

# UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras de Software Basado en el Proceso Unificado



Universidad  
del Cauca

Monografía de trabajo de grado para optar por el título de  
*Ingeniero de Sistemas*

Jhon Jairo Alvarez Londoño

**Director: Dr. Julio Ariel Hurtado Alegría**

*Universidad del Cauca*

Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Sistemas

Grupo IDIS – Investigación y Desarrollo en Ingeniería de Software  
Línea de Investigación en Procesos Software  
Popayán, 18 de noviembre de 2014

## Dedicatoria

*El camino ha sido un desierto, pero tu apoyo incondicional un oasis.  
A mi padre quien sin él no sería lo que soy.  
Mi mas grande modelo de persistencia y de superación.  
**¡¡que orgullo es ser tu hijo y que honor ser tu orgullo!!***

*Gracias por tu total cariño y entrega.  
Estaré eternamente agradecido con Dios por ser mi guía  
mi roca y mi fortaleza.  
Gracias por tu empeño en sacar esta familia adelante  
a pesar de que a veces no compartimos puntos de vista.*

*Tu rigidez y nobleza en tu forma de ser me inspira a ser mejor persona  
un mejor hijo y a estar mas decidido  
a las pruebas que la vida me colocará.*

*Por tu sacrificio en mantener mis obligaciones cumplidas  
por estar allí para animarme día a día  
por los momentos que no hemos podido estar en familia  
por que aprendí, aprendo y aprenderé de ti por el resto de mis días  
este trabajo también es tuyo.*

***¡¡Que Dios nos de salud y vida para que sigamos celebrando mas  
victorias y venciendo malos momentos juntos!!***

***¡¡Ahora es mi turno de ser tu oasis!!***

## Agradecimientos

Agradezco todo el apoyo que me han brindado:

- A Dios por permitir uno de los logros mas invaluables en mi vida, por brindarme la oportunidad de ser mejor persona, por ser mi más constante compañía y por haberme dado lo mejor para mi siempre.
- A mi padre Sigifredo Alvarez principalmente, sin él no hubiera podido ser quien soy, su "berraquera", (ante todo colombiano), su animo y su incondicional apoyo y amor en toda mi vida y en todos sus campos y facetas, reconozco que aún obteniendo éste título y los que vendrán aún seguiré recibiendo y aplicando sus consejos dado a su enorme e invaluable experiencia y sabiduría hasta el último día.
- Al Dr. Julio Ariel Hurtado cuyo trabajo ha sido de gran profesionalismo y entrega, quien me ha guiado en el logro de cada uno de mis objetivos académicos y ha representado un gran apoyo motivacional, componente esencial que permitió un desarrollo de calidad de este trabajo.
- A mi abuelo Carlos Alvarez Q.E.P.D. quien ha llevado los últimos años de su vida con gran sabiduría y paciencia frente a las adversidades en materia de salud, y quien me ha demostrado que se lucha una meta hasta el último momento.
- A mi hermana Paula Andrea Alvarez, que me refiere como principal modelo académico a seguir.
- A mi abuela Isaura Cajiao, quien me amparó en la niñez y cuyo cariño maternal aún me sigue animando en los tiempos difíciles.
- A mi tía Isaura Alvarez, quien también me ha ilustrado y aconsejado en diferentes aspectos y ha estado pendiente constantemente de mi progreso en todos los aspecto.
- Finalmente a mi tío Jorge Alvarez, quien ha sido mi principal modelo académico a seguir, ha demostrado gran interes en mi progreso a pesar de la distancia y me ha apoyado económicamente.

A todos muchas gracias por creer en mi y espero que puedan seguir compartiendo mas triunfos conmigo.

# Índice general

<b>Índice general</b>	<b>III</b>
<b>Índice de figuras</b>	<b>VII</b>
<b>Índice de tablas</b>	<b>X</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Planteamiento del problema . . . . .	2
1.2. Pregunta de investigación . . . . .	3
1.3. Objetivos . . . . .	3
1.3.1. Objetivo general . . . . .	3
1.3.2. Objetivos específicos . . . . .	4
1.4. Respecto al uso del estandar ISO/IEC 29110, El Proceso Unificado y de la VSE . . . . .	4
1.5. Método de exploración y evaluación a través de la ISO/IEC 29110-3 . . .	5
<b>2. Estado del arte y trabajos relacionados</b>	<b>8</b>
2.1. Marco conceptual . . . . .	8
2.2. Procesos software . . . . .	10
2.3. Modelos de proceso . . . . .	10
2.3.1. Clasificación de los modelos de proceso software . . . . .	10
2.4. El Proceso Unificado (UP) . . . . .	11
2.5. La norma ISO/IEC 29110 . . . . .	11
2.5.1. Visión General (TR ISO 29110-1) . . . . .	12
2.5.2. Perfiles (ISP) . . . . .	12
2.5.3. Guías(TR) . . . . .	12
2.6. Las normas ISO/IEC 15504 e ISO/IEC 12207 . . . . .	13
2.7. Eclipse Process Framework Project (EPF composer) . . . . .	15
2.8. Analysis and Visualization in Software Process Assesment (AVISPA) . . .	16
2.8.1. Salidas generadas por AVISPA . . . . .	16
2.8.2. Patrones de error identificados por AVISPA . . . . .	17
2.9. Trabajos relacionados . . . . .	17
<b>3. Modelo de evaluación sobre el estandar ISO/IEC 29110-5-1-1 para los pro- cesos basados en el Proceso Unificado</b>	<b>24</b>
3.1. Introducción . . . . .	24

3.2. Evaluación del Proceso Unificado . . . . .	25
3.2.1. Guía de evaluación respecto a la norma ISO/IEC 29110-5-1-1 . . . . .	25
3.2.1.1. Planeación de la evaluación . . . . .	25
3.2.1.2. Modelo de evaluación de la ISO/IEC 15504 . . . . .	26
3.2.1.3. Respecto a los artefactos de la ISO/IEC 29110-5-1-1 . . . . .	31
3.2.2. Ejecución y resultados del proceso de evaluación con UP . . . . .	32
3.2.2.1. Los flujos de trabajo del UP . . . . .	32
3.2.2.2. Actividades adicionales del UP involucradas en la evaluación . . . . .	33
3.2.2.3. Observaciones durante la ejecución de la evaluación . . . . .	33
3.2.2.4. Presentación y análisis de los resultados de la evaluación . . . . .	33
3.2.3. Consideraciones adicionales . . . . .	36
3.2.3.1. Equivalencia de roles . . . . .	36
3.2.3.2. Acerca de la ejecución del proceso de evaluación en UP . . . . .	36
3.3. Análisis de los resultados . . . . .	36
<b>4. UP-VSE . . . . .</b>	<b>38</b>
4.1. Introducción . . . . .	38
4.2. Características del proceso . . . . .	38
4.3. Construcción del proceso UP-VSE . . . . .	39
4.4. Ciclo de vida de UP-VSE . . . . .	40
4.4.1. El proceso UP-VSE: . . . . .	41
4.4.2. Disciplinas de UP-VSE . . . . .	52
4.4.3. Criterios para la definición UP-VSE . . . . .	53
4.5. Evaluación de UP-VSE con el proceso de evaluación . . . . .	54
4.5.1. Ejecución y resultados del proceso de evaluación con UP-VSE . . . . .	54
4.5.1.1. Clasificación de tareas por disciplinas de UP-VSE . . . . .	54
4.5.1.2. Evaluación de las tareas de UP-VSE . . . . .	54
4.5.1.3. Evaluación de los productos de trabajo de UP-VSE . . . . .	56
4.5.1.4. Relación de los roles de UP-VSE e ISO/IEC 29110-5-1-1 . . . . .	56
4.5.1.5. Acerca de la ejecución del proceso de evaluación en UP-VSE . . . . .	57
4.6. Evaluación de UP-VSE con AVISPA . . . . .	57
4.6.1. Resultados del análisis de UP-VSE con AVISPA . . . . .	57
4.6.1.1. Vistas obtenidas por AVISPA del proceso UP-VSE . . . . .	57
4.6.2. Acciones realizadas según los errores identificados a través de AVISPA . . . . .	60
4.6.3. UP-VSE con correcciones, visualización con AVISPA . . . . .	63
<b>5. Evaluación Incremental de UP-VSE . . . . .</b>	<b>65</b>
5.1. Introducción . . . . .	65
5.2. Evaluación a través del estudio de caso PRODIGIA . . . . .	66
5.3. Desarrollo del caso: . . . . .	66
5.3.1. Resumen . . . . .	66
5.3.2. Introducción . . . . .	67

5.3.3.	Diseño del estudio de caso . . . . .	67
5.3.3.1.	Objetivo . . . . .	67
5.3.3.2.	Preguntas de investigación . . . . .	67
5.3.3.3.	Descripción del caso . . . . .	67
5.3.3.4.	Selección del caso . . . . .	68
5.3.3.5.	Hipótesis . . . . .	69
5.3.3.6.	Indicadores e instrumentos de recolección . . . . .	69
5.3.3.7.	Herramientas de soporte para el estudio de caso . . . . .	70
5.3.3.8.	Ejecución del caso . . . . .	72
5.3.3.9.	Evaluación de PRODIGIA respecto a la ISO/IEC 29110-3 del perfil ISO/IEC 29110-5-1-1:2012 y usando AVISPA . . . . .	77
5.3.4.	Resultados . . . . .	78
5.3.4.1.	Ejecución y resultados del proceso de evaluación respecto a la ISO/IEC 29110-3 sobre PRODIGIA: . . . . .	78
5.3.4.2.	Presentación y análisis de los resultados respecto al modelo de evaluación de la ISO/IEC 29110-3 . . . . .	78
5.3.4.3.	Consideraciones adicionales respecto al modelo de evaluación . . . . .	80
5.3.4.4.	Acerca de la ejecución del proceso de evaluación en PRODIGIA . . . . .	81
5.3.4.5.	Vistas obtenidas por AVISPA del proceso PRODIGIA . . . . .	81
5.3.5.	Análisis de los resultados . . . . .	85
5.3.5.1.	Respecto al estudio de caso . . . . .	85
5.3.5.2.	Respecto a las encuestas . . . . .	85
5.3.5.3.	Respecto al modelo de evaluación de la ISO/IEC 29110-3 . . . . .	86
5.3.5.4.	Respecto a los datos obtenidos por AVISPA . . . . .	87
5.3.5.5.	PRODIGIA con correcciones en base a AVISPA . . . . .	87
5.3.6.	Evaluación cualitativa de la ejecución del estudio de caso . . . . .	89
5.4.	Correcciones sobre UP-VSE . . . . .	92
<b>6.</b>	<b>Conclusiones, limitaciones, trabajo futuro y aportes adicionales</b> . . . . .	<b>93</b>
6.1.	Conclusiones . . . . .	94
6.2.	Limitaciones . . . . .	96
6.3.	Trabajos futuros . . . . .	96
6.4.	Aportes adicionales . . . . .	97
	<b>Bibliografía</b> . . . . .	<b>99</b>
<b>A.</b>	<b>PRODIGIA: Ciclo de vida</b> . . . . .	<b>105</b>
A.1.	Introducción . . . . .	105
A.2.	Características del proceso . . . . .	105
A.3.	Ciclo de vida . . . . .	106
A.3.1.	Fases: . . . . .	107
A.3.1.1.	Fase de Formulación: . . . . .	107
A.3.1.2.	Fase de Especificación: . . . . .	109

A.3.1.3. Fase de Programación: . . . . .	113
A.3.1.4. Fase de Entrega: . . . . .	116
A.3.2. Actividades principales . . . . .	118
A.3.3. Roles: . . . . .	118
<b>B. Diseño de los instrumentos de recolección, caso PRODIGIA</b>	<b>119</b>
B.1. Introducción . . . . .	119
B.2. Encuestas . . . . .	119
B.2.1. Encuestas para tesis: . . . . .	120
B.2.1.1. Encuesta general: . . . . .	120
B.2.2. Encuesta para tutores de tesis: . . . . .	121
B.2.3. Encuesta para los integrantes del equipo de procesos: . . . . .	121
B.3. Artefactos . . . . .	121
B.4. Modelo de evaluación de la ISO/IEC 29110-3 . . . . .	122
B.5. Herramienta de soporte a la documentación . . . . .	122
B.6. Herramienta de soporte al trabajo colaborativo COLLAB . . . . .	122
<b>C. Especificación e implementación de la herramienta en Drupal</b>	<b>124</b>
C.1. Resumen . . . . .	124
C.2. Documentación de la herramienta de soporte . . . . .	124
C.2.1. Funcionalidades sobre el rol de tesista . . . . .	125
C.2.2. Funcionalidades sobre el rol de tutor de tesis . . . . .	127
C.3. Capturas de pantalla, prototipos funcionales . . . . .	130
<b>D. Minutas de reunión</b>	<b>131</b>
D.1. Minuta de reunión (09-04-2013): . . . . .	131
D.2. Minuta de reunión (28-04-2013): . . . . .	132
D.3. Minuta de reunión (08-10-2013): . . . . .	133
D.4. Minuta de reunión (02-05-2014): . . . . .	135
D.5. Minuta de reunión (14-05-2014): . . . . .	136
<b>E. Artículo del proyecto UP-VSE</b>	<b>138</b>
<b>F. Artículo uso del COLLAB como herramienta de colaboración en la definición de PRODIGIA</b>	<b>153</b>

## Índice de figuras

2.1. Marco conceptual . . . . .	9
2.2. Elementos de la evaluación de procesos en VSE [1]. . . . .	13
2.3. El entorno de trabajo EPF Composer. . . . .	15
2.4. Atributos de los nodos considerados como métricas en la vista de tareas por AVISPA . . . . .	17
3.1. Elementos instanciados de la evaluación en ISO/IEC 15504-2. . . . .	27
4.1. Ciclo de vida de UP-VSE . . . . .	40
4.2. Fases de UP-VSE . . . . .	41
4.3. Fase de Inicio . . . . .	41
4.4. Actividad Inicializar el proyecto . . . . .	42
4.5. Iteraciones de la fase de inicio . . . . .	42
4.6. Actividad planeación y gestion por iteración . . . . .	43
4.7. Actividad Identificar y refinar requisitos. . . . .	44
4.8. Actividad Diseñar arquitectura. . . . .	45
4.9. Actividad Integrar el desarrollo a la configuración . . . . .	45
4.10. Fase de Elaboración . . . . .	46
4.11. Iteraciones de la fase de Elaboración . . . . .	46
4.12. Actividad Desarrollo de incremento de software . . . . .	47
4.13. Actividad Pruebas de integración . . . . .	48
4.14. Fase de Construcción . . . . .	49
4.15. Fase de Transición . . . . .	49
4.16. Iteraciones de la fase de Transición . . . . .	50
4.17. Actividad Desplegar la versión actual a producción . . . . .	51
4.18. Evaluación de las tareas de UP-VSE 0.5 en AVISPA . . . . .	58
4.19. Evaluación de los roles de UP-VSE 0.5 en AVISPA . . . . .	59
4.20. Evaluación de los artefactos de UP-VSE 0.5 en AVISPA . . . . .	59
4.21. Excepción "SubscriptOutOfBounds" de AVISPA . . . . .	61
4.22. Evaluación de las tareas de UP-VSE alpha en AVISPA . . . . .	63
4.23. Evaluación de los roles de UP-VSE alpha en AVISPA . . . . .	64
4.24. Evaluación de los artefactos de UP-VSE alpha en AVISPA . . . . .	64
5.1. Proceso de evaluación de UP-VSE. . . . .	65
5.2. Estrategia de trabajo para el estudio de caso. . . . .	68
5.3. Herramienta de gestión documental de PRODIGIA . . . . .	71



5.4. Instancia de COLLAB para ser utilizado en PRODIGIA . . . . .	72
5.5. Ciclo de vida de PRODIGIA . . . . .	73
5.6. Reuniones para la socialización de proceso y de la herramienta de soporte al grupo AI. . . . .	76
5.7. Configuración de artefactos de PRODIGIA v. 0.3 arrojada por AVISPA. . . . .	82
5.8. Configuración de tareas de PRODIGIA v. 0.3 arrojada por AVISPA. . . . .	82
5.9. Configuración de roles de PRODIGIA v. Alpha arrojada por AVISPA. . . . .	82
5.10. Configuración de artefactos de PRODIGIA v. 0.3 arrojada por AVISPA antes (arriba) y después (abajo) de las correcciones . . . . .	90
5.11. Configuración de tareas de PRODIGIA v. 0.3 arrojada por AVISPA antes (arriba) y después (abajo) de las correcciones . . . . .	91
5.12. Configuración de roles de PRODIGIA v. Alpha arrojada por AVISPA. . . . .	92
A.1. Ciclo de vida de PRODIGIA . . . . .	107
A.2. Fases de PRODIGIA . . . . .	107
A.3. Fase de formulación . . . . .	108
A.4. Actividad Construcción del Anteproyecto . . . . .	108
A.5. Fase de Especificación . . . . .	109
A.6. Iteraciones de la fase de especificación . . . . .	109
A.7. Actividad investigar aspectos teóricos del simulador . . . . .	110
A.8. Actividad Definir la arquitectura . . . . .	110
A.9. Actividad Identificar y actualizar requisitos . . . . .	111
A.10. Actividad Probar software . . . . .	111
A.11. Actividad Construir el diagrama de clases . . . . .	112
A.12. Actividad Construir el diagrama de comunicación . . . . .	112
A.13. Actividad Documentar el software . . . . .	113
A.14. Fase de programación . . . . .	113
A.15. Iteraciones de la fase de programación . . . . .	114
A.16. Actividad Identificar y actualizar requisitos . . . . .	114
A.17. Actividad Construir el software . . . . .	115
A.18. Actividad Documentar el software . . . . .	115
A.19. Fase de entrega . . . . .	116
A.20. Actividad Documentar el software . . . . .	116
A.21. Actividad Liberar entregables . . . . .	117
B.1. Instancia de COLLAB para ser utilizado en PRODIGIA (Menú principal)	123
B.2. Instancia de COLLAB para ser utilizado en PRODIGIA (Reuniones y estado de las tareas) . . . . .	123
C.1. Modelo de caso de uso, funcionalidades del rol de tesista . . . . .	125
C.2. Modelo de casos de uso, funcionalidades del tutor de tesis . . . . .	127
C.3. Página principal de la herramienta de soporte . . . . .	130
C.4. Funcionalidad de "Agregar tesis" de la herramienta de soporte. . . . .	130
D.1. Proceso acordado sobre la captura de requisitos en el grupo IA. . . . .	133
D.2. Proceso acordado sobre la captura de requisitos en el grupo IA. . . . .	134
D.3. Reuniones para validación y verificación de la herramienta PRODIGIA. . . . .	135

UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras  
de Software Basado en el Proceso Unificado

---

D.4. Socialización de la herramienta de soporte de PRODIGIA. . . . . 136

## Índice de tablas

2.1. Patrones de error identificados por AVISPA . . . . .	18
2.2. Facilidades de Implementación de algunos modelos de procesos de software descriptivos. . . . .	19
2.3. Implementaciones de normas a través de algunos modelos de procesos de software prescriptivos. . . . .	22
3.1. Equivalencia entre procesos y tareas involucradas en el mapeo . . . . .	28
3.2. Relación de subprocesos entre la norma ISO/IEC 12207 y UP . . . . .	29
3.3. Rangos de capacidad y puntajes asignados . . . . .	29
3.4. Numeración de tareas en I. Jacobson et. al. [2] . . . . .	32
3.5. Evaluación de las tareas de UP respecto al estándar ISO/IEC 29110-5-1-1:2012 . . . . .	34
3.6. Evaluación de los artefactos del UP y UP-VSE respecto al estándar ISO/IEC 29110-5-1-1:2012 . . . . .	35
3.7. Equivalencia de roles entre ISO/IEC 29110-5-1-1 y UP . . . . .	36
4.1. Numeración de tareas de UP-VSE . . . . .	54
4.2. Evaluación de las tareas de UP-VSE respecto al estándar ISO/IEC 29110-5-1-1:2012 . . . . .	55
4.3. Equivalencia de roles entre ISO/IEC 29110-5-1-1 y UP-VSE . . . . .	56
4.4. Acciones correctivas sobre resultados arrojados por AVISPA del proceso UP-VSE . . . . .	62
5.1. Métricas del estudio de caso . . . . .	69
5.2. Reutilización de artefactos de UP-VSE en PRODIGIA . . . . .	74
5.3. Numeración de tareas de PRODIGIA . . . . .	77
5.4. Evaluación de las tareas de PRODIGIA respecto al estándar ISO/IEC 29110-5-1-1:2012 . . . . .	79
5.5. Evaluación de los artefactos de PRODIGIA respecto al estándar ISO/IEC 29110-5-1-1:2012 . . . . .	80
5.6. Equivalencia de roles entre ISO/IEC 29110-5-1-1 y PRODIGIA . . . . .	80
5.7. Resultados de la encuesta general para tesisistas . . . . .	84
5.8. Resultados de la encuesta para los tutores de tesis y el equipo de procesos . . . . .	84
5.9. Acciones correctivas sobre resultados arrojados por AVISPA del proceso UP-VSE . . . . .	88

B.1. Encuesta para obtener la completitud del proceso PRODIGIA . . . . .	120
B.2. Encuesta para obtener el nivel de satisfacción de los tutores de tesis sobre el proceso PRODIGIA . . . . .	121
B.3. Encuesta para obtener el nivel de satisfacción del equipo de procesos sobre el proceso PRODIGIA . . . . .	121
C.1. Caso de uso "Gestor avance de tesis". . . . .	125
C.2. Caso de uso "Ver tesis". . . . .	126
C.3. Caso de uso "Ver equipos asignados". . . . .	126
C.4. Caso de uso "Ver todos los informes de tesis". . . . .	126
C.5. Caso de uso "Ver los informes revisados por el tutor". . . . .	126
C.6. Caso de uso "Ver los informes sin revisar por el tutor". . . . .	126
C.7. Caso de uso "Aceptar solicitud de relación del tutor de tesis". . . . .	127
C.8. Caso de uso "Iniciar Sesión". . . . .	127
C.9. Caso de uso "Gestor tesis". . . . .	128
C.10.Caso de uso "Gestor asignar un equipo a un estudiante". . . . .	128
C.11.Caso de uso "Gestor página básica". . . . .	129
C.12.Caso de uso "Gestor usuarios y permisos". . . . .	129
C.13.Caso de uso "Enviar solicitud de relación de tutor de tesis". . . . .	129

## Capítulo 1

# Introducción

Una industria de software es una industria capaz de hacer la diferencia con productos de calidad y proyectos productivos. Estos dos aspectos se pueden lograr a partir de muchas estrategias técnicas y de gestión, puesto que este atributo permite el incremento de su competitividad [3], sin embargo, ésta requiere la especialización de las actividades organizacionales y de modelos de procesos [4] aptos para su respectivo contexto.

Un proceso es un “conjunto de actividades relacionadas que conducen a la creación de un producto software”[5, 6]. Según I. Jacobson et al. “Un proceso define el quién está haciendo qué, cuándo y cómo construir un producto software o mejorar uno existente”[2].

La forma de controlar, asegurar y gestionar la calidad en la industria del software normalmente requiere grandes recursos y una compleja estructura organizacional, sin embargo la industria de desarrollo de software se compone mayoritariamente de VSE<sup>1</sup> de en donde hay limitaciones en los recursos y en la formación del personal [7]. Por este motivo se hace necesario enfocar recursos y esfuerzos en el mejoramiento de calidad de estas organizaciones, y en el caso particular de éste estudio, de las VSE.

La propuesta de modelo de ciclo de vida UP-VSE<sup>2</sup> pretende disminuir este desconocimiento de la norma ISO/IEC 29110, creada por ISO<sup>3</sup> para las VSE que desarrollan software y necesitan estandarizar sus procesos, aprovechando las buenas prácticas del Proceso Unificado que es mas conocido tanto en la industria como en el ámbito académico [8] evaluada a través de un conjunto de 3 filtros: La norma ISO/IEC 29110-3 que es la sección de la norma que se encarga de presentar lineamientos para evaluación de procesos (ver sección 2.5), un estudio de caso (ver sección 5.2) y

---

<sup>1</sup>VSE: Acrónimo de Very Small Entities, cuya traducción asociada a éste proyecto es pequeñas entidades de software. Haciendo uso de éste termino, se conocerán como VSE para el resto de este informe.

<sup>2</sup>El Proceso Unificado orientado a Pequeñas Entidades Desarrolladoras de Software

<sup>3</sup>ISO: Organización Internacional de Normalización, organismo encargado de buscar la estandarización de normas de productos y seguridad para las empresas u organizaciones públicas o privadas a nivel internacional

utilizando la herramienta AVISPA (ver sección 4.6).

## 1.1. Planteamiento del problema

La mayoría de los modelos o normas de calidad existentes en el mercado están sobredimensionadas para el contexto de entidades pequeñas [9]. Para abordar este problema ha sido creada la norma ISO/IEC 29110, un nuevo estándar publicado en el 2011 enfocado en la mejora de los procesos de software en entidades muy pequeñas, desarrollado por el grupo SC7-WG24 de la ISO [10].

A pesar de este esfuerzo, es aún muy difícil la adopción de este estándar en las pequeñas entidades debido a la dificultad para interpretarlo, por lo que las VSE continúan sin la posibilidad de adoptar prácticas de calidad acordes a sus necesidades [9]. A todo esto se suma el no tener la disponibilidad financiera para capacitarse en cada norma y referente que aparece en su sector, que normalmente la industria no está convencida del directo retorno a la inversión de un programa de mejoramiento [11]. Todas las dificultades anteriores no le permiten a la industria de software VSE mostrarse competitivas frente al mercado [12, 13]. Se han realizado intentos de mejorar la comprensión de esta norma a través del uso de paquetes de implementación (DP) [14, 15, 16, 17], sin embargo, estos paquetes poseen limitaciones, debido a que no suelen tener información de prácticas empíricas que sustenten su uso a través de un ciclo de vida adecuado para su implementación en un proyecto, ni experiencias de su ejecución para su reutilización en otros proyectos [15].

Más específicamente, la norma ISO / IEC 29110-5-1-1 (Guía de ingeniería y gestión [18]) minimiza el alcance y hace más fácil la implementación a una VSE, pero, a pesar de esta ventaja, aún es difícil de adecuar la interpretación, debido a que estas entidades requieren asistencia para definir sus procesos software, acción que indica un llamado a la norma en términos de su facilidad para ser implementada [15].

O'Connor et al. [19] presenta una evaluación del compromiso de la gestión a través de la norma ISO/IEC 29110 en VSEs utilizando como principal método la entrevista a un total de 10 gestores experimentados en una serie de compañías de desarrollo de software muy pequeñas para saber su opinión, actitud y percepción del nuevo estándar ISO/IEC 29110, la información ha sido coleccionada y organizada en 3 categorías: Intereses, conocimiento y aceptación de los estándares del ciclo de vida. El estudio señaló que la razón principal para no adoptar los estándares fue la falta de estabilidad en los requerimientos de software afectando la estabilidad de algunas tareas que exige cumplir la norma, la escasez de recursos y las dificultades percibidas para definir un proceso organizacional.

Por otro lado, el proceso unificado (UP) es una metodología de desarrollo de software<sup>1</sup> ampliamente utilizada por todo tipo de empresas desarrolladoras de software y por los programas de ingeniería de sistemas y afines en la formación de sus ingenie-

---

<sup>1</sup> Metodología de desarrollo de software: es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información

ros de software. Debido a la cercanía que el proceso tiene con la industria, puede ser considerado un activo científico valioso para la implementación de modelos de referencia o normas. Dado que el UP es considerado un framework de procesos, un conjunto de soluciones adecuadas a cada empresa pueden ser derivadas de éste [20]. De esta forma, el UP puede servir como un activo reutilizable que apalanque la rápida aceptación del modelo ISO 29110. Sin embargo al problema de interpretar la norma, se le estaría agregando la dificultad de interpretarla en el contexto de un proceso específico que no es aún su proceso organizacional. De esta forma es necesario contar con implementaciones asimilables de la norma que describan procesos orientados hacia la organización y no hacia la norma.

Como parte del proceso para responder la pregunta de investigación (ver sección: 1.2), éste proyecto busca definir una implementación aplicable de la norma apalancados en una especificación del proceso unificado. El grupo IDIS ha venido trabajando en este proceso y en su adaptación como una forma de acelerar la definición de procesos para la industria regional [21]. Además, ha venido realizando estudios sobre la importancia y el impacto que tiene la adaptación de normas y modelos de calidad para lograr reconocimiento y mejoras. Gracias a estos resultados comprendemos la importancia de una implementación fácil de realizar, específicamente, de la norma ISO 29110 [7].

## 1.2. Pregunta de investigación

Basándose en el contexto de las VSE y tomando en cuenta los progresos significativos sobre la formalización del desarrollo de software que han sido llevados a cabo durante los últimos años, se ha elaborado la siguiente pregunta de investigación: *¿Cómo facilitar la implementación<sup>1</sup> de la norma ISO 29110-5-1-1 sobre las pequeñas organizaciones de desarrollo de software a través del Proceso Unificado?*

## 1.3. Objetivos

### 1.3.1. Objetivo general

Desarrollar una implementación reutilizable de la norma ISO/IEC 29110-5-1-1 (Guía de ingeniería y gestión, sección técnica) para la industria de software VSE basada en el proceso unificado UP.

---

<sup>1</sup>Implementación: en este contexto se trata de la ejecución de un estándar sobre una metodología de desarrollo de software [22]

### 1.3.2. Objetivos específicos

- Establecer el estado del arte sobre la implementación de la norma ISO/IEC 29110-5-1-1 (Guía de ingeniería) basada en el Proceso Unificado y en los reportes de experiencia en la industria.
- Determinar el grado en el que el Proceso Unificado (UP) implementa la sección técnica del estándar ISO 29110-5-1-1 a través de una relación de implementación<sup>1</sup> entre estos dos modelos de proceso sobre un estudio de caso.
- Proponer un modelo de proceso que cumpla parcialmente con la norma ISO 29110-5-1-1 (subsección de ingeniería) a través de la adaptación y extensión de los elementos del proceso unificado como una base reutilizable para la implementación de la norma.
- Evaluar el modelo de proceso propuesto a través de la herramienta de evaluación de procesos AVISPA<sup>2</sup> y sobre un estudio de caso en un ambiente VSE real con el fin de hallar aspectos a mejorar.

## 1.4. Respecto al uso del estandar ISO/IEC 29110, El Proceso Unificado y de la VSE

Las VSE representan un porcentaje alto de la industria de software mundial [7], por eso es muy importante trabajar sobre su fortalecimiento y por eso esta tesis se enfoca hacia éstas. Una VSE (pequeña entidad de desarrollo de software) es una entidad (empresa, organización, departamento o proyecto) que tiene hasta 25 personas. Las VSE también desarrollan y/o mantienen software que es usado en sistemas más amplios; por lo tanto, el reconocimiento de las VSE como proveedores de software de alta calidad a menudo es requerida [23]. Las Pymes<sup>3</sup> constituyen un amplio porcentaje dentro de la industria de desarrollo de software [7] dentro de las cuales se encuentran las VSE.

---

<sup>1</sup>Implementación entre modelos de proceso, que en este caso, se refiere a la correspondencia entre prácticas de un modelo (ISO/IEC 29110) y las de la metodología de desarrollo de software (Proceso Unificado UP).

<sup>2</sup>AVISPA (Analysis and Visualization for Software Process Assessment): es una herramienta que construye planos a partir de métricas de software y en ellos resalta los errores en patrones de procesos que facilitan su análisis.

<sup>3</sup>Pymes: micro, pequeñas y medianas empresas



La VSE a la que hará referencia la presente tesis (grupo de investigación AI<sup>1</sup>) se clasifica como una VSE debido a que dicha norma está creada para grupos de desarrollo de menos de 25 personas [1] y su selección obedece a los siguientes criterios:

1. La disponibilidad de una VSE para trabajar en el proyecto y
2. Lo típico de la realización del proceso debido a que guarda amplia relación con los artefactos de esta VSE y el Proceso Unificado como el modelo de casos de uso, modelo de clases, modelo de comunicación de objetos, entre otros.

La norma ISO/IEC 29110 ha sido creada para este sector de la industria; está acotada a sus necesidades y cada vez se posiciona como el modelo para la industria de software VSE.

El estándar ISO/IEC 29110 ha sido escogido principalmente para realizar una disminución en los elementos del proceso que ofrece el Proceso Unificado obteniendo un modelo adecuado y específico para una pequeña entidad de desarrollo de software. Este estándar ha sido utilizado en una amplia variedad de trabajos que se orientan por el entendimiento en las VSE [16, 17, 24] y su implementación [15, 12, 13, 7] demostrando una gran importancia y uso de esta norma en la industria de desarrollo de software en las pequeñas entidades en cualquier país.

El Proceso Unificado ofrece diversas ventajas, principalmente sobre el uso de la documentación organizada, completa y disponible constantemente que ofrece este modelo de proceso y del desarrollo de elementos de software en paralelo, lo cual representa una ventaja organizacional para las VSE que desean alcanzar una certificación en la norma ISO/IEC 29110 [2]. El Proceso Unificado ha sido también utilizado como referencia para el aprendizaje y adopción de distintos modelos de calidad [25]. A nivel regional, el Proceso Unificado también ha demostrado su aplicabilidad en las VSE [26] demostrando así su amplio nivel de utilidad para la industria de desarrollo de software.

Basándose en el amplio recorrido del Proceso Unificado, de su aplicabilidad a gran variedad de proyectos y en la amplia aceptación de la ISO/IEC 29110 como estándar para certificación de las VSE ha sido basada la estrategia seguida para la construcción de la propuesta de esta tesis: UP-VSE.

## 1.5. Método de exploración y evaluación a través de la ISO/IEC 29110-3

Con el fin de lograr los objetivos de la sección 1.3 y responder a la pregunta de investigación de la sección 1.2 se ha seguido los marcos metodológicos de investigación de A. Hurtado [27] y de M. Bunge [28] en conjunto con el trabajo de Runeson et.

---

<sup>1</sup>El grupo AI (Automática Industrial) es el grupo de investigación adscrito a la Universidad del Cauca que facilitó la VSE para la ejecución del proceso PRODIGIA

al. [29] como de los estudios de caso y de M. Shaw [30] en el uso de las recomendaciones particulares de una buena investigación en ingeniería de software. El método de investigación consta de 3 fases (e hitos): exploración (problema definido), formulación (modelo definido), y validación (hipótesis validada) que en conjunto ofrecen la vista de gestión para la cuales se definieron los aspectos específicos del proyecto por cada fase:

■ **Fase de Exploración:**

- **Planteamiento del problema:** Dado a la evidencia de las dificultades que las VSE mantienen a pesar de los estándares y las buenas prácticas [14], se ha desarrollado un prototipo de proceso inicial pensado en bajos costos de implementación, los pocos roles presentes en el desarrollo de cada uno de los artefactos en las VSE y en la disminución del desarrollo caótico de código y la documentación inadecuada de software.
- **Producto:** Aproximación inicial del proceso PRODIGIA<sup>1</sup> (ver A).

■ **Fase de Formulación:**

- **Construcción y documentación del proceso PRODIGIA:** Se construye un proceso incrementalmente para la elaboración de software a través de un estudio de caso exploratorio (PRODIGIA - requerimientos, ver sección 5.2 y el anexo A) en SPEM<sup>2</sup> con el fin de hallar un punto inicial en la construcción de un modelo final validado.
- **Selección del modelo de evaluación y evaluación de UP bajo la norma:** Tanto la propuesta (UP-VSE) como el estudio de caso (PRODIGIA) están basados en el Proceso Unificado de I. Jacobson et. al. [2]. Para su construcción se encontró necesario hallar oportunidades de mejora del UP, puesto que este trabajo está orientados a VSE se ha utilizado la sección 3 de evaluación de modelos de proceso de la ISO/IEC 29110.
- **Construcción del proceso UP-VSE:** Basándose en los resultados preliminares durante la construcción de PRODIGIA se elabora la propuesta de modelo de ciclo de vida UP-VSE en SPEM.

■ **Fase de Validación:**

- **Diseño del estudio de caso:** Para realizar el estudio de caso de inició un diseño en el que se definieron variables, metodología de acción del proceso en el estudio de caso y herramientas de captura de dichas variables con el fin de conducir y ejecutar correctamente la metodología.

---

<sup>1</sup>PRODIGIA (PROceso de Desarrollo Industrial del Grupo Automática Industrial): Es un proceso de desarrollo de software elaborado en el estudio de caso de la presente tesis y orientado para la construcción de simuladores de intervención quirúrgica soportando formatos académicos de la Universidad del Cauca.

<sup>2</sup>Software & Systems Process Engineering Metamodel

- **Construcción y evaluación del proceso UP-VSE:** Se finaliza la construcción del proceso UP-VSE con sus respectivos ajustes finales y se complementa con la evaluación en AVISPA y a través del segundo estudio de caso de PRODIGIA.
- **Construcción, ejecución y validación del proceso prototipo PRODIGIA:** Se realiza la validación del proceso PRODIGIA con un segundo estudio de caso y se registran los resultados.

## Capítulo 2

# Estado del arte y trabajos relacionados

### 2.1. Marco conceptual

La figura 2.1 ilustra la distribución de este capítulo en los diversos temas utilizados para realizar el desarrollo de esta tesis.

Primero se introducen conceptos sobre el uso de procesos y modelos creados para el desarrollo de software. Luego se introducen los dos modelos de proceso, el Proceso Unificado y la ISO/IEC 29110 con su estructuración que son utilizados principalmente para efectos de esta tesis. La norma ISO/IEC 29110-3 utiliza otros modelos de ciclo de vida que son referenciados para construir un modelo de evaluación de procesos (ISO/IEC 155504 e ISO/IEC 12207) para VSE que son introducidos en este capítulo. También han sido utilizadas un conjunto de herramientas para reforzar la evaluación y documentación de los modelos de proceso utilizados en este trabajo (EPF y AVISPA), estas herramientas están descritas en el presente capítulo. Por último se exponen los trabajos relacionados ilustrando las diversas problemáticas al rededor de la elaboración de la propuesta del presente trabajo y sus conclusiones.

UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras de Software Basado en el Proceso Unificado

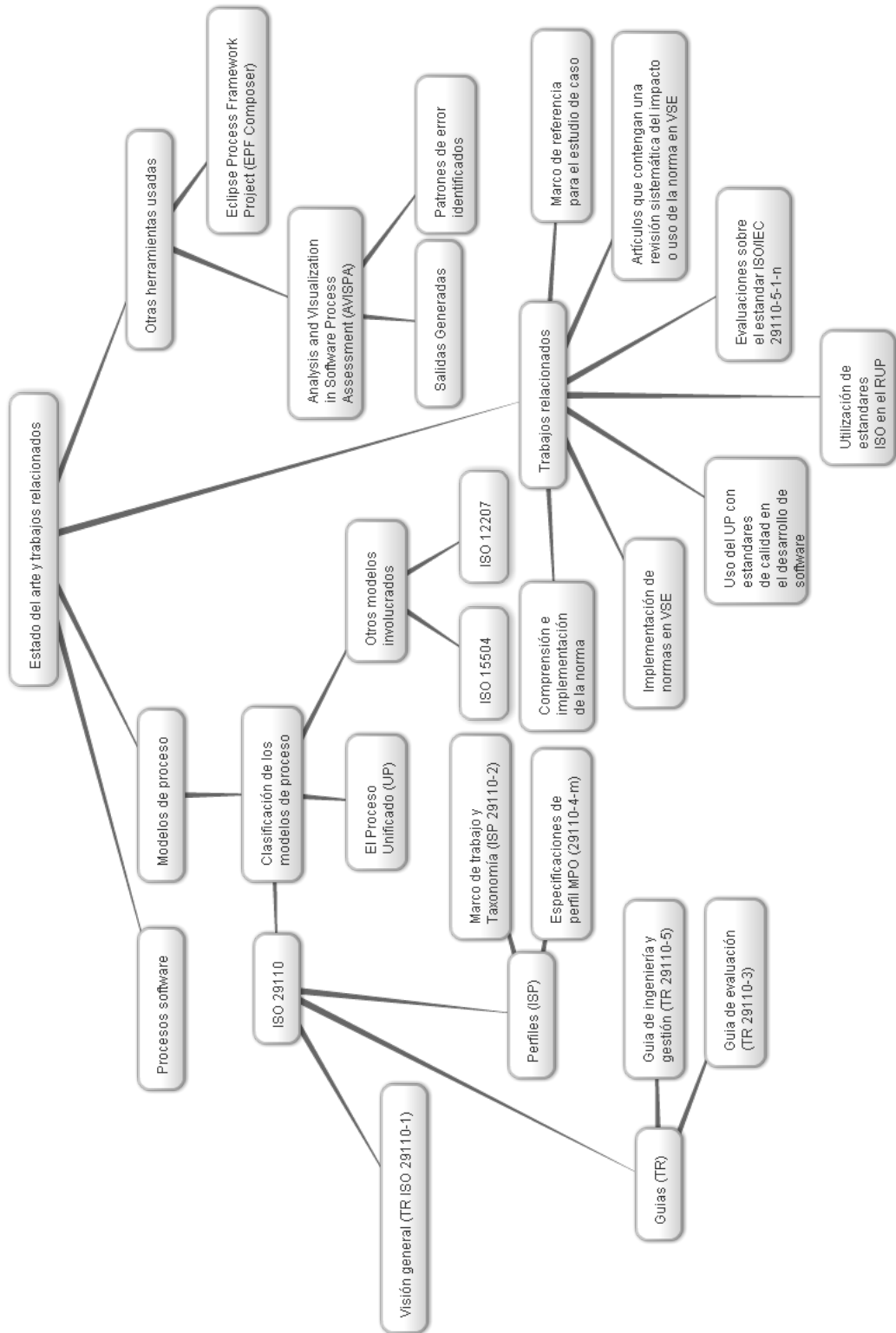


Figura 2.1: Marco conceptual

## 2.2. Procesos software

Un proceso es un “conjunto de actividades relacionadas que conducen a la creación de un producto software”[6, 5]. Introduce el concepto de proceso software como un método para determinar de manera sistemática los pasos necesarios para soportar todo lo relacionado con la creación del producto software, además, expone que todos los procesos software deben incluir 4 actividades que son fundamentales para la ingeniería del software: *Especificación del software*, *Diseño e implementación del software*, *Validación del software* y *Evolución del software*. Especifica también que tanto las actividades, como los procesos, deben incluir productos, roles y pre y post condiciones los cuales son sentencias que deben ser verdaderas antes y después de una actividad del proceso [6]. Según I. Jacobson et al. “Un proceso define el quién está haciendo qué, cuándo y cómo construir un producto software o mejorar uno existente”[2]. Estos trabajos introducen el concepto del proceso software sobