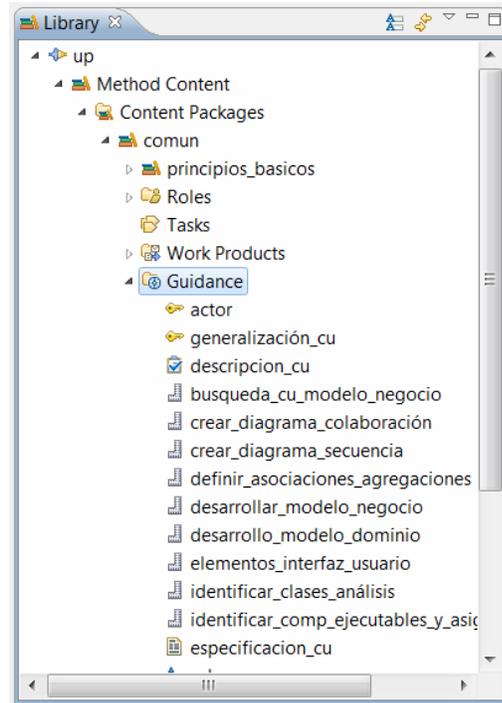


Figura 42. Instrucciones en el *Content Package* “comun”.

## Tareas

Cada tarea se implementa como una *Task* en el respectivo paquete de contenido para cada disciplina. En la Figura 43 se muestra un ejemplo de las tareas en el paquete de contenido “diseno”. Con fines organizativos, cada tarea se adscribe a una Categoría estándar (*Disciplines*) según la disciplina a la que pertenece, como se muestra en la Figura 44.

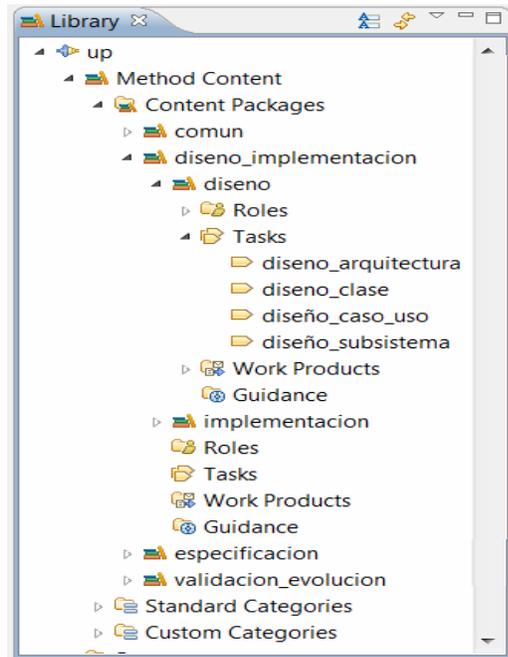


Figura 43. Tareas de la disciplina de Diseño.

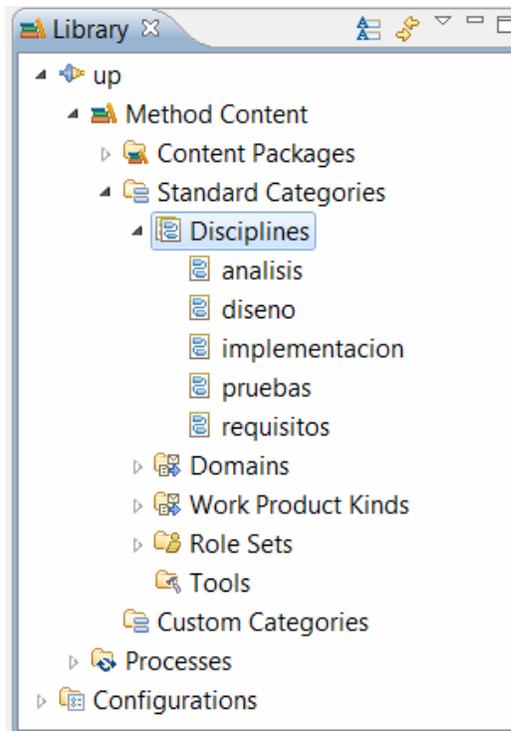


Figura 44. *Disciplines* del *Method Content* para el UP.

## Propósito y descripción de los Elementos de método

El propósito se coloca como *Purpose* y el texto descriptivo adicional se añade como *Main Description*. En la Figura 45 se muestra el propósito y la descripción de una tarea.

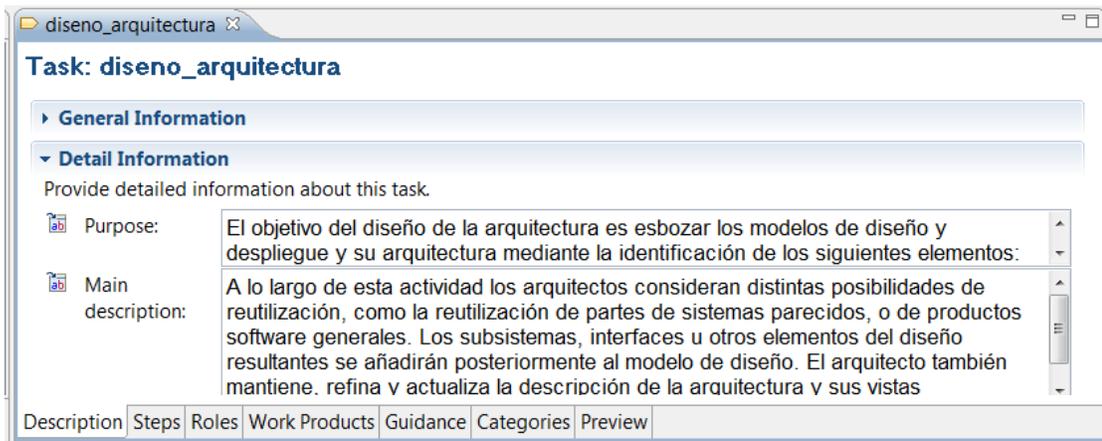


Figura 45. Propósito y Descripción principal de una tarea

## Relaciones entre Roles, Tareas y Productos de trabajo

Una vez definidos los elementos de método correspondientes a cada paquete de contenido se procedió a establecer la relación entre ellos.

Para cada rol en cada paquete de contenido se añadieron los productos de trabajo (previamente definidos) de los cuales éste es responsable. La Figura 46 muestra la vista de edición para la asignación de productos que son responsabilidad de un rol.

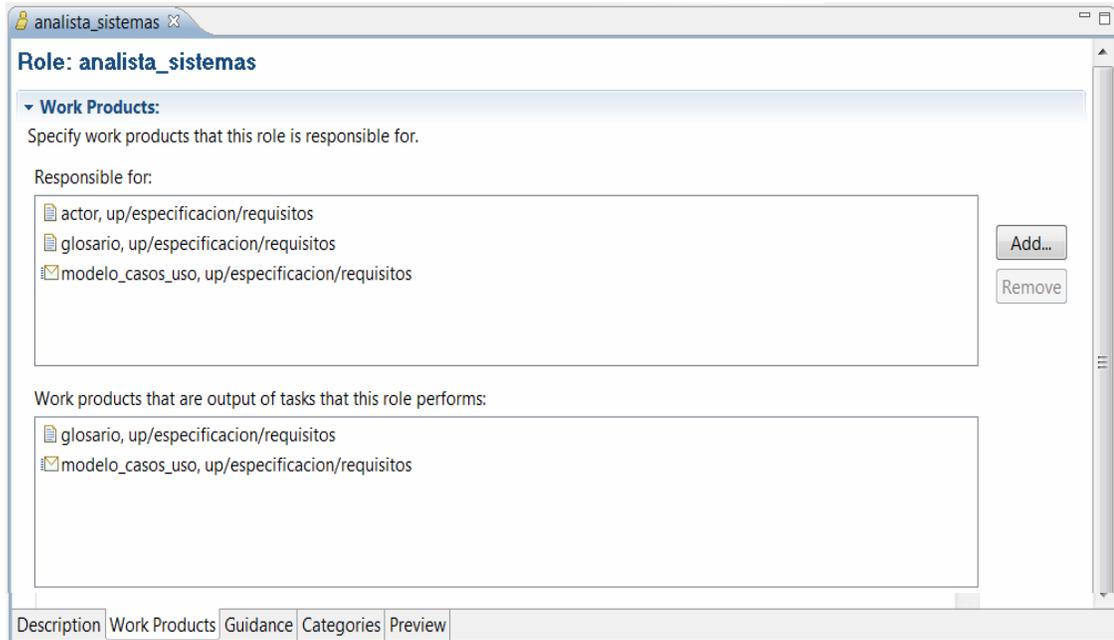


Figura 46. Productos de trabajos de los que un rol es responsable.

Así mismo, un rol específico se asigna a una categoría definida. En la Figura 47 se muestra un ejemplo de categorización del rol “Analista de sistemas”.

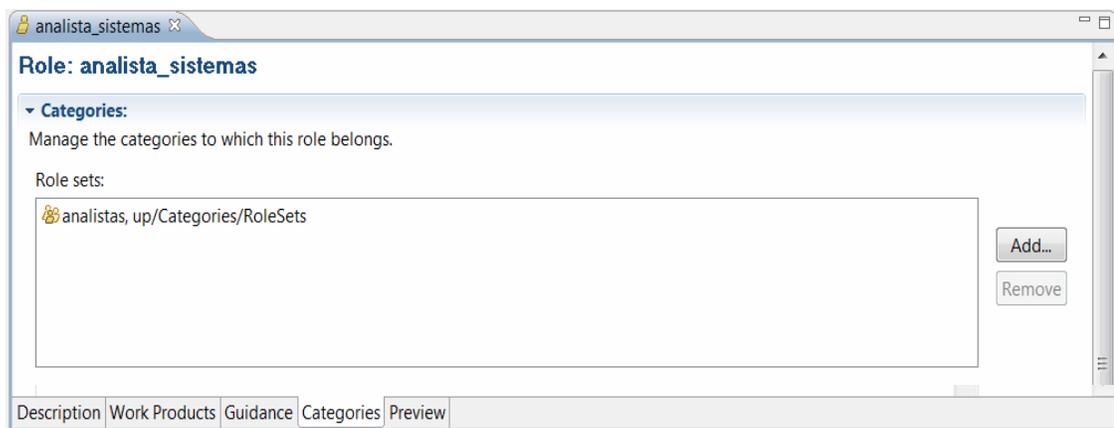


Figura 47. Asignación de un rol a una categoría definida.

En el caso de las tareas, éstas pueden incluir pasos que detallen su realización, se establecen los productos de trabajo que son entradas y

salidas, las guías relacionadas y una categorización. En las Figuras 48 a la 51 se muestran respectivamente la asignación en *EPF Composer* de las tareas, productos de trabajo, guías asociadas y la categorización de la tarea “Encontrar actores y casos de uso”.

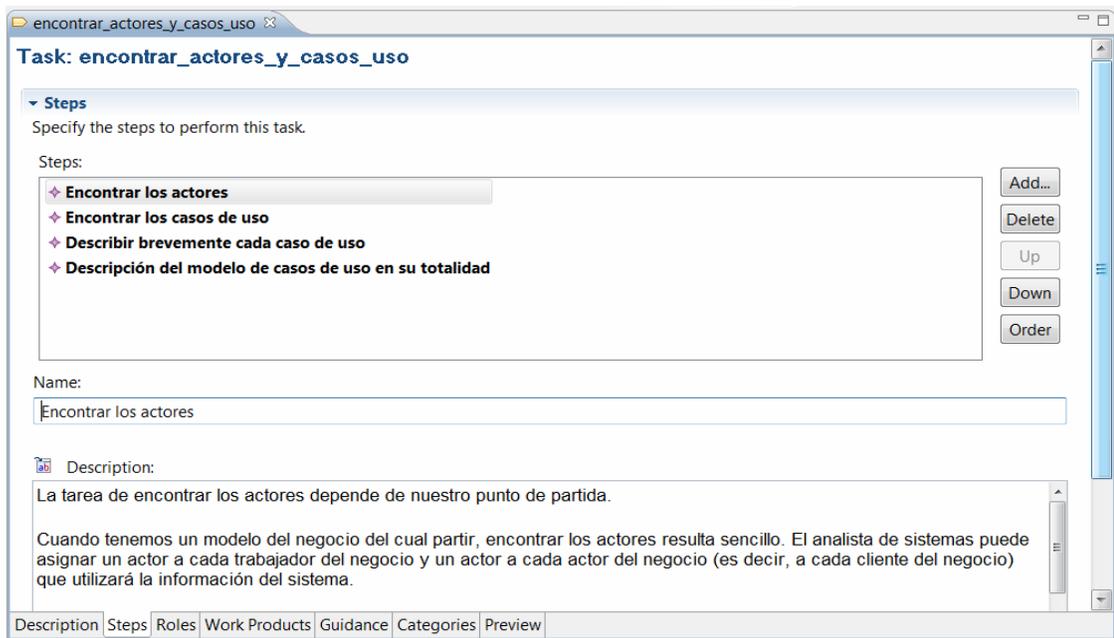


Figura 48. Asignación de pasos a una tarea.

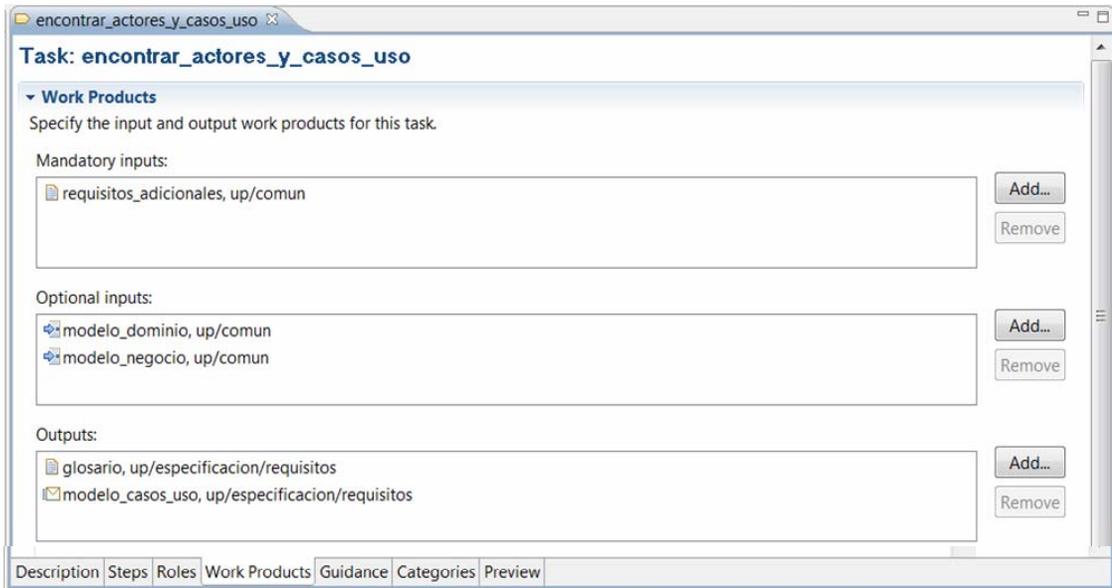


Figura 49. Asignación de productos de trabajo a una tarea.

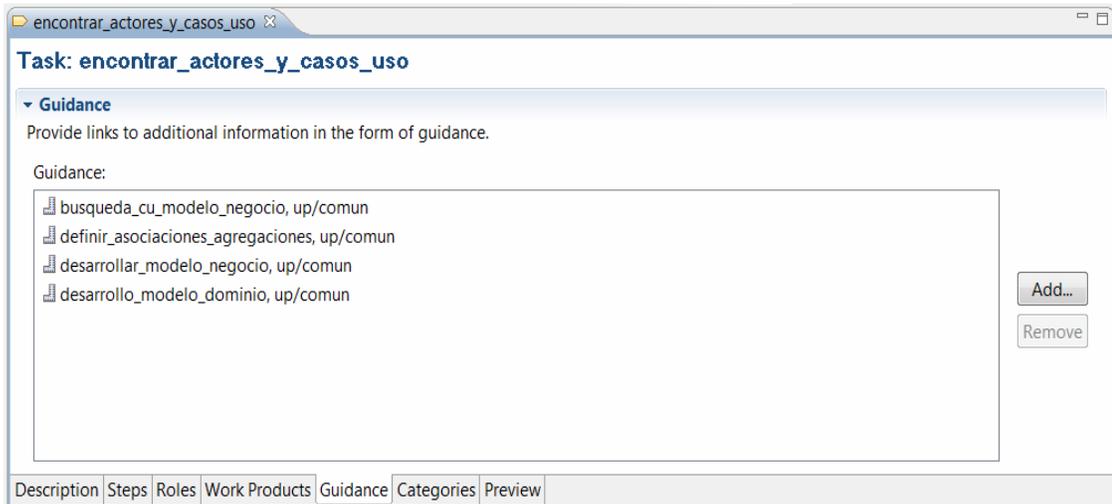


Figura 50. Asignación de guías a una tarea.



Figura 51. Asignación de una tarea a una categoría (*Disciplines*).

La Figura 52 muestra la relación entre tareas y productos de trabajo con el rol

“Analista de sistemas”. Éste realiza las tareas “Encontrar actores y casos de uso” y “Estructurar el modelo de casos de uso” y es responsable de los productos de trabajo “Actor”, “Glosario” y “Modelo de casos de uso”.



Figura 52 .Relación entre tareas y productos de trabajo con un rol

En el Apéndice A se muestran las relaciones entre tareas y productos de trabajo para los demás roles del proceso.

En cuanto a la organización del proceso (*Process*) las reglas son las siguientes.

## Patrones de proceso

Una vez definidos los elementos de contenido de método, éstos se reutilizaron para crear procesos. Para cada disciplina y actividad del proceso se definió un patrón de proceso (*Capability Pattern*). En el caso de las disciplinas se creó un paquete de proceso “Disciplinas” y otro paquete “Plantillas de Iteración” para los patrones de las iteraciones. La Figura 53 muestra la estructura resultante.

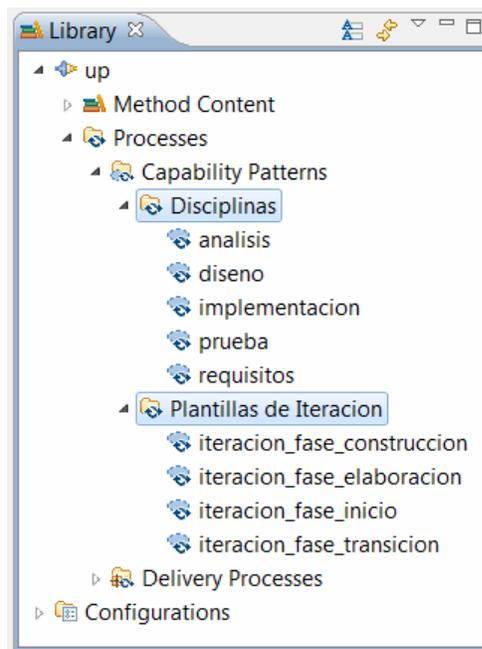


Figura 53. Patrones de proceso para el UP

El paquete de procesos “Disciplina” contiene cinco (5) patrones de proceso que reutilizan la información relacionada a cada tarea en el contenido de método. Cada patrón de proceso en este paquete representa una disciplina del UP. En la Figura 54 se muestra la estructura de desglose de trabajo resultante para el patrón de procesos “Análisis” y en el Apéndice B se puede observar los detalles de las restantes estructuras de desglose resultantes

para los otros cuatro (4) patrones de proceso.

Presentation Name	Index	Predec...	Model In...	Type	Plann...	Repeata...	Multipl...	Ongoi...	Event...	Optional
Analisis	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Analisis de la arquitectura	1			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analizar un caso de uso	2			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analizar una clase	3			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analizar un paquete	4			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 54. WBS del patrón de proceso “Análisis”.

El paquete de procesos “Plantillas de Iteración” contiene cuatro (4) patrones que reutilizan los cinco (5) patrones anteriores para cada iteración del UP. Estas iteraciones fueron armadas según los “Flujos de Trabajo Arquetípicos” de una iteración que son descritos en los Capítulos 13 al 16 de la Parte III del libro sobre UP. En la Figura 55 se muestra la estructura de desglose resultante para el patrón de procesos “Iteración de la Fase de Inicio [1..n]” y los restantes tres (3) patrones se pueden observar en el Apéndice C.

Presentation Name	Index	Predecessors	Model Info	Type	Plann...	Rcpc...	Multi...	Ongo...	Event...	Optio...
Iteración de la Fase de Inicio [1..n]	0			Capabili...	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Requisitos	1		extends 'requisi...	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Encontrar actores y casos de uso	2			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Priorizar casos de uso	3	2		Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Detallar un caso de uso	4	3		Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipar la interfaz de usuario	5	4		Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Estructurar el modelo de casos de uso	6	2,3,4		Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	7	1	extends 'analisis...	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Análisis de la arquitectura	8			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analizar un caso de uso	9	8		Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analizar una clase	10			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analizar un paquete	11			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	12	7	extends 'diseno...	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Diseño de la arquitectura	13			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño de una clase	14			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño de un caso de uso	15			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño de un subsistema	16			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	17		extends 'imple...	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Implementación de la arquitectura	18			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar una clase	19			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar un subsistema	20			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrar el sistema	21			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 55. Estructura del patrón “Iteración de la Fase de Inicio [1..n]”.

## ASPECTOS SOBRE TRABAJADORES

### Categorización de roles

Adicionalmente, los roles se organizan en conjuntos específicos de categorías estándar (*Standard Categories: Role Sets*): “Analistas”, “Diseñadores”, “Arquitectos” e “Ingenieros”, como se muestra en la Figura 56. Por ejemplo, el *Role Set* “Analistas” agrupa los roles “Analista de sistemas”, “Especificador de casos de uso” y “Diseñador de interfaz de usuario” (como se establece en el UP) del paquete para la disciplina de “Requisitos”.

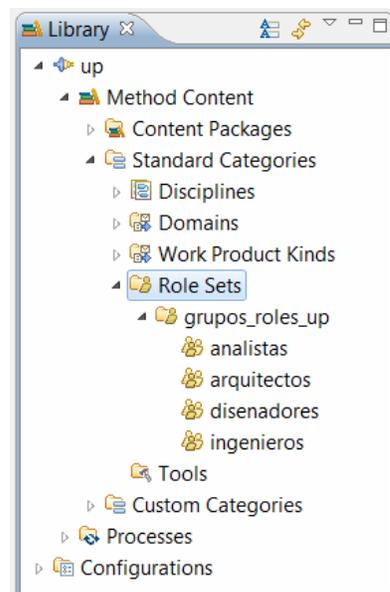


Figura 56. Categorización (*Role Sets*) de los roles del UP.

## ASPECTOS SOBRE PRODUCTOS

### Categorización de los productos

Los productos de trabajo se clasifican de dos maneras en paralelo. En las categorías estándar (*Standard Categories*), se emplean dominios (*Domains*) para distinguir entre artefactos de la disciplina de Requisitos, Análisis, Diseño, Implementación o Pruebas y adicionalmente, se emplean tipos de productos (*Work Product Kinds*) para establecer grupos de productos varios: Documentos, Modelos, Elementos de modelo, Plan y Especificación. La Figura 57 muestra esta clasificación.

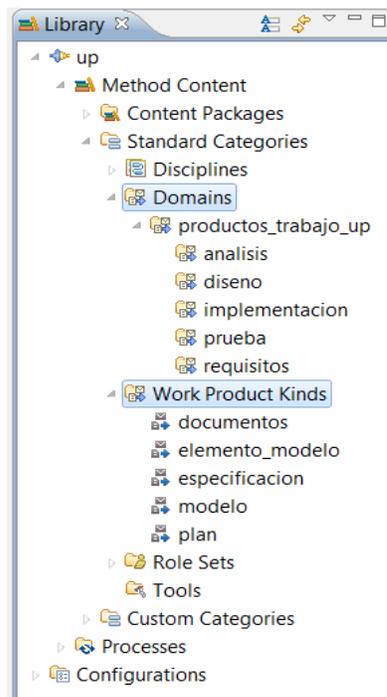


Figura 57. Categorización de productos de trabajo.

## **ASPECTOS SOBRE PROCESOS, ACTIVIDADES Y TAREAS**

A continuación se presentan las reglas relacionadas con las disciplinas, las actividades y tareas del UP, complementarias a las ya indicadas en el apartado de **Aspectos generales**. Estas reglas se extraen igualmente del análisis de los Capítulos 6 al 11 de Jacobson y cols. (2000).

Las reglas a nivel de **proceso** son:

Proceso: del apartado inicial de descripción y objetivos se derivan los valores de varias propiedades del *capability pattern* correspondiente al proceso: *name*, *presentation name*, *brief description*, *purpose*, y *main description*.

Desglose del trabajo: la WBS del *Capability Pattern* de cada proceso está formada, en su primer nivel de desglose, por los *Capability Patterns* de las actividades que forman el proceso.

Flujo del proceso: los diagramas que muestran el flujo de actividades se implementan en forma de diagramas de actividad del *Capability Pattern* del proceso.

Las reglas a nivel de **actividad** son:

Actividad: una actividad se representa en *EPF Composer* como un *Capability Pattern*. Las actividades de un mismo proceso se crearán en un paquete de procesos con el nombre del proceso al que pertenecen.

Desglose del trabajo: la WBS del *Capability Pattern* de cada actividad está formado por los *Task Descriptors* de las tareas que forman el proceso.

Dichos *Task Descriptors* se obtienen por herencia de las *Tasks* correspondientes del *Method Content*.

Flujo de trabajo: el flujo de trabajo de las actividades se representa mediante el *Activity Diagram*. También se utiliza el *Activity Detail Diagram* para mostrar las correspondencias entre participantes responsables de cada tarea y los productos de entrada y de salida.

Las reglas sobre aspectos del nivel de **tarea** son:

Tarea: cada tarea se implementa como una *Task* en el *Content Package*, dentro de la jerarquía del *Method Content*, que corresponde a la disciplina a la que pertenece la tarea.

Pasos de tarea: si en el texto descriptivo de la tarea se indica una lista de acciones o pasos a realizar, éstos se implementan como *Steps*. En otro caso, se pueden indicar como pasos la realización ordenada de los diferentes productos de salida. Para ahorrar la descripción del paso, de ser posible, el nombre del paso debe indicar quién hace qué o quién produce qué producto.

Productos de entrada: dependiendo de cuáles son obligatorios, cuyos productos de trabajo se asocian a la tarea como *Mandatory Input*, y cuáles opcionales, cuyos *Work Products* se asocian como *Optional Input*.

Productos de salida: sus productos de trabajo se asocian a la *Task* como *Outputs*.

Técnicas, prácticas, ejemplos, definiciones, plantillas, entre otros: se asocian

a la *Task* como tipos específicos (según se considere) de *Guidelines*.

Participantes: se asocian como *roles* a la *Task*. Se debe elegir uno como *Primary Performer* y, de ser el caso, el resto como *Additional Performers*.

## ASPECTOS SOBRE LA PRESENTACIÓN

Una vez creados todos los elementos de método y proceso esta parte del desarrollo de la especificación consistió en configurar los elementos que formarían parte de la publicación de contenidos para el UP.

### Configuración de método

Se creó la configuración de método correspondiente, que contiene los elementos que se mostrarán en la especificación resultante; se le dio una descripción que incluye un nombre, un nombre de presentación y una descripción breve. Se incorporó a esta configuración el contenido que la conformaría (todo el *Method Plug-in* desarrollado) y se definió la vista a presentar. La Figura 58 muestra este aspecto.

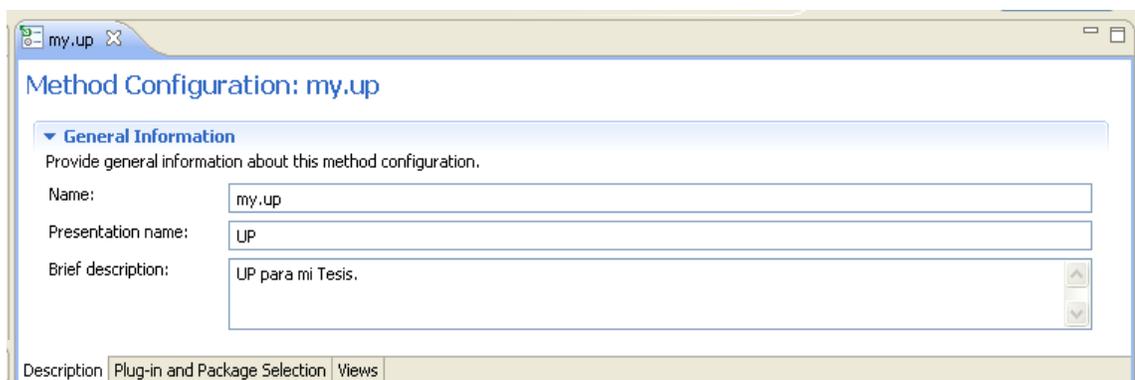


Figura 58. Descripción de la Configuración de método “my.up”.

Para especificar la vista que se incluiría cuando la configuración fuera publicada se crearon categorías personalizadas (*Custom Categories*). La estructura resultante (su contenido) se detalla a continuación y se muestra en La Figura 59.

La categoría personalizada “navegador\_up” que incluye otras dos categorías personalizadas:

- “introducción\_up” con los conceptos básicos del UP (dirigido por casos de uso, centrado en la arquitectura e iterativo e incremental) que fueron definidos previamente como conceptos (*Guidance*) en el paquete de contenido “principios\_basicos” dentro del paquete de contenido “comun”.
- “plantillas” que agrupa lógicamente todas las guías que son plantillas.

Las categorías estándares *Disciplines*, *Work Products Kinds* y *Roles Sets*.

Finalmente, el proceso de entrega (*Delivery Pattern*) “up” que reutiliza los patrones de proceso como fue descrito.

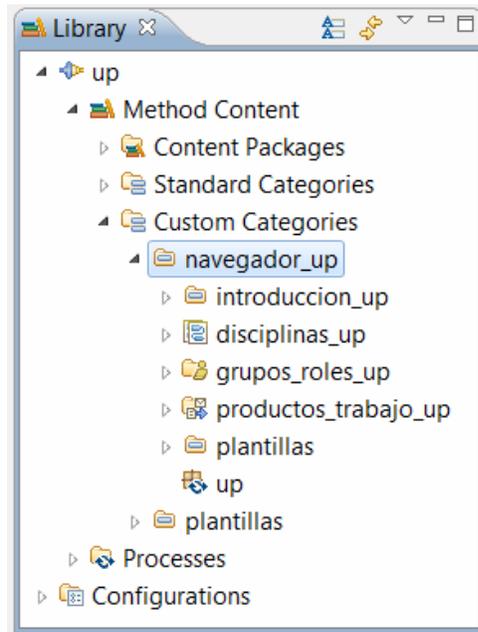


Figura 59. Organización del contenido para la especificación del UP.

La Figuras 60 y 61 muestran la selección de paquetes y *plug-in* y la vista para el UP, respectivamente.

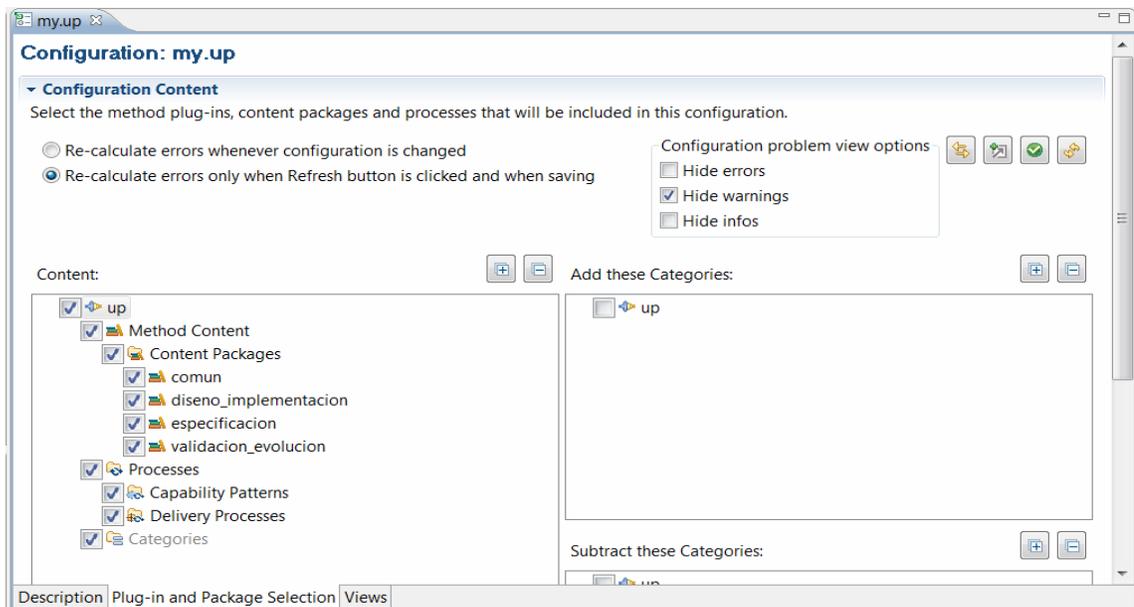


Figura 60. Selección de Paquetes y *Plug-in* para especificación del UP.

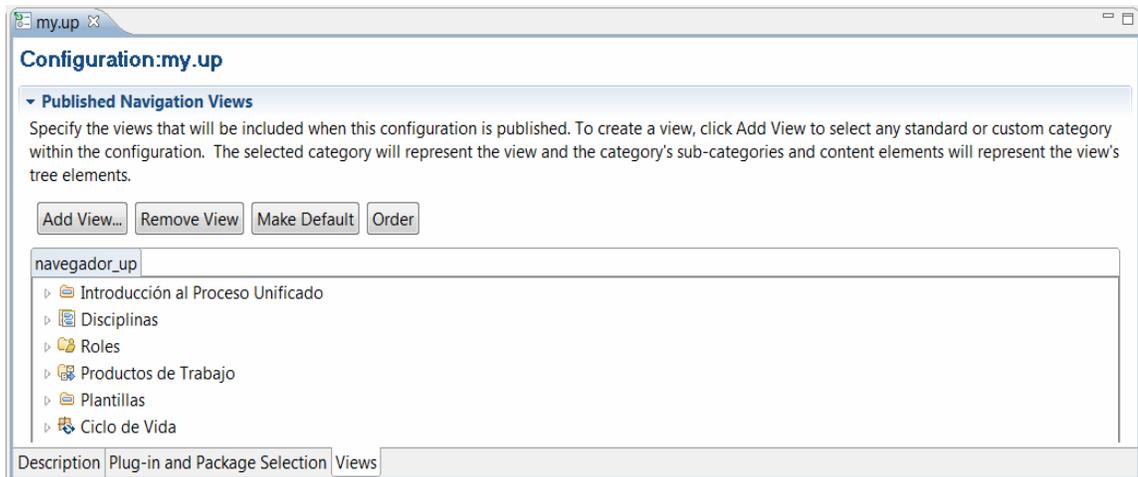


Figura 61. Vista de navegación de la especificación del UP.

Una vez establecida la configuración de método deseada se generó la especificación del UP. En la Figura 62 se muestra la página principal de la representación generada por *EPF Composer* a partir de las configuraciones realizadas. La misma puede ser accedida a través de un navegador web para explorar todo el contenido.

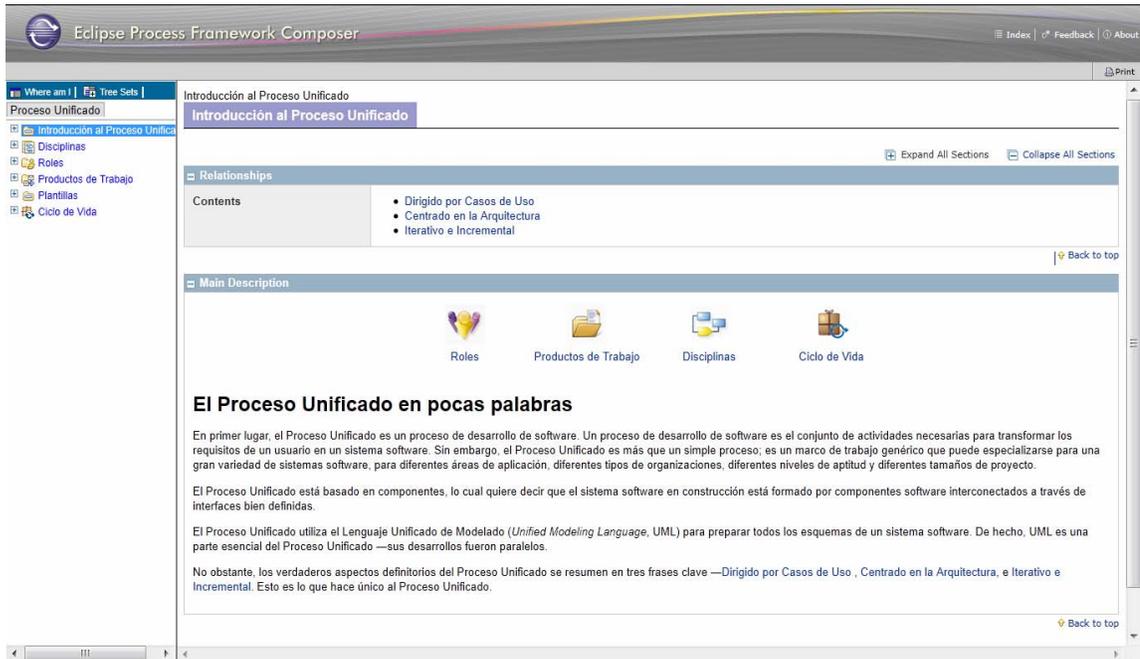


Figura 62. Homepage de la especificación resultante para el UP.

## CONCLUSIONES

De manera general, el uso de estándares es útil porque agrupan las buenas prácticas del desarrollo de software, engloban los conocimientos que son patrimonio de una organización, proporcionan un marco para implementar procedimientos de aseguramiento de la calidad y proporcionan continuidad entre el trabajo de distintas personas.

En este sentido, la especificación de PDS con el SPEM facilita la comprensión y la comunicación humana ya que se dispone de la representación de sus conceptos y elementos en una notación estándar que elimina ambigüedades, fomentando un lenguaje común. Es difícil combinar e integrar contenidos y procesos que están disponibles en su propio formato propietario; cada libro, manual, herramienta, utiliza un lenguaje y estilo diferentes. Además, el uso del SPEM proporciona capacidades para facilitar la reutilización; permite dar soporte a la mejora de procesos; dar soporte a la gestión de procesos; guiar la automatización de procesos y dar soporte para la ejecución automática.

La creación de procesos con el SPEM es de importante ayuda para las organizaciones que llevan a cabo proyectos de software, así como para los proyectos de desarrollo en el ámbito académico, de múltiples formas. Como un repositorio de conocimientos en el que se hallen almacenados los elementos de contenido que sirvan como biblioteca de conocimientos en la que se gestiona información sobre los productos que se generan, descripciones del trabajo que hay que realizar para elaborar dichos productos y guías sobre cómo se realizan, etc., realmente útiles para la educación y formación. Esta información puede quedar accesible de forma

organizada para las personas interesadas en ella, independientemente de que luego se reutilice en la creación de procesos o no.

En cuanto al mecanismo de implementación impuesto al caso de estudio elegido para este trabajo, se decidió implementar el SPEM a través del mecanismo de meta-modelado, ya que de esa forma se pueden definir todas las estructuras y reglas de estructuración para representar contenidos de métodos y procesos además de que se dispone de la sintaxis y semántica proporcionada por el lenguaje, permitiendo cumplir con los requisitos impuestos a la especificación del caso de estudio en la etapa de creación de la hipótesis. Esta elección tiene sus ventajas respecto al empleo del perfil UML (el otro mecanismo de implementación del SPEM) con el cual es posible utilizar una herramienta genérica de modelado UML pero con una importante desventaja: las reglas específicas de estructuración del SPEM, que sí están en el meta-modelo, no pueden ser manejadas con herramientas genéricas, salvo que se empleen restricciones OCL, lo que supone aumentar significativamente el nivel de dificultad de UML. Por ejemplo, para la asociación entre roles y tareas, no es posible controlar que debe haber siempre un realizador principal y, opcionalmente, varios adicionales.

La elección de implementación del SPEM a través de la herramienta *EPF Composer* permitió brindar un repositorio donde almacenar procesos, permitiendo su edición y recuperación de forma clara y fácil y el poder reutilizar y personalizar dichos componentes en un futuro. Además, dicha elección ofreció dos grandes ventajas. Desde el un punto de vista económico, la existencia de una herramienta gratuita permitió una utilización más rápida ya que no se hubo que invertir en gastos por el costo de la propia herramienta. El movimiento *Open Source* y el gran auge que está teniendo la plataforma Eclipse va a permitir que se conecte con otro tipo de herramientas

desarrolladas en esta plataforma.

El estándar SPEM, y por tanto la herramienta *EPF Composer*, ofrecen gran flexibilidad a la hora de crear métodos, metodologías y/o procesos de desarrollo de software. Se pueden crear repositorios de elementos de método sin la necesidad de crear ningún proceso, o se puede dar el caso contrario, es decir, crear un proceso definiendo sus elementos directamente sin haber definido antes ningún repositorio de elementos de método.

Para la implementación del SPEM sobre el caso de estudio se adoptó un enfoque de especificación ascendente que permitió la reutilización de los elementos de contenido básicos en la creación de patrones de proceso para representar actividades que se fueron componiendo para formar elementos de proceso más complejos (otros patrones) hasta, finalmente, definir la estructura completa del proceso.

En cuanto a la introducción del UP como caso de estudio, fue de gran utilidad debido a que se aprovecharon los beneficios de un proceso ya establecido y maduro. En este sentido, fue favorable su utilización debido a la separación que se hace en el UP de la estructura estática respecto a la estructura dinámica del proceso que se corresponde cónsonamente con la separación clara que hace SPEM del contenido de método con su utilización en el proceso.

Respecto a la metodología de investigación adoptada se puede puntualizar que se ha adquirido experiencia en la utilización de un método de investigación en IS, adaptada al tema concreto del trabajo, que se basó en el método experimental (hipotético-deductivo) de investigación científica y el método general de trabajo en la IS. El trabajo realizado ha seguido dicho

método. Aunque esta aportación no es relativa al objetivo de este Trabajo de Grado, se considera importante destacarla, porque sí constituye una aportación en cuanto al proceso de investigación en sí mismo, dentro del ámbito de la IS. No hay que olvidar, que el objetivo de un Trabajo de Grado no es sólo el de presentar aportaciones en un área de estudio, sino también el de realizar un trabajo de investigación que, si es riguroso, deberá seguir un método.

Así mismo, la incorporación de un método para el análisis y revisión de la literatura permitió enriquecer los resultados de la búsqueda de documentación para ser sistematizados y presentados en el marco teórico del presente informe de Trabajo de Grado.

En cuanto a los resultados obtenidos, éstos son principalmente los productos de trabajo resultantes de la realización de cada de una de las actividades planteadas en la metodología de investigación adoptada. Así mismo, son resultados de este trabajo la experiencia de investigación en IS por medio de la redefinición y sistematización de la propuesta de Marcos, el enriquecimiento y la especificación del caso de estudio plasmado de manera sistemática en este informe de Trabajo de Grado.

Por otro lado, también se brinda la información recopilada en la investigación y el estudio realizado sobre SPEM y el uso de la herramienta *EPF Composer*, así como también dar la posibilidad del surgimiento de dudas e iniciativas para investigar nuevos temas relacionados, como por ejemplo, la implementación de otros PML.

La implementación del SPEM sobre el UP permitió obtener un *plug-in* que brinda la posibilidad de adaptación para cada proyecto particular según las necesidades y prioridades que se consideren más relevantes. El *plug-in*

permitirá contar con una aplicación estática (la especificación), con posibilidad, incluso, de ser puesta a disposición de los interesados en estudiar el UP, y que podrá ser reutilizado en otros proyectos. Gracias a él, se podrá también obtener diferentes vistas de la información según la configuración creada para el trabajo y además se posibilitará generar otras instancias del mismo; con más o menos contenido según el paradigma del proceso a modelar.

## RECOMENDACIONES

Luego de la culminación de este trabajo, es consideración del autor del mismo enunciar las siguientes recomendaciones que se deberán tener en cuenta en virtud de la experiencia adquirida y los resultados obtenidos.

En primer lugar, se recomienda fomentar la descripción de procesos y/o metodologías de desarrollo de software con algún PML por los beneficios que ya se han considerado en las conclusiones, por lo que es conveniente y de vital importancia estudiar, por ejemplo, el ISO/IEC 24744 que describe otro enfoque para el modelado y especificación de procesos.

En cuanto a la implementación del SPEM en *EPF Composer*.

A la hora de especificar procesos y atendiendo a que las propiedades de mantenibilidad, reusabilidad y escalabilidad sean las mejores posibles se recomienda la adopción de un enfoque ascendente en el que se crean primero los elementos más sencillos que se van componiendo para formar elementos de proceso más complejos hasta, finalmente, definir la estructura completa del proceso.

Aunque el orden en la creación de los elementos de método no es relevante, por comodidad se aconseja seguir el siguiente orden:

1. Crear las guías, ya que pueden ser referenciadas desde cualquier otro elemento de método.
2. Crear los productos de trabajo y asociarle las guías convenientes.
3. Crear los roles e indicar los productos de trabajo de los que son responsables y asociarle las guías convenientes.

4. Crear las tareas e indicar los roles que participan en la tarea, los productos de trabajo que son entrada o salida de la tarea y las guías convenientes.

La ventaja de respetar este orden es que cuando se crea un elemento ya existen todos los elementos con los que está relacionado y, por tanto, su edición es más cómoda.

Para especificar un proceso con SPEM en *EPF Composer* hay que tener en cuenta que esta herramienta no impone restricciones de jerarquía sobre los tipos de actividad (por ejemplo, podemos incluir una fase dentro de una iteración, lo cual no tiene sentido) por lo que, es responsabilidad del ingeniero de procesos definir una jerarquía que sea lógica.

Finalmente, y sin otra intención más que aportar al desarrollo, avance y acercamiento de la academia a las tecnologías inherentes a la IS, se aconseja la inclusión de temas relacionados con la Ingeniería de procesos de software, tales como la definición de procesos y notaciones para el modelado de los mismos, como se plantea en el SWEBOK, al programa de estudio de la Licenciatura en Informática en la UDO.

## BIBLIOGRAFÍA

Acuña, S. y Ferré, X. 2001. Software process modelling. [Documento en línea]. Consultado el 02 de enero de 2009 en: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.23.5438&rep=rep1&type=url&i=0>

Arias, F. 2006. El proyecto de la investigación. introducción a la metodología científica. Quinta edición. Editorial Episteme. Caracas, Venezuela.

Bocchino, W. 1990. Sistemas de información para la administración: técnicas e instrumentos. Segunda edición. Editorial Trillas, S.A. de C.V. México.

Booch, G; Rumbaugh, J. y Jacobson I. 2006. El lenguaje unificado de modelado. Pearson Educación S.A., Madrid.

Botella i López, P.; Franch-Gutiérrez, X. y Ribó-Balust, J. 2004. Lenguajes de modelización de procesos de software basados en UML (Resumen). Novática: Revista de la asociación de técnicos de informática, 171. [Revista en línea]. Consultado el 02 de febrero de 2008 en: <http://www.ati.es/novatica/2004/171/nv171sum.html#art27>

Bunge, M. 1976. La investigación científica. Quinta edición. Editorial Ariel. Barcelona, España.

Caro, M.; Rodríguez, A.; Calero, C.; Fernández-Medina, E. y Piattini, M. 2005. Análisis y revisión de la literatura en el contexto de proyectos de fin de carrera: una propuesta. Revista Sociedad Chilena de Ciencia de la Computación [Documento en línea]. Consultado el 02 de febrero de 2008 en: <http://www.dcc.uchile.cl/~mmarin/revista-sccc/sccc-web/volumen6-1.html>

Chalmers, A. 1989. ¿Qué es esa cosa llamada ciencia?. Octava edición. México Siglo XXI. México.

Chirinos, L. 2003. Un proceso para especificar, medir y controlar la calidad de productos de software durante el proceso de desarrollo. [Documento en línea]. Consultado el 21 de diciembre de 2009. <http://www.postgrado.ucv.ve/biblioteca/tesis.asp?id=TCi857&fecha=3>

Curtis, B.; Kellner, M. I. y Over, J. 1992. Process modeling. *Communications*

of the ACM 35, 9 (Sep. 1992), 75-90. Consultado el 12 de febrero de 2009 en: <http://doi.acm.org/10.1145/130994.130998>

Fuggetta, A. 2000. Software process: a roadmap. International Conference on Software Engineering. Proceedings of the Conference on The Future of Software Engineering. Consultado el 02 de febrero de 2008 en: <http://portal.acm.org/citation.cfm?id=336521>

García, G. y Rodríguez, J. 2001. Aplicación del modelado de procesos en un curso de ingeniería de software. Revista Electrónica de Investigación Educativa, 3 (2). Consultado el 21 de febrero de 2008 en: <http://redie.uabc.mx/vol3no2/contenido-mireles.html>

Hernández, R.; Fernández, C. y Baptista, P. 2003. Metodología de la investigación. Tercera edición. McGraw-Hill/Interamericana Editores, S.A. de C.V. México.

IEEE Computer Society. 2004. *Guide to the Software Engineering Body of Knowledge*. "IEEE Computer Society". Consultado el 02 de febrero de 2008 en: <http://www.computer.org/portal/web/swebok/html/ch1#Ref1>

Jacobson I.; Booch, G. y Rumbaugh, J. 2000. El proceso unificado de desarrollo de software. Pearson Educación S.A., Madrid.

Kendall, K. y Kendall, J. 1997. Análisis y diseño de sistemas. Tercera edición. Prentice Hall Hispanoamericana, S.A. México.

Larman, C. 1999. UML y patrones. Introducción al análisis y diseño orientado a objetos. Prentice Hall, México.

Llórens, J. 1991. Sistemas de información. Planificación, análisis y diseño. Ciclo de desarrollo de sistemas. Tercera edición. Editorial Miro C.A. Venezuela.

Lonchamp, J. 1993. A structured conceptual and terminological framework for software process engineering. Proceedings of the Second International Conference on the Software Process. [Documento en línea]. Consultado el 21 de mayo de 2009 en: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.658>

Marcos, E. 2002. Investigación en ingeniería del software vs. desarrollo software. Métodos de Investigación y Fundamentos Filosóficos en Ingeniería del Software y Sistemas de Información (MIFISIS) [Documento en línea].

Consultado el 02 de febrero de 2008 en:  
<http://kybele.escet.urjc.es/MIFISIS2002/programa.htm>

Ministerio de Administraciones Públicas. MÉTRICA Versión 3. Metodología de Planificación, Desarrollo y Mantenimiento de Sistemas de Información. España, Madrid. Consultado el 05 de febrero de 2008 en:  
<http://administracionelectronica.gob.es/?nfpb=true&pageLabel=P60085901274201580632&langPae=es>

Montilva, J. 1986. Desarrollo de sistemas de información. ULA, Consejo de Publicaciones. Mérida, Venezuela.

Naur, P. y Randell, B. (eds). 1969. Software Engineering: Report on a Conference Sponsored by the NATO Science Committee, (7 – 11 October 1968). Brussels, Scientific Affairs Division, NATO. [Documento en línea]. Consultado el 21 de mayo de 2009 en:  
<http://homepages.cs.ncl.ac.uk/brian.randell/NATO/>

OMG-MOF. Object Management Group. 2006. Meta Object Facility (MOF) Core Specification Version 2.0. [Documento en línea]. Consultado el 16 de noviembre de 2007 en: <http://www.omg.org/spec/MOF/2.0/PDF>

OMG-SPEM. Object Management Group. 2005. Software Process Engineering Metamodel Specification Version 1,1. [Documento en línea]. Consultado el 16 de noviembre de 2007 en:  
<http://www.omg.org/docs/formal/06-01-01.pdf>

OMG-SPEM. Object Management Group. 2008. Software & System Process Engineering Meta-model Specification version 2.0. [Documento en línea]. Consultado el 10 de mayo de 2008 en: <http://www.omg.org/spec/SPEM/2.0/>

Pollice, G. 2003. Using the RUP for small projects: Expanding upon eXtreme Programming. A Rational Software White Paper. [Documento en línea]. Consultado el 11 de febrero de 2009 en:  
<http://www.cs.buap.mx/~dpinto/semadoo/tp183.pdf>

Pressman, R. 2002. Ingeniería del software. Un enfoque práctico. Quinta edición. McGraw-Hill/Interamericana de España, S.A.U.  
Pressman, R. 2005. Ingeniería del software. Un enfoque práctico. Sexta edición. McGraw-Hill Companies, Inc.

Real Academia Española. Diccionario de la Lengua Española. Edición Electrónica, versión 22.1.0, Espasa Calpe S.A. 2001. Disponible en:

<http://buscon.rae.es/drael/>

Romero, D. y Uva, M. 2005. De los procesos de desarrollo a la definición de procesos de un workflow. Argentine Symposium on Information Systems (ASIS), en el marco de las 34 Jornadas Argentinas de Informática e Investigación Operativa (JAIIO34). Universidad Nacional del Rosario. Argentina. [Documento en línea]. Consultado el 05 de febrero de 2008 en: <http://www.danielomarromero.com.ar/main0194.html?op=pub>

Riesco, D. y Romero, D. 2004. Un workflow que automatice los procesos de negocio del Proceso Unificado Rational. VI Workshop de Investigadores en Ciencias de la Computación (WICC), organizado por la Red de Universidades Nacionales con carreras de Informática (RedUNCI) Universidad Nacional del Comahue, Argentina. Consultado el 05 de febrero de 2008 en: <http://users.dsic.upv.es/~dromero/papers/wicc2004.pdf>

Rumbaugh, J.; Jacobson I. y Booch, G. 2000. El lenguaje unificado de modelado. Manual de referencia. Pearson Educación S.A., Madrid.

Sabino, C. 1978. Metodología de la investigación. El Cid Editor. Caracas, Venezuela.

Sabino, C. 1994. Cómo hacer una tesis. Editorial Panapo, C.A. Caracas, Venezuela.

Sabino, C. 2000. El proceso de investigación. Editorial Panapo, C.A. Caracas, Venezuela.

Senn, J. 1992. Análisis y diseño de sistemas de información. Segunda edición. McGraw-Hill Interamericana de México S.A. de C.V. México.

Sommerville, I. 2005. Ingeniería del software. Séptima edición. Pearson Educación, S.A. Madrid.

Standler, R. 1998. Creativity in Science and Engineering. Consultado el 05 de febrero de 2008 en: <http://www.rbs0.com/create.htm>

Upchurch, R. y Sims-Knight, J. 1997. Designing process-based software curriculum. Proceedings of the 10th Conference on Software Engineering Education and Trainin. Virginia Beach, VA, USA. Consultado el 21 de mayo de 2009 en: <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4479>

Whitten, J.; Bentley, L. y Barlow, V. 2003. Análisis y diseño de sistemas de información. Tercera edición. McGraw-Hill Interamericana Editores, S.A. de C.V.

## **APÉNDICES**

## APÉNDICE A

### **Relaciones entre Roles con las Tareas y Productos de Trabajo en el UP**

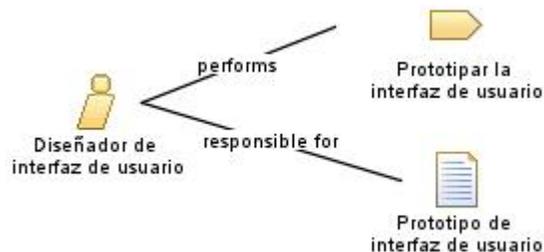
En este apéndice se muestran las relaciones entre cada Rol del proceso con las Tareas que realiza y los Productos de Trabajo de los cuales es responsable en cada una de las disciplinas del UP.

### Roles en la disciplina de Requisitos

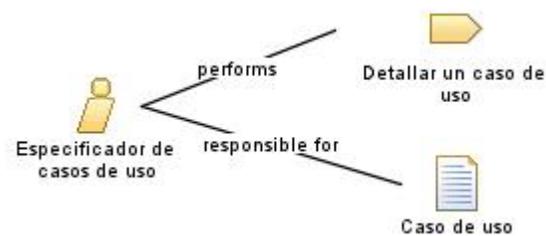
Tareas que realiza el rol “Analista de sistemas” y los Productos de Trabajo de los cuales es responsable



Tareas que realiza el rol “Diseñador de interfaz de usuario” y los Productos de Trabajo de los cuales es responsable



Tareas que realiza el rol “Especificador de casos de uso” y los Productos de Trabajo de los cuales es responsable





Tareas que realiza el rol “Diseñador de pruebas” y los Productos de Trabajo de los cuales es responsable



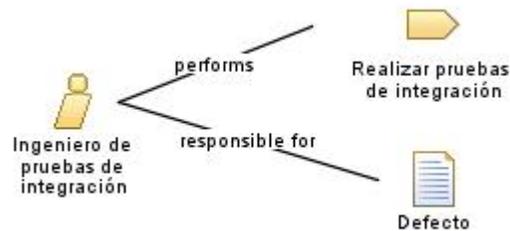
Tareas que realiza el rol “Ingeniero de componentes” y los Productos de Trabajo de los cuales es responsable



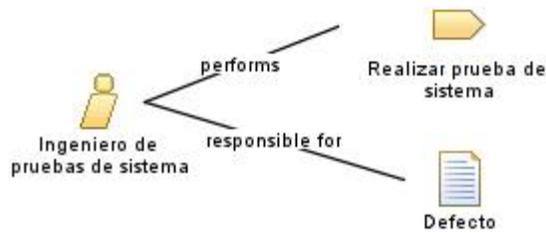
Tareas que realiza el rol “Ingeniero de casos de uso” y los Productos de Trabajo de los cuales es responsable



Tareas que realiza el rol “Ingeniero de pruebas de integración” y los Productos de Trabajo de los cuales es responsable



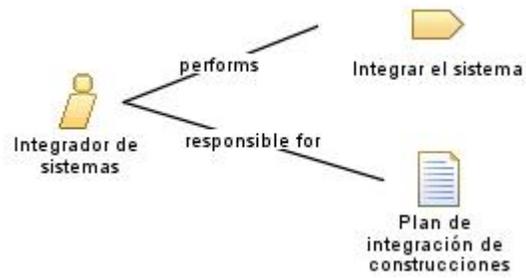
Tareas que realiza el rol “Ingeniero de pruebas de sistema” y los Productos de Trabajo de los cuales es responsable



Tareas que realiza el rol “Arquitecto” y los Productos de Trabajo de los cuales es responsable



Tareas que realiza el rol “Integrador de sistemas” y los Productos de Trabajo de los cuales es responsable



## **APÉNDICE B**

**Estructuras de desglose de los Patrones de proceso configurados para  
las disciplinas del UP**

En este apéndice se muestran las estructuras de desglose (*Breakdown Structure*) de los patrones de procesos (*Capability Patterns*) configurados para cada una de las disciplinas del UP en el paquete de procesos (*Process Package*) “Disciplinas”.

### Estructuras de desglose del Patrón de proceso para la disciplina “Requisitos”.

Estructura de desglose de trabajo (*Work Breakdown Structure*):

Presentation Name	Index	Predecessors	Mo...	Type	Pla...	Repe...	Multi...	Ongo...	Event...	Optio...
Requisitos	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Encontrar actores y casos de uso	1			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Priorizar casos de uso	2	1		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Detallar un caso de uso	3	2		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipar la interfaz de usuario	4	3		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Estructurar el modelo de casos de uso	5	1,2,3		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Asignación de equipos (*Team Allocation*):

Presentation Name	Model Info	Team	Type	Planned	Multiple Occurrences	Optional
Requisitos			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Especificador de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de interfaz de usuario			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Productos de trabajo en uso (*Work Product Usage*):

Presentation Name	Model Info	E...	E...	Deliverable	Type	Plann...	Multi...
Requisitos					Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Glosario	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input,Output				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Actor				Modelo de casos de uso	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso				Modelo de casos de uso	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input,Optional Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Prototipo de interfaz de usuario	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Actor					Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*):

Presentation Name	Index	Predecessors	Model Info	Type	Plann...	Repe...	Multi...	Ongo...	Event...	Optio...
Requisitos	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Encontrar actores y casos de uso	1			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Priorizar casos de uso	2	1		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del moc			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Detallar un caso de uso	3	2		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Especificador de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipar la interfaz de usuario	4	3		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de interfaz de usuario			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Optional Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipo de interfaz de usuario			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Estructurar el modelo de casos de uso	5	1,2,3		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

## Estructuras de desglose del Patrón de proceso para la disciplina de “Análisis”.

Estructura de desglose de trabajo (*Work Breakdown Structure*):

Presentation Name	Index	Predecessors	Model Info	Type	Plann...	Repe...	Multi...	Ongo...	Event...	Optio...
▲ Análisis	0			Capabilit...	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
↳ Análisis de la arquitectura	1			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Analizar un caso de uso	2			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Analizar una clase	3			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Analizar un paquete	4			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Asignación de equipos (*Team Allocation*):

Presentation Name	Model Info	Team	Type	Planned	Multiple Occurrences	Optional
▲ Análisis			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Ingeniero de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Productos de trabajo en uso (*Work Product Usage*):

Presentation Name	Model Info	Entry ...	Exit S...	Deliverable	Type	Plann...	Multi...	Optio...
▲ Análisis					Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Descripción de la arquitectura (vista del modelo de casos de uso)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Modelo de casos de uso	Mandatory Input				Deliverable Descript...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Modelo del dominio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Modelo del negocio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Clase de análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Paquete del análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
↳ Realización de caso de uso-análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*):

Presentation Name	Index	Predecessors	Model Info	Type	Plann...	Repe...	Multi...	Ong...
▲ Análisis	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Análisis de la arquitectura	1			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar un caso de uso	2			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar una clase	3			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar un paquete	4			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## Estructuras de desglose del Patrón de proceso para la disciplina de “Diseño”.

Estructura de desglose de trabajo (*Work Breakdown Structure*):

Presentation Name	In...	Predecessors	Model Info	Type	Plann...	Repe...	Multi...	Ongo...	Event...	Optio...
▲ Diseño	0			Capabilit...	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Diseño de la arquitectura	1			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño de una clase	2			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño de un caso de uso	3			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño de un subsistema	4			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Asignación de equipos (*Team Allocation*):

Presentation Name	Model Info	Team	Type	Planned	Multiple Occurrences	Optional
▲ Diseño			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Productos de trabajo en uso (*Work Product Usage*):

Presentation Name	Model Info	Entry ...	Exit S...	Deliverable	Type	Plann...	Multi...	Optio...
▲ Diseño					Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño	Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*):

Presentation Name	Index	Predecessors	Model Info	Type	Plann...	Repe...	Multi...	Ongo...	Ev...
Diseno	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Diseno de la arquitectura	1			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Clase del diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Descripción de la arquitectura (vista del modelo de despliegue)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de despliegue			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Subsistema de diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Diseno de una clase	2			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Clase de análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Clase del diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Realización de caso de uso-diseño			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Clase del diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Diseno de un caso de uso	3			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Ingeniero de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de despliegue			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Clase del diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Realización de caso de uso-diseño			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Subsistema de diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Diseno de un subsistema	4			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Subsistema de diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

## Estructuras de desglose del Patrón de proceso para la disciplina de “Implementación”.

Estructura de desglose de trabajo (*Work Breakdown Structure*):

Presentation Name	Index	Predecessors	Model Info	Type	Plann...	Repe...	Multi...	Ongo...	Event...	Optio...
Implementación	0			Capabilit...	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Implementación de la arquitectura	1			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar una clase	2			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar un subsistema	3			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrar el sistema	4			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de unidad	5			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Asignación de equipos (*Team Allocation*):

Presentation Name	Model Info	Team	Type	Planned	Multiple Occurrences	Optio...
Implementación			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrador de sistemas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Productos de trabajo en uso (*Work Product Usage*):

Presentation Name	Model Info	Entry ...	Exit S...	Deliverab...	Type	Planned	Multi...	Optio...
Implementación					Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente	Mandatory Input,Output				Deliverable Descri...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de implementación	Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descri...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*):

Presentation Name	Index	Pre...	Model Info	Type	Plann...	Repe...	Multi...	Ongo...	Event...
Implementation	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementation of the architecture	1			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar una clase	2			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar un subsistema	3			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de implementación			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrar el sistema	4			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrador de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de unidad	5			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description  
 Work Breakdown Structure  
 Team Allocation  
 Work Product Usage  
 Consolidated View

## Estructuras de desglose del Patrón de proceso para la disciplina de “Prueba”.

Estructura de desglose de trabajo (*Work Breakdown Structure*):

Presentation Name	Index	Predecessors	Model Info	Type	Plann...	Repe...	Multi...	Ongo...	Event...	Optio...
Prueba	0			Capabilit...	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Planificar prueba	1			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñar prueba	2			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar prueba	3			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar pruebas de integración	4			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de sistema	5			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Evaluar prueba	6			Task Des...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Asignación de equipos (*Team Allocation*):

Presentation Name	Model Info	Team	Type	Planned	Multiple Occurrences	Optional
Prueba			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de sistema			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de integración			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

### Productos de trabajo en uso (*Work Product Usage*):

Presentacion Name	Model Info	Entry ...	Exit S...	Deliverable	Type	Plann...	Multi...	Optio...
Prueba					Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Defecto	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Evaluación de prueba	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*):

Presentation Name	Index	Predecessors	Model Info	Type	Plann...	Repe...	Multi...
Prueba	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Planificar prueba	1			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñar prueba	2			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar prueba	3			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar pruebas de integración	4			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de integración			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Defecto			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de sistema	5			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de sistema			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Defecto			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Evaluar prueba	6			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Defecto			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Evaluación de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## **APÉNDICE C**

**Estructuras de desglose de los Patrones de proceso para las  
iteraciones arquetípicas del UP**



### Asignación de equipos (*Team Allocation*):

Presentation Name	Model Info	Team	Type	Planned	Multiple Occurrences	Optional
Iteración de la Fase de Inicio [1..n]			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	extends 'requisitos, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Especificador de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de interfaz de usuario			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	extends 'analisis, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	extends 'diseno, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	extends 'implementacion, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrador de sistemas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	extends 'prueba, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de sistema			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de integración			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

### Productos de trabajo en uso (*Work Product Usage*):

Presentation Name	Model Info	Ent...	Exit S...	Del...	Type	Planned	Multi...	Optio...
Iteración de la Fase de Inicio [1..n]					Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	extends 'requisitos, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input,Output				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input,Optional Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipo de interfaz de usuario	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Actor					Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	extends 'analisis, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Productos de trabajo en uso (*Work Product Usage*). Continuación:

Presentation Name	Model Info	Ent...	Exit S...	Del...	Type	Planned	Multi...	Optio...
Iteración de la Fase de Inicio [1..n]					Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	extends 'diseno, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño	Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	extends 'implementacion, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente	Mandatory Input,Output				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de implementación	Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	extends 'prueba, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Defecto	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Evaluación de prueba	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*):

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repeata...	Multi...
Iteración de la Fase de Inicio [1..n]	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	1		extends 'requisitos, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Encontrar actores y casos de uso	2			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Priorizar casos de uso	3	2		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Detallar un caso de uso	4	3		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Especificador de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipar la interfaz de usuario	5	4		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de interfaz de usuario			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Optional Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipo de interfaz de usuario			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Estructurar el modelo de casos de uso	6	2,3,4		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repeata...	Multi...
▲ Análisis	7	1	extends 'análisis, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Análisis de la arquitectura	8			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar un caso de uso	9	8		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar una clase	10			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar un paquete	11			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repeata...	Multi...
▲ Diseño	12	7	extends 'diseno, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Diseño de la arquitectura	13			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Diseño de una clase	14			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Diseño de un caso de uso	15			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Diseño de un subsistema	16			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description Work Breakdown Structure Team Allocation Work Product Usage Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repeata...	Multi...
Implementation	17		extends 'implementacio...	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementation of the architecture	18			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar una clase	19			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar un subsistema	20			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de implementación			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrar el sistema	21			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrador de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de unidad	22			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repeata...
Prueba	23		extends 'prueba, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Planificar prueba	24			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación,			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseñar prueba	25			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación,			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Implementar prueba	26			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Realizar pruebas de integración	27			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de integración			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Defecto			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de sistema	28			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de sistema			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Defecto			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Evaluar prueba	29			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Defecto			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Evaluación de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>



### Asignación de equipos (*Team Allocation*):

Presentation Name	Model Info	Team	Type	Planned	Multiple Occurrences	Optional
Iteration de la Fase de Elaboración [1..n]			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	extends 'requisitos, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Especificador de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de interfaz de usuario			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	extends ' analisis, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	extends 'diseño, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	extends 'implementación, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrador de sistemas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	extends 'prueba, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de sistema			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de integración			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Productos de trabajo en uso (*Work Product Usage*):

Presentation Name	Model Info	Entr...	Exi...	Del...	Type	Planned	Multi...
Iteración de la Fase de Elaboración [1..n]					Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Requisitos	extends 'requisitos, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Glosario	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input,Output				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input,Optional Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Prototipo de interfaz de usuario	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Actor					Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	extends 'analisis, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	extends 'diseno, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño	Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	extends 'implementacion, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Componente	Mandatory Input,Output				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de implementación	Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	extends 'prueba, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Defecto	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Evaluación de prueba	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>

Vista consolidada (*Consolidated View*):

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repea...	Multipl...
Iteración de la Fase de Elaboración [1..n]	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	1		extends 'requisitos, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Encontrar actores y casos de uso	2			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Output	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Priorizar casos de uso	3	2		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Detallar un caso de uso	4	3		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Especificador de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipar la interfaz de usuario	5	4		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de interfaz de usuario			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Optional Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipo de interfaz de usuario			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Estructurar el modelo de casos de uso	6	2,3,4		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Output	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repea...	Multipl...
▲ Análisis	7	1	extends 'análisis, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Análisis de la arquitectura	8			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar un caso de uso	9			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar una clase	10			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar un paquete	11			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repea...	Multipl...
▾ Diseño	12	7	extends 'diseno, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▾ Diseño de la arquitectura	13			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▾ Diseño de una clase	14			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▾ Diseño de un caso de uso	15			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▾ Diseño de un subsistema	16			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repea...	Multipl...
▲ Implementación	17	12	extends 'implementacion, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Implementación de la arquitectura	18			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Implementar una clase	19			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Implementar un subsistema	20			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de implementación			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Integrar el sistema	21			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrador de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Output	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Realizar prueba de unidad	22			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Mandatory Input	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repea...
Prueba	23	17	extends 'prueba, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Planificar prueba	24			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseñar prueba	25			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Desc...	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Implementar prueba	26			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Realizar pruebas de integración	27			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de integración			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Defecto			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de sistema	28			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de sistema			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descrip...	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Defecto			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Evaluar prueba	29			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Defecto			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Evaluación de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

## Estructuras de desglose del Patrón de proceso “Iteración de la Fase de Construcción [1..n]” para la iteración arquetípica de la fase de Construcción del UP.

Estructura de desglose de trabajo (*Work Breakdown Structure*):

Presentation Name	Index	Predecessors	Model Info	Type	Plann...	Repe...	Multi...	Ongo...	Event...	Optio...
Iteración de la Fase de Construcción [1..n]	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Requisitos	1		extends 'requisitos, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Encontrar actores y casos de uso	2			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Priorizar casos de uso	3	2		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Detallar un caso de uso	4	3		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipar la interfaz de usuario	5	4		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Estructurar el modelo de casos de uso	6	2,3,4		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	7	1	extends 'analisis, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Análisis de la arquitectura	8			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analizar un caso de uso	9			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analizar una clase	10			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analizar un paquete	11			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	12	7	extends 'diseno, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Diseño de la arquitectura	13			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño de una clase	14			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño de un caso de uso	15			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño de un subsistema	16			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	17	12	extends 'implementacio...	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Implementación de la arquitectura	18			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar una clase	19			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar un subsistema	20			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrar el sistema	21			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de unidad	22			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	23	17	extends 'prueba, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Planificar prueba	24			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñar prueba	25			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar prueba	26			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar pruebas de integración	27			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de sistema	28			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Evaluar prueba	29			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

### Asignación de equipos (*Team Allocation*):

Presentation Name	Model Info	Team	Type	Planned	Multi...	Optio...
Iteración de la Fase de Construcción [1..n]			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	extends 'requisitos, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Especificador de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de interfaz de usuario			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	extends 'analisis, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	extends 'diseno, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	extends 'implementacion, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrador de sistemas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	extends 'prueba, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de sistema			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de integración			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Productos de trabajo en uso (*Work Product Usage*):

Presentation Name	Model Info	Entry ...	Exit S...	De...	Type	Planned	Multi...	Optio...
Iteración de la Fase de Construcción [1..n]					Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	extends 'requisitos, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input,Output				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input,Optional Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipo de interfaz de usuario	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Actor					Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	extends 'análisis, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	extends 'diseño, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño	Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Productos de trabajo en uso (*Work Product Usage*). Continuación:

Presentation Name	Model Info	Entry ...	Exit S...	De...	Type	Planned	Multi...	Optio...
Implementation	extends 'implementacion, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente	Mandatory Input,Output				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de implementación	Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	extends 'prueba, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Defecto	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Evaluación de prueba	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*):

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repe...	Mult
Iteración de la Fase de Construcción [1..n]	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	1		extends 'requisitos, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Encontrar actores y casos de uso	2			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Priorizar casos de uso	3	2		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Detallar un caso de uso	4	3		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Especificador de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipar la interfaz de usuario	5	4		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de interfaz de usuario			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Optional Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipo de interfaz de usuario			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Estructurar el modelo de casos de uso	6	2,3,4		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repe...	Mult
▲ Análisis	7	1	extends 'analysis, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Análisis de la arquitectura	8			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar un caso de uso	9			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar una clase	10			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar un paquete	11			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description Work Breakdown Structure Team Allocation Work Product Usage Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repe...	Mult
4 Diseño	12	7	extends 'diseno, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4 Diseño de la arquitectura	13			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4 Diseño de una clase	14			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4 Diseño de un caso de uso	15			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4 Diseño de un subsistema	16			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repe...	Mult
Implementation	17	12	extends 'implementacion, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementation of the architecture	18			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar una clase	19			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar un subsistema	20			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de implementación			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrar el sistema	21			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrador de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de unidad	22			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repe...
Prueba	23	17	extends 'prueba, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Planificar prueba	24			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseñar prueba	25			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Implementar prueba	26			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Realizar pruebas de integración	27			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de integración			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Defecto			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de sistema	28			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de sistema			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Componente de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Defecto			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Evaluar prueba	29			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Defecto			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Evaluación de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>

Description Work Breakdown Structure Team Allocation Work Product Usage Consolidated View

## Estructuras de desglose del Patrón de proceso “Iteración de la Fase de Transición [1..n]” para la iteración arquetípica de la fase de Transición del UP.

Estructura de desglose de trabajo (*Work Breakdown Structure*):

Presentation Name	Index	Predecessors	Model Info	Type	Plann...	Repe...	Multi...	Ongo...	Event...	Optio...
Iteración de la Fase de Transición [1..n]	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>				
Requisitos	1		extends 'requisitos, up'	Activity	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Encontrar actores y casos de uso	2			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Priorizar casos de uso	3	2		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Detallar un caso de uso	4	3		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipar la interfaz de usuario	5	4		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Estructurar el modelo de casos de uso	6	2,3,4		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	7		extends 'analisis, up'	Activity	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Análisis de la arquitectura	8			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analizar un caso de uso	9			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analizar una clase	10			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analizar un paquete	11			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	12		extends 'diseno, up'	Activity	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Diseño de la arquitectura	13			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño de una clase	14			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño de un caso de uso	15			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño de un subsistema	16			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	17	12	extends 'implementa...	Activity	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Implementación de la arquitectura	18			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar una clase	19			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar un subsistema	20			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrar el sistema	21			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de unidad	22			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	23	17	extends 'prueba, up'	Activity	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>				
Planificar prueba	24			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñar prueba	25			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementar prueba	26			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar pruebas de integración	27			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realizar prueba de sistema	28			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Evaluar prueba	29			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description Work Breakdown Structure Team Allocation Work Product Usage Consolidated View

### Asignación de equipos (*Team Allocation*):

Presentation Name	Model Info	Team	Type	Planned	Multi...	Optio...
Iteración de la Fase de Transición [1..n]			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	extends 'requisitos, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Especificador de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de interfaz de usuario			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	extends 'analisis, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	extends 'diseno, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	extends 'implementacion, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Arquitecto			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrador de sistemas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	extends 'prueba, up'		Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Diseñador de pruebas			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de sistema			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de pruebas de integración			Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Productos de trabajo en uso (*Work Product Usage*):

Presentation Name	Model Info	Entry ...	Exit S...	Del...	Type	Planned	Multi...
Iteración de la Fase de Transición [1..n]					Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Requisitos	extends 'requisitos, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Glosario	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input,Output				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input,Optional Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Prototipo de interfaz de usuario	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Actor					Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	extends 'analisis, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio	Optional Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	extends 'diseno, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño	Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	extends 'implementacion, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Componente	Mandatory Input,Output				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de implementación	Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación	Mandatory Input,Output				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	extends 'prueba, up'				Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación	Mandatory Input				Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Plan de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso	Mandatory Input				Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Caso de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Procedimiento de prueba	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Defecto	Mandatory Input,Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de pruebas	Mandatory Input				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>
Evaluación de prueba	Output				Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>

Vista consolidada (*Consolidated View*):

Presentation Name	Index	Predece...	Model Info	Type	Planned	Repea...	Multi...
Iteración de la Fase de Transición [1..n]	0			Capability Pattern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	1		extends 'requisitos, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Encontrar actores y casos de uso	2			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Priorizar casos de uso	3	2		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Detallar un caso de uso	4	3		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Especificador de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipar la interfaz de usuario	5	4		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseñador de interfaz de usuario			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Optional Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prototipo de interfaz de usuario			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Estructurar el modelo de casos de uso	6	2,3,4		Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analista de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Caso de uso			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Glosario			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predece...	Model Info	Type	Planned	Repea...	Multi...
▲ Análisis	7		extends 'analysis, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Análisis de la arquitectura	8			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar un caso de uso	9			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del dominio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo del negocio			Optional Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar una clase	10			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Analizar un paquete	11			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Paquete del análisis			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description Work Breakdown Structure Team Allocation Work Product Usage Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Precede...	Model Info	Type	Planned	Repea...	Multi...
▲ Diseño	12		extends 'diseno, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Diseño de la arquitectura	13			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Diseño de una clase	14			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase de análisis			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Diseño de un caso de uso	15			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de casos de uso			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Realización de caso de uso-diseño			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Diseño de un subsistema	16			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predece...	Model Info	Type	Planned	Repea...	Multi...
▲ Implementación	17	12	extends 'implementacion, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Implementación de la arquitectura	18			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Arquitecto			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de despliegue			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de la arquitectura (vista del modelo de despliegue)			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Implementar una clase	19			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Clase del diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Implementar un subsistema	20			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Subsistema de implementación			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Integrar el sistema	21			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Integrador de sistemas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Modelo de implementación			Output	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plan de integración de construcciones			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Realizar prueba de unidad	22			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Interfaz			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Componente			Output	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description Work Breakdown Structure Team Allocation Work Product Usage Consolidated View

Vista consolidada (*Consolidated View*). Continuación:

Presentation Name	Index	Predece...	Model Info	Type	Planned	Repea...	M
iteration_fase_transicion							
Prueba	23	17	extends 'prueba, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Planificar prueba	24			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Plan de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Diseñar prueba	25			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Descripción de arquitectura (vista del modelo de implementación)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Descripción de la arquitectura (vista del modelo de análisis)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Descripción de la arquitectura (vista del modelo de casos de uso)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Descripción de la arquitectura (vista del modelo de despliegue)			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de análisis			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de casos de uso			Mandatory Input	Deliverable Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de despliegue			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de diseño			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Plan de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Requisitos adicionales			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Caso de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Procedimiento de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Implementar prueba	26			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Ingeniero de componentes			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de pruebas			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Componente de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Realizar pruebas de integración	27			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Ingeniero de pruebas de integración			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Componente de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Procedimiento de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Defecto			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Realizar prueba de sistema	28			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Ingeniero de pruebas de sistema			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Caso de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Componente de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de implementación			Mandatory Input	Outcome Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Procedimiento de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Defecto			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Evaluar prueba	29			Task Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Diseñador de pruebas			Primary Performer	Role Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Defecto			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Modelo de pruebas			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Plan de prueba			Mandatory Input	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Evaluación de prueba			Output	Artifact Descriptor	<input type="checkbox"/>	<input type="checkbox"/>	
Description	Work Breakdown Structure	Team Allocation	Work Product Usage	Consolidated View			

## **APÉNDICE D**

**Proceso de despliegue configurado para el UP**

En este apéndice se muestra el proceso de despliegue (*Delivery Process*) configurado para el proceso del UP bajo el paquete de procesos (*Process Package*) “Plantillas de Iteración”.

Proceso de despliegue (*Delivery Process*) configurado para el UP.

Presentation Name	Index	Predecessors	Model Info	Type	Planned	Repeata...	Multi...	Ongo...	Event...	Optio...
▲ Ciclo de Vida	0			Delivery Process	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Fase de Inicio	1			Phase	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Iteración de la Fase de Inicio [1..n]	2		extends 'iteracion_fase_inicio, up'	Iteration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	3		extends 'requisitos, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	9	3	extends 'analisis, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	14	9	extends 'diseno, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	19		extends 'implementacion, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	25		extends 'prueba, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Hito de los Objetivos del Ciclo de Vida	32	2		Milestone	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Fase de Elaboración	33	32		Phase	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Iteración de la Fase de Elaboración [1..n]	34	32	extends 'iteracion_fase_elaboracion, up'	Iteration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	35		extends 'requisitos, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	41	35	extends 'analisis, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	46	41	extends 'diseno, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	51	46	extends 'implementacion, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	57	51	extends 'prueba, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Hito de la Arquitectura del Ciclo de Vida	64	34		Milestone	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Fase de Construcción	65	64		Phase	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Iteración de la Fase de Construcción [1..n]	66	64	extends 'iteracion_fase_construccion, up'	Iteration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	67		extends 'requisitos, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Análisis	73	67	extends 'analisis, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Diseño	78	73	extends 'diseno, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Implementación	83	78	extends 'implementacion, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Prueba	89	83	extends 'prueba, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Hito de la Funcionalidad Operativa Inicial	96	66		Milestone	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Fase de Transición	97	96		Phase	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
▲ Iteración de la Fase de Transición [1..n]	98	96	extends 'iteracion_fase_transicion, up'	Iteration	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Requisitos	99		extends 'requisitos, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Análisis	105		extends 'analisis, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Diseño	110		extends 'diseno, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Implementación	115	110	extends 'implementacion, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Prueba	121	115	extends 'prueba, up'	Activity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Hito de la Versión del Producto	128	98		Milestone	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Description | Work Breakdown Structure | Team Allocation | Work Product Usage | Consolidated View

## **ANEXOS**

## **ANEXO A**

**Términos e íconos de SPEM y *EPF Composer***

Entre corchetes se indican las partes de un término opcionales. Entre paréntesis se representan los términos alternativos.

Término (Castellano)	Término (Inglés)	Ícono
Actividad	<i>Activity</i>	
Activo Reutilizable	<i>Reusable Asset</i>	
Artefacto	<i>Artifact</i>	
Biblioteca [de Métodos] (Repositorio)	<i>[Method] Library</i>	
Categoría	<i>Category</i>	
Categoría Estándar	<i>Standard Category</i>	
Categoría Personalizada	<i>Custom Category</i>	
Clase de Producto de Trabajo	<i>Work Product Kind</i>	
Componente de Entregable	<i>Deliverable Component</i>	
Componente de Proceso	<i>Process Component</i>	
Concepto	<i>Concept</i>	
Configuración [de Método]	<i>[Method] Configuration</i>	
Conjunto de Roles	<i>Role Set</i>	
Consideraciones para el Cálculo	<i>Estimation Considerations</i>	
Contenido de Método	<i>Method Content</i>	
Definición de Término	<i>Term Definition</i>	
Definición de Trabajo	<i>Work Definition</i>	
Descripción de Contenido	<i>Content Description</i>	
Diagrama de Actividad	<i>Activity Diagram</i>	
Diagrama de Dependencias de	<i>Work Product Dependency</i>	

Producto de Trabajo	<i>Diagram</i>	
Diagrama de Detalle de Actividad	<i>Activity Detail Diagram</i>	
Directriz	<i>Guideline</i>	
Disciplina	<i>Discipline</i>	
Documentación	<i>Whitepaper</i>	
Dominio	<i>Domain</i>	
Ejemplo	<i>Example</i>	
Elemento de Contenido	<i>Content Element</i>	
Elemento de Desglose (de Descomposición)	<i>Breakdown Element</i>	
Elemento de Desglose de Trabajo	<i>Work Breakdown Element</i>	
Elemento de Método	<i>Method Element</i>	
Elemento de Proceso	<i>Process Element</i>	
Elemento de Variabilidad	<i>Variability Element</i>	
Elemento Describible	<i>Describable Element</i>	
Entregable	<i>Deliverable</i>	
Estructura de Asignación de Equipos (de Personal)	<i>Team Allocation Structure</i>	
Estructura de Desglose de Trabajo	<i>Work Breakdown Structure (WBS)</i>	
Estructura de Utilización de Productos de Trabajo	<i>Work Product Usage Structure</i>	
Fase	<i>Phase</i>	
Grupo de Disciplinas	<i>Discipline Group</i>	

Guía (Instrucción)	<i>Guidance</i>	
--------------------	-----------------	---

Guía de Herramienta	<i>Tool Mentor</i>	
Guía para la Estimación (Consideraciones para el Cálculo)	<i>Estimation Considerations</i>	
Herramienta	<i>Tool</i>	
Hito (Objetivo)	<i>Milestone</i>	
Hoja de Ruta (Mapa)	<i>Roadmap</i>	
Informe	<i>Report</i>	
Iteración	<i>Iteration</i>	
Lista de Comprobación	<i>Checklist</i>	
Material de Soporte	<i>Supporting Material</i>	
Paquete de Categorías	<i>Category Package</i>	
Paquete de Contenido	<i>Content Package</i>	
Paquete de Métodos	<i>Method Package</i>	
Paquete de Proceso	<i>Process Package</i>	
Parámetro de Proceso	<i>Process Parameter</i>	
Participante [de Proceso]	<i>[Process] Participant</i>	
Paso	<i>Step</i>	
Patrón de Proceso (patrón de capacidad)	<i>Capability pattern</i>	
Plantilla	<i>Template</i>	
Plantilla de Planificación de Proceso	<i>Process Planning Template</i>	

<i>Plug-in</i> [de Método]	<i>[Method] Plug-in</i>	
Práctica	<i>Practice</i>	
Proceso	<i>Process</i>	
Proceso para Despliegue (de entrega)	<i>Delivery Process</i>	
Producto de Trabajo [Definición de]	<i>Work Product [Definition]</i>	
Producto de Trabajo en Uso (Descriptor de Producto de Trabajo)	<i>Work Product Use (Work Product Descriptor)</i>	
Realizador [de Proceso]	<i>[Process] Performer</i>	
Resultado	<i>Outcome</i>	
Rol [Definición de]	<i>Role [Definition]</i>	
Rol Compuesto (Equipo)	<i>Composite Role</i>	
Rol en Uso (Descriptor de Rol)	<i>Role Use (Role Descriptor)</i>	
Sección	<i>Section</i>	
Secuencia de Trabajo	<i>Work Sequence</i>	
Tarea [Definición de]	<i>Task [Definition]</i>	
Tarea en Uso (Descriptor de Tarea)	<i>Task Use (Task Descriptor)</i>	
Tipo de Reutilización de Actividad	<i>Activity Use Kind</i>	

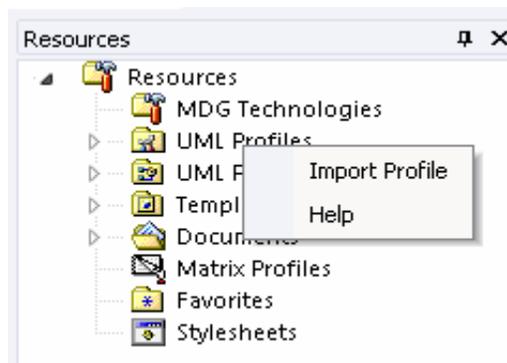
## **ANEXO B**

### **Trabajando con perfiles UML en Enterprise Architect**

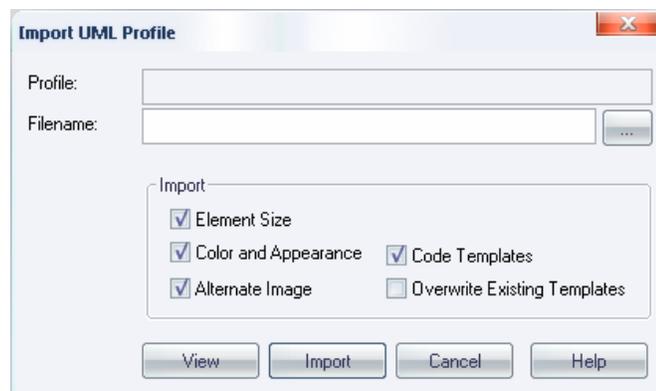
## IMPORTANDO UN PERFIL UML

Para importar un perfil necesitará un archivo XML adecuado. Si el perfil incluye referencias a varios meta-archivos, ellos deberían estar en el mismo directorio del perfil XML.

1. Haga clic sobre el árbol de perfiles UML (*UML Profiles*) en la ventana de Recursos (*Resources*) y seleccione Importar Perfil (*Import Profile*) desde el menú contextual (como se muestra en la figura).



2. Se abrirá la ventana Importar Perfil UML (*Import UML Profile*).

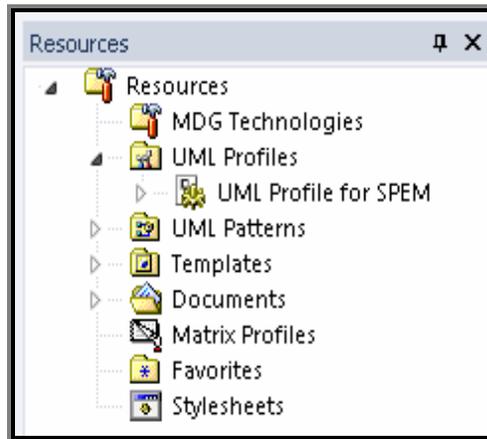


3. Ubique el archivo de Perfil XML a importar usando el botón Explorador [...]

4. Configure las opciones necesarias para importar para todos los estereotipos definidos en el perfil - puede seleccionar para importar:
  - Tamaño del elemento si/no - marque esto para importar el tamaño del elemento.
  - Color y apariencia si/no - marque esto para importar el color (fondo, bordes y fuente) y apariencia.
  - Imágenes alternativas si/no - marque esto para importar una imagen de meta-archivo.
  - Plantillas de Código si/no - marque esto para importar las plantillas de código si es que existen.
  - Sobrescribir Plantillas existentes si/no - marque esto para sobre-escribir cualquier plantilla de código existente definida en el proyecto actual.
5. Presione Importar.

### **Usando los elementos del perfil**

La imagen siguiente muestra las ramas abiertas de los perfiles para el esquema XSD y los elementos UML estereotipados disponibles.



Puede utilizar los elementos del perfil de la siguiente forma:

- Los elementos tales como clases e interfaces se pueden arrastrar directamente desde la ventana de recursos al diagrama actual.
- Los atributos se pueden arrastrar sobre un elemento receptor (por ejemplo una clase). Se agregarán automáticamente a la lista de características del elemento.
- Las operaciones son como los atributos -arrastre sobre un elemento receptor para agregar la operación.
- Los vínculos tales como asociaciones, generalización y dependencia se agregan al seleccionarlos en el navegador, hacer clic en el objeto de inicio en un diagrama y sin soltar arrastrar hasta el objeto final (en la misma forma como se agregan vínculos normales). El vínculo se agregará con la nueva información de estereotipo y valor etiquetado.
- Los extremos de las asociaciones se pueden agregar arrastrando el elemento final del vínculo sobre el extremo de la asociación en el diagrama.

### **Eliminando un perfil**

Para eliminar un perfil, haga clic con el botón derecho en el perfil a eliminar y seleccione la opción Eliminar Perfil del menú contextual. Tenga en cuenta que esto no afectará a los elementos que utilicen este perfil ya definidos. Si un estereotipo que se importó utilizando el perfil está en uso, no se eliminará del modelo cuando elimine el perfil.

**Cargando un perfil nuevamente**

Para cargar un perfil nuevamente, elimine el perfil como se indica en el punto anterior e impórtelo de nuevo.

## **ANEXO C**

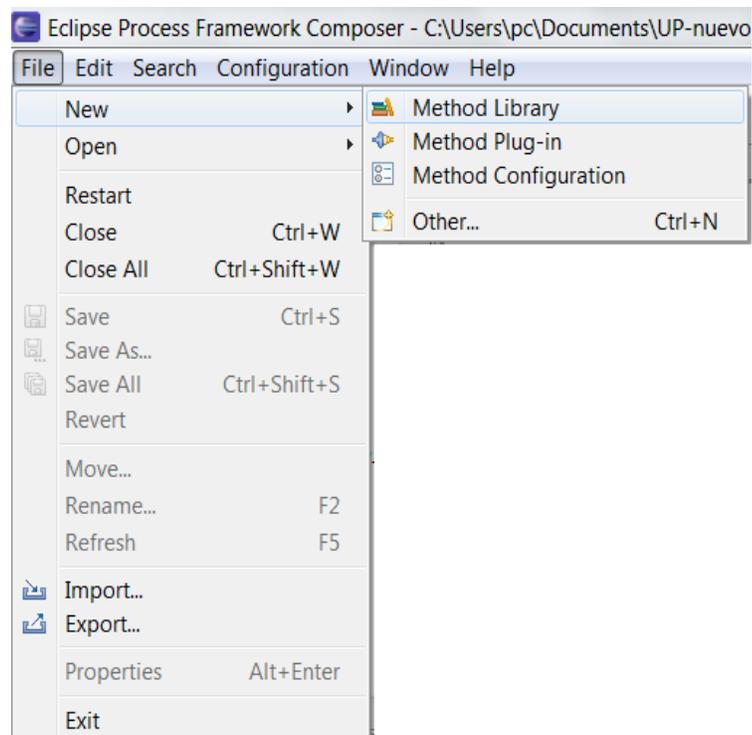
**Pasos recomendados para la especificación de procesos con *EPF***

***Composer***

Existen diferentes opciones a la hora de definir métodos, metodologías y/o procesos de desarrollo de software con SPEM y *EPF Composer*, cada una con sus ventajas y desventajas, especialmente en cuanto a la mantenibilidad, reusabilidad y escalabilidad del Método definido. A continuación se explica la manera en que se aconseja crear *plug-ins* desde cero con EPF Composer, atendiendo a que dichas propiedades de mantenibilidad, reusabilidad y escalabilidad sean las mejores posibles.

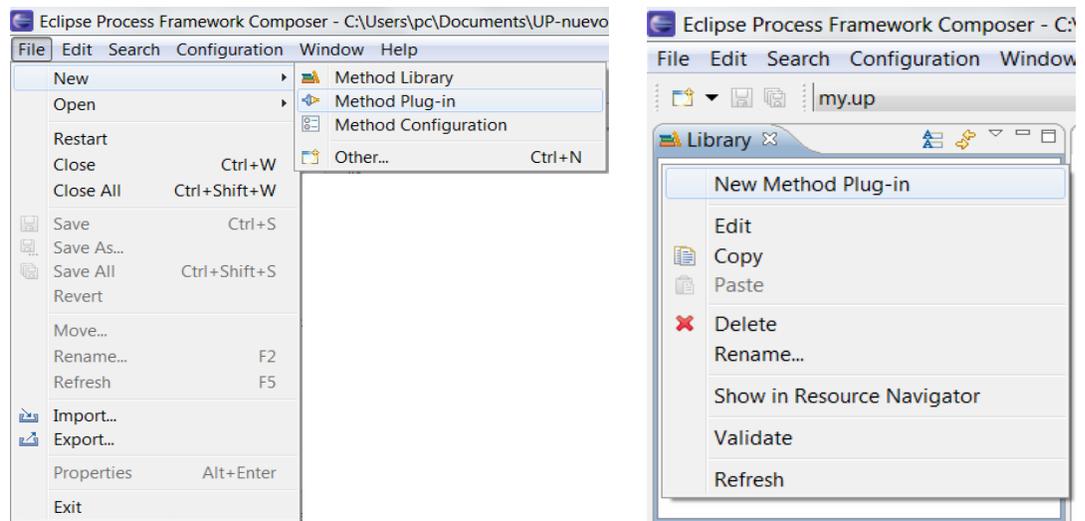
## PASO 1

El primer paso es crear una biblioteca de métodos (*File->New->Method Library*) que albergue todos los *plug-ins* y configuraciones de la metodología que va a definirse.



## PASO 2

El segundo paso es crear un *plug-in* de método (*File->New->Method Plug-in*, o clic derecho en la vista *Library->New Method Plug-in*). La estructuración mediante *plug-ins* facilita la modularización y la organización del contenido. Un *plug-in* contendrá los elementos necesarios para la definición de un proceso (o varios si tienen una relación estrecha).

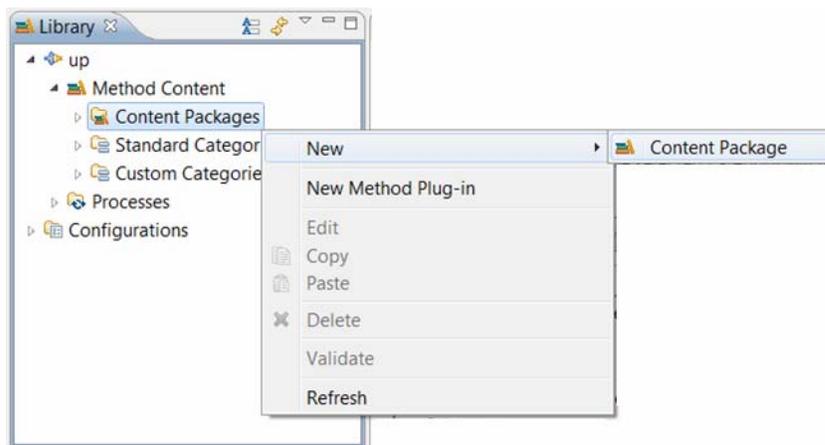


## PASO 3

El siguiente paso es crear los elementos de método (guías, productos de trabajo, roles y tareas). Se aboga por crear primeramente los elementos de método que se van a utilizar posteriormente al definir los procesos, ya que la reusabilidad y mantenibilidad mejoran notablemente. Dichos elementos se podrán reutilizar tantas veces como sea necesario en los procesos, y si hay que hacer cambios en dichos elementos habrá que hacerlos únicamente en un sitio (en la definición del elemento de método), y no en cada una de las ocurrencias del elemento en los procesos.

### Subpaso 3.1

Para crear elementos de método, previamente se deben crear paquetes de contenido (como mínimo uno) que servirán para organizar el contenido del *plug-in*, creando jerarquías de paquetes. Para crear un paquete de contenido se hace clic derecho sobre *Content Packages*->*New*->*Content Package* (para acceder a *Content Packages* en la vista *Library*: nombre\_del\_plug-in->*Method content*->*Content Packages*). También se pueden crear paquetes de contenido dentro de otros paquetes de contenido. Dentro de cada paquete de contenido se pueden crear los elementos de método de cualquier tipo que se consideren oportunos.



### Subpaso 3.2

En cada paquete se crean de forma automática cuatro carpetas para almacenar las definiciones de cada tipo de elemento. Para crear un elemento de método se hace clic derecho sobre una de dichas carpetas (la del tipo que nos interese)->*New*->*Tipo\_del\_elemento*.

El orden en la creación de los elementos de método no es relevante, aunque por comodidad se aconseja seguir el siguiente orden:

5. Crear las guías, ya que pueden ser referenciadas desde cualquier otro elemento de método.
6. Crear los productos de trabajo. Al crear cada producto, asociarle las guías convenientes.
7. Crear los roles. Al crear cada rol, indicar los productos de trabajo de los que son responsables y asociarle las guías convenientes.
8. Crear las tareas. Al crear cada tarea, indicar los roles que participan en la tarea, los productos de trabajo que son entrada o salida de la tarea y las guías convenientes.

La ventaja de respetar este orden es que cuando se crea un elemento ya existen todos los elementos con los que está relacionado y, por tanto, su edición es más cómoda. Las descripciones de los elementos de método y sus relaciones con otros elementos se indican en las distintas pestañas de la vista de edición de cada elemento.

#### **PASO 4**

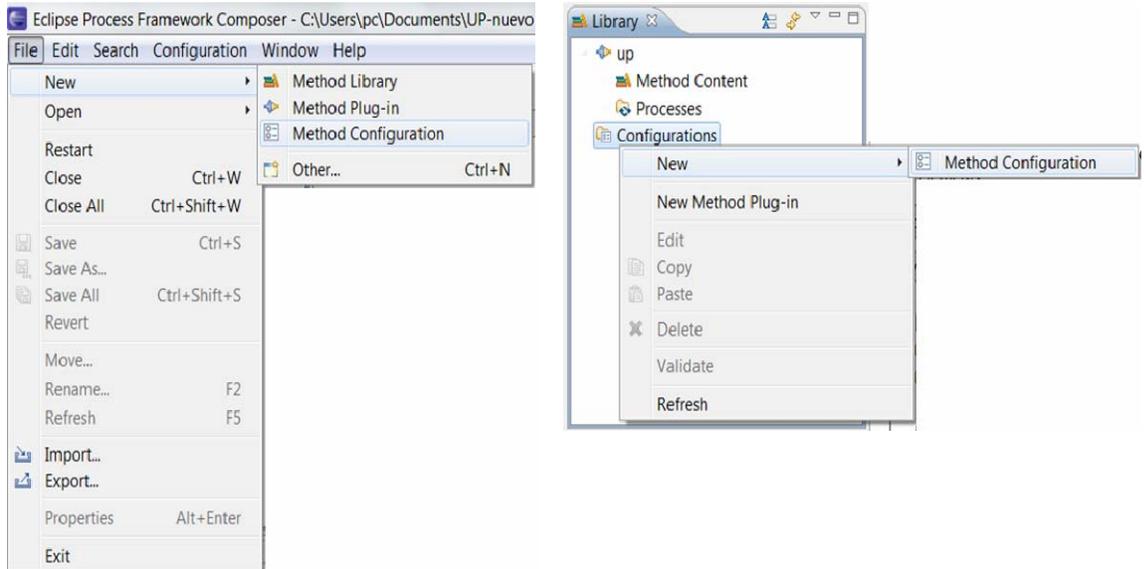
Una vez se han creado los elementos de método, el siguiente paso es crear las categorías estándar, las cuales permiten categorizar los elementos de método creados. Existen diferentes tipos de categorías estándar: disciplinas para categorizar tareas, dominios y clases de productos de trabajo para categorizar productos de trabajo, conjuntos de roles para categorizar roles y herramientas para categorizar guías de herramienta (un tipo específico de guía) de una herramienta específica. Además, las disciplinas y los conjuntos de roles se pueden agrupar mediante agrupaciones de disciplinas y agrupamientos de conjuntos de roles, respectivamente.

Para crear una categoría estándar se utiliza el método habitual, clic derecho sobre el tipo de categoría que se quiere crear->*New*->*Tipo\_de\_categoria* (para acceder a las categorías estándar en la vista *Library*, nombre\_del\_plug-in->*Method Content*->*Standard Categories*). La asignación de elementos a una categoría se hace desde una pestaña específica de la vista de edición de la categoría.

## **PASO 5**

Una vez que se han creado los elementos de método y las categorías estándar, y antes de poder crear un proceso que hace uso de dichos elementos (sea patrón de proceso o proceso para despliegue) se debe crear una configuración. Esto es necesario porque cada elemento de proceso debe especificar su configuración (o configuraciones, ya que un proceso puede estar asociado con varias configuraciones). Las configuraciones son subconjuntos de la biblioteca de métodos que nos permiten restringir la vista de todos los elementos de la biblioteca a únicamente el conjunto de elementos que nos interese.

Para crear una configuración, *File*->*New*->*Method Configuration*, o desde la vista *Library*: clic derecho sobre *Configurations*->*New*->*Method Configuration*. La selección de elementos de la configuración se hace desde la pestaña *Plug-in and Package Selection* y de la vista de edición de la configuración.



## PASO 6

Una vez que se ha creado al menos una configuración, se pueden empezar a crear los patrones de procesos. A la hora de crear procesos se recomienda utilizar un enfoque ascendente, ya que facilita la reutilización de los procesos definidos.

### Subpaso 6.1

Para aplicar dicho enfoque ascendente, lo primero es crear los patrones de proceso básicos en los que se reutilizan las tareas definidas en el contenido de método. Para crear un patrón de proceso desde la vista *Library*: hacer clic derecho sobre *Capability Patterns*->*New*->*Capability Pattern* (se accede a Patrones de proceso desde la vista *Library*: nombre\_del\_plug-in->*Processes*). La descripción y la estructura del proceso se definen en las distintas pestañas de la vista de edición del proceso. También cabe señalar que los procesos se pueden organizar y modularizar mediante paquetes de

proceso.

### **Subpaso 6.2**

Una vez que se ha definido la estructura del proceso, se pueden definir los diagramas de actividad, de detalle de actividad y de dependencias de productos de trabajo para cada patrón de proceso.

### **Subpaso 6.3**

A partir de los patrones básicos se pueden crear patrones de proceso más complejos. En estos patrones se reutilizan los patrones creados con anterioridad. De esta forma se pueden crear jerarquías de patrones que reflejen la estructura de fases, iteraciones, actividades, etc. de un proceso.

A la hora de aplicar un patrón en otro existen dos opciones: copiarlo o ampliarlo. Si el patrón que se va a aplicar no se piensa personalizar dentro del patrón aplicado, se recomienda utilizar la opción ampliar, ya que resulta mucho más fácil de mantener. Las modificaciones posteriores sólo hace falta hacerlas en el patrón original, y los cambios se reflejarán en todos los patrones donde se aplicó dicho patrón mediante ampliación. Sin embargo, si el patrón debe personalizarse dentro del proceso en que se aplica, la aplicación mediante ampliación no permite hacerlo (permite añadir nuevos elementos, pero no eliminar elementos existentes que no sean útiles en el contexto en el que se aplica el patrón), por lo que habría que aplicarlo mediante copia. Un patrón copiado sí puede personalizarse para adaptarse al contexto en el que se aplica, sin embargo, su mantenibilidad disminuye, ya que los cambios posteriores deben hacerse sobre todas las copias manualmente.

Una vez se han aplicado los patrones se pueden definir los diagramas para

el patrón de proceso creado según se indica en el subpaso 6.2.

## PASO 7

Una vez que se han creado los patrones de proceso necesarios se pueden crear los procesos para despliegue reutilizando dichos patrones de proceso. La reutilización de patrones de proceso en procesos para despliegue se hace de igual forma que la reutilización de patrones de proceso dentro de otros patrones comentada en el punto anterior. También se pueden definir diagramas para el proceso para despliegue.

Con lo indicado en los pasos 6 y 7 se puede implementar una jerarquía tarea-actividad- subproceso-proceso (típica de los estándares ISO) de la siguiente forma: Se crearían los patrones básicos de las actividades que sólo incluyen tareas (subpaso 6.1). Después se crearían los patrones de los subprocesos en los que se reutilizan los patrones de las actividades (subpaso 6.3). Finalmente se crearían los procesos como procesos para despliegue en los que se reutilizan los patrones de los subprocesos (paso 7).

## PASO 8

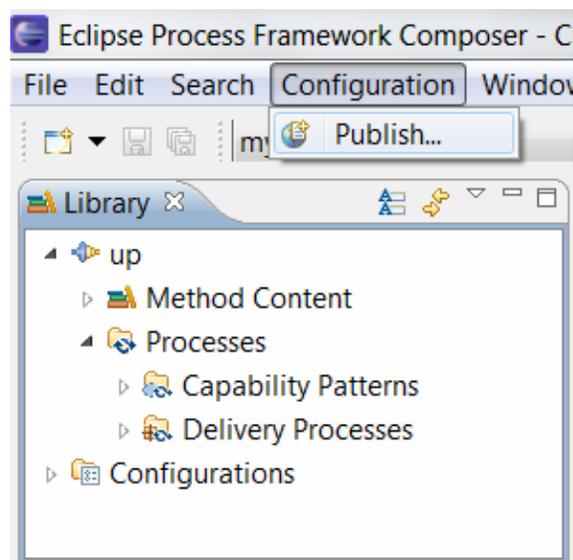
Una vez que tenemos todos los procesos y todos los elementos de método definidos podemos publicarlos. Como se ha comentado en el apartado de crear una configuración, cuando se publica un método lo que realmente se hace es publicar una configuración de método, la cual puede corresponderse con el contenido de toda la biblioteca, el de un *plug-in* o cualquier otro subconjunto de elementos de la biblioteca que sea conveniente. Por ello, antes de publicar la configuración hay que definir la vista de la configuración (o vistas, ya que se pueden definir varias). Una vista de la configuración es una estructura jerárquica que sirve para navegar por los contenidos publicados de forma más accesible, ya que dicha estructura se publica como

un árbol de navegación (menú desplegable que aparece a la izquierda) en el sitio web generado en la

publicación. Los contenidos de dicha vista se pueden personalizar completamente, ya que las vistas se definen mediante categorías personalizadas, las cuales pueden contener los elementos que queramos con la estructura jerárquica que queramos.

## PASO 9

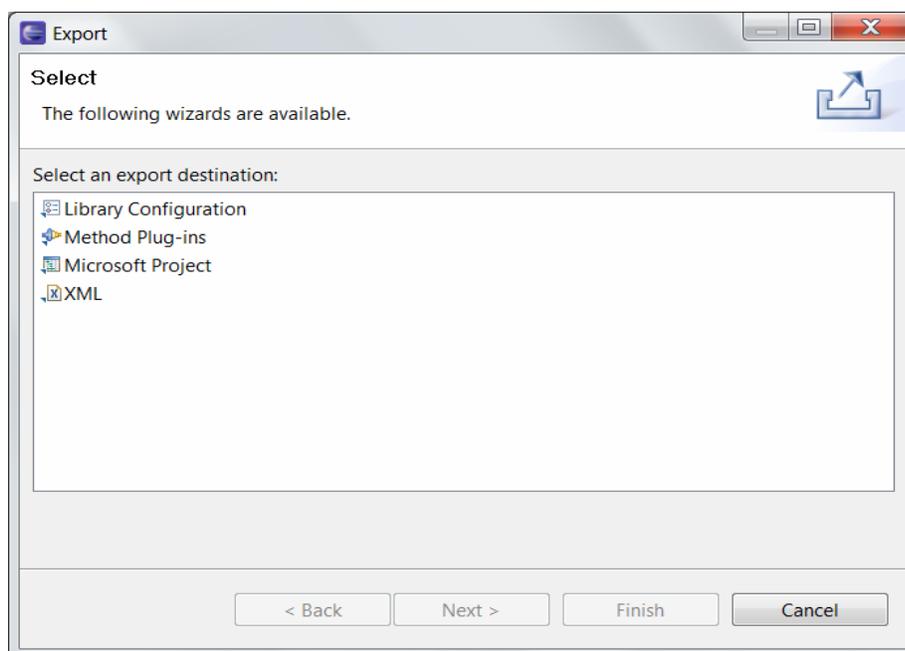
Una vez se ha definido la vista (o vistas) de la configuración se puede publicar la configuración.



## **ANEXO D**

### **Opciones de Exportación en EPF Composer**

EPF Composer ofrece varias opciones para exportar el contenido (*File->Export*) definido con la herramienta, las cuales tienen diferentes finalidades. En la siguiente figura se muestran las opciones de exportación que se comentarán a continuación.



### D.1: Exportar Configuraciones y *Plug-ins* de Método

Mediante las opciones de exportación *Library Configuration* y *Method Plug-ins* se pueden exportar configuraciones y *plug-ins* respectivamente, para posteriormente importarlos desde otra biblioteca de método. Esta opción es útil cuando una biblioteca necesita reutilizar elementos definidos en otra biblioteca.

Puesto que los elementos de una biblioteca no pueden hacer referencia a los elementos de otra biblioteca, la solución pasa por exportar los elementos

necesarios y posteriormente importarlos en la biblioteca que los necesite.  
Hay que tener en cuenta que al exportar se crea una copia del contenido, y  
por

tanto, los cambios posteriores a la exportación sobre dichos elementos de la biblioteca original no se reflejarán en las bibliotecas que los hayan importado.

La mayoría de las veces lo que interesa es exportar *plug-ins* de método, pero la opción de exportación de configuraciones permite definir con la granularidad deseada la selección de elementos que se exportarán. Esta granularidad puede ir desde unos pocos elementos hasta una biblioteca de métodos entera.

## **D.2: Exportar a Microsoft Project**

Mediante la opción de exportación *Microsoft Project* se pueden exportar procesos a un formato procesable por la herramienta *Microsoft Project*®. En este apartado se explica la relación de esta opción con las Plantillas para Planificación de Procesos (*Process Planning Templates*) de SPEM y cómo se realiza la exportación.

*EPF Composer* implementa ampliamente el estándar SPEM 2, aunque no al cien por cien. Uno de los elementos del estándar SPEM que *EPF Composer* no implementa son las Plantillas para Planificación de Procesos. SPEM define las Plantillas para Planificación de Procesos como un tipo especial de Proceso, el cual se ha preparado para su instanciación en una herramienta de planificación de proyectos. Aunque *EPF Composer* no permite definir Plantillas para Planificación de Procesos, sí permite exportar los otros dos tipos de Procesos (Patrones de Proceso y Procesos para Despliegue) a herramientas de planificación de proyectos, en concreto a *Microsoft Project*®.

*EPF Composer* no permite añadir información de instanciación de la

realización (fecha de comienzo, fecha de fin, duración, etc.) a los elementos de un proceso. Lo que sí se puede hacer, de interés para los planes de proyectos, es definir las relaciones de precedencia entre elementos y el tipo de dependencia. Estas

opciones son muy útiles a la hora de exportar el proceso a una herramienta de planificación de proyectos.

Antes de exportar un proceso a un proyecto de *Microsoft Project*® hay que comprobar en la pestaña *Work Breakdown Structure* de la vista de edición del proceso que los elementos que queremos exportar tienen activada la casilla *Planned*. También se debe comprobar que los tipos de dependencias entre los elementos de desglose son los adecuados. Cuando se crea una dependencia de precedencia entre dos elementos de desglose, el tipo de dependencia por defecto es *Finish to Start* (Finalización a inicio). El tipo de dependencia de un elemento se puede modificar en la pestaña *General* de la vista *Properties* cuando se tiene seleccionado dicho elemento en la vista de edición del proceso.

Para exportar se siguen los siguientes pasos:

1. Seleccionar la opción *File->Export*.
2. Seleccionar *Microsoft Project*.
3. Seleccionar el proceso (patrón de proceso o proceso para despliegue) que se quiere exportar.
4. Seleccionar la configuración asociada con el proceso con la que se quiere hacer la exportación, ya que un proceso puede tener varias configuraciones asociadas. En función de la configuración seleccionada, habrá elementos que no se exporten (los que no pertenezcan a la configuración seleccionada). Al exportar el proceso existe la opción de publicar el sitio web del proceso, publicándose todos los elementos de la configuración o sólo aquellos relacionados con el proceso. También existe una opción para ignorar la propiedad *Planned*, definida en la estructura de desglose de trabajo, y así exportar todos los elementos del proceso, tengan activada o no dicha

propiedad.

5. Si se ha seleccionado la opción para publicar el sitio web del proceso, seleccionar las opciones de publicación del mismo. Dichas opciones son las mismas que cuando se publica una configuración normalmente.
6. Seleccionar el nombre y el directorio del archivo generado.

En la exportación se genera un archivo XML que se puede abrir con *Microsoft Project*®.

El archivo generado contiene la estructura de tareas y actividades (tareas resumen) definidas en el proceso exportado, es decir, la WBS. Las dependencias entre tareas también se reflejan en el archivo XML generado. Los roles se exportan como recursos de *Microsoft Project*®. Dichos roles se conectan con las tareas o tareas resumen correspondientes. En el caso de elementos no exportados (al no marcar su propiedad *Planned*), los roles definidos se asociarán a la tarea o tarea resumen correspondiente al elemento padre de dichos elementos no exportados. Las fechas del proyecto serán unas fechas por defecto que deberán modificarse desde *Microsoft Project*®, así como el resto de elementos del plan de proyecto que no se pudieron definir y exportar desde *EPF Composer*, como los recursos no humanos, costes, esfuerzos, etc.

### **D.3: Exportar a XML**

*EPF Composer* ofrece la posibilidad de exportar a formato XML uno o varios *plug-ins* o una biblioteca de métodos entera. El archivo XML generado contendrá toda la información de los *plug-ins* exportados, excepto los diagramas de los procesos, los cuales se mantendrán como archivos XMI con la estructura de directorios habitual en las bibliotecas de métodos de

*EPF Composer.*

Con la exportación a XML se consigue tener toda la información de varios *plug-ins* o una biblioteca de métodos en un único archivo (sin tener en cuenta los diagramas y otros recursos complementarios como imágenes, archivos, etc.). Hay que tener en cuenta que si se exporta una biblioteca de métodos con muchos *plug-ins* y muchos elementos, el archivo XML resultante puede ser muy grande.

Al igual que mediante las formas de exportación comentadas antes (ver apartado **D.1: Exportar Configuraciones y *Plug-ins* de Método**), los elementos exportados en un archivo XML se pueden importar posteriormente desde una biblioteca de métodos.

El archivo XML generado sigue un esquema XSD. Dicho XSD puede descargarse desde [http://www.eclipse.org/epf/tool\\_component/xml\\_schema.php](http://www.eclipse.org/epf/tool_component/xml_schema.php)

Al usar un XSD/XML estándar, otras herramientas que permiten trabajar con esquemas XML, también podrán leer y procesar los archivos XML exportados.

## HOJA DE METADATOS

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 1/6

<b>Título</b>	Especificación De Procesos De Desarrollo De Software A Través Del Metamodelo De Ingeniería De Procesos De Software (SPEM)
---------------	---

### Autor(es)

<b>Apellidos y Nombres</b>	<b>Código CVLAC / e-mail</b>	
Vivenes R., Charli J.	<b>CVLAC</b>	<b>V-17.672.978</b>
	<b>e-mail</b>	cjvrinf@gmail.com
	<b>e-mail</b>	

### Palabras o frases claves:

Procesos de Desarrollo de Software

Lenguajes de Modelado de Procesos

Metamodelo de Ingeniería de Procesos de Software

## Hoja de Metadatos para Tesis y Trabajos de Ascenso – 2/6

### Líneas y sublíneas de investigación:

Área	Subárea
Ciencias	Informática

### Resumen (abstract):

Con el desarrollo de este trabajo se logró sistematizar una experiencia en la aplicación de principios y tecnologías inherentes a la Ingeniería de Software (IS) en el contexto del Metamodelo de Ingeniería de Procesos de Software (SPEM, *Software Process Engineering Meta-model*) a través de su implementación en el Proceso Unificado de Desarrollo de Software (UP, *Unified Process*), introducido como caso de estudio que permitió ensayar una alternativa válida para la definición, especificación y mantenimiento de métodos, metodologías y/o procesos de desarrollo de software impulsados como resultados provenientes de proyectos de investigación que desarrollan miembros del Centro de Ingeniería de Software del Programa de la Licenciatura en Informática del Núcleo de Sucre de la Universidad de Oriente; contribuyendo además al fortalecimiento de las actividades de docencia e investigación en la academia. Para lograr este objetivo se adaptó una propuesta metodológica para la investigación en IS, que plantea cinco etapas, incorporando otros métodos y tareas necesarias y combinando éstos con técnicas, modelos, lenguajes y herramientas de software para enriquecer y complementar los resultados de este trabajo. En la etapa de búsqueda de la documentación se obtuvo el basamento teórico, sistematizado en el marco referencial que sustenta este trabajo; la etapa de determinación del problema permitió precisar concretamente el universo de discurso para la investigación, representado a través de un modelo de dominio expresado en UML; en la etapa de creación de la hipótesis se especificó en lenguaje natural las características deseables de la especificación del caso de estudio elegido que se implementó a través de los elementos conceptuales del SPEM; en la etapa de definición del método de trabajo se establecieron cada una de las actividades que permitirían lograr la especificación del caso de estudio haciendo uso de la experiencia adquirida y la creatividad como herramientas que permitieron configurar todo el proceso de investigación; finalmente, la etapa de resolución, verificación y validación consistió en la puesta en práctica de las actividades anteriormente establecidas que permitieron especificar, verificando y validando, los requisitos impuestos en la etapa de creación de la hipótesis.

## Hoja de Metadatos para Tesis y Trabajos de Ascenso – 3/6

### Contribuidores:

Apellidos y Nombres	ROL / Código CVLAC / e-mail	
Acuña, Leopoldo	ROL	CA <input type="checkbox"/> AS <input checked="" type="checkbox"/> TU <input type="checkbox"/> JU <input type="checkbox"/>
	CVLAC	V-9.976.457
	e-mail	scglas@hotmail.com
	e-mail	
Galantón, Alejandra	ROL	CA <input type="checkbox"/> AS <input type="checkbox"/> TU <input type="checkbox"/> JU <input type="checkbox"/> <input checked="" type="checkbox"/>
	CVLAC	V-11.383.261
	e-mail	agalanto@gmail.com
	e-mail	
Hamana, Manuel	ROL	CA <input type="checkbox"/> AS <input type="checkbox"/> TU <input type="checkbox"/> JU <input type="checkbox"/> <input checked="" type="checkbox"/>
	CVLAC	V-14.284.408
	e-mail	hamanammanuel@gmail.com
	e-mail	

### Fecha de discusión y aprobación:

Año Mes Día

2012	01	13
------	----	----

Lenguaje: SPA

## Hoja de Metadatos para Tesis y Trabajos de Ascenso – 4/6

### Archivo(s):

Nombre de archivo	Tipo MIME
Tesis-VivenesCharli.doc	Application/Word

### Alcance:

**Espacial:** (Opcional) \_\_\_\_\_

**Temporal:** (Opcional) \_\_\_\_\_

**Título o Grado asociado con el trabajo:** Licenciatura en Informática

**Nivel Asociado con el Trabajo:** Licenciado

**Área de Estudio:** Informática - Ingeniería de Software

**Institución(es) que garantiza(n) el Título o grado:**

Universidad de Oriente

# Hoja de Metadatos para Tesis y Trabajos de Ascenso – 5/6



UNIVERSIDAD DE ORIENTE  
CONSEJO UNIVERSITARIO  
RECTORADO

CU N° 0975

Cumaná, 04 AGO 2009

Ciudadano  
**Prof. JESÚS MARTÍNEZ YÉPEZ**  
Vicerrector Académico  
Universidad de Oriente  
Su Despacho

Estimado Profesor Martínez:

Cumplo en notificarle que el Consejo Universitario, en Reunión Ordinaria celebrada en Centro de Convenciones de Cantaura, los días 28 y 29 de julio de 2009, conoció el punto de agenda **"SOLICITUD DE AUTORIZACIÓN PARA PUBLICAR TODA LA PRODUCCIÓN INTELECTUAL DE LA UNIVERSIDAD DE ORIENTE EN EL REPOSITORIO INSTITUCIONAL DE LA UDO, SEGÚN VRAC N° 696/2009"**.

Leído el oficio SIBI – 139/2009 de fecha 09-07-2009, suscrita por el Dr. Abul K. Bashirullah, Director de Bibliotecas, este Cuerpo Colegiado decidió, por unanimidad, autorizar la publicación de toda la producción intelectual de la Universidad de Oriente en el Repositorio en cuestión.

UNIVERSIDAD DE ORIENTE  
SISTEMA DE BIBLIOTECA  
RECIBIDO POR *[Firma]*  
FECHA 5/8/09 HORA 5:30

Comunicación que hago a usted a los fines consiguientes.

Cordialmente,

*[Firma]*  
**JUAN A. BOLANOS CUNVELO**  
Secretario



C.C: Rectora, Vicerrectora Administrativa, Decanos de los Núcleos, Coordinador General de Administración, Director de Personal, Dirección de Finanzas, Dirección de Presupuesto, Contraloría Interna, Consultoría Jurídica, Director de Bibliotecas, Dirección de Publicaciones, Dirección de Computación, Coordinación de Teleinformática, Coordinación General de Postgrado.

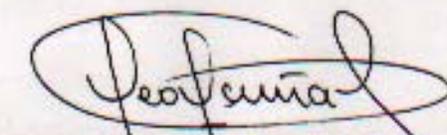
JABC/YGC/maruja

## Hoja de Metadatos para Tesis y Trabajos de Ascenso- 6/6

**Artículo 41 del REGLAMENTO DE TRABAJO DE PREGRADO (vigente a partir del II Semestre 2009, según comunicación CU-034-2009) :** "los Trabajos de Grado son de la exclusiva propiedad de la Universidad de Oriente, y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quien deberá participarlo previamente al Consejo Universitario para su autorización".



**Charli José Vivenes Rengel**  
Autor



**Leopoldo Acuña**  
Asesor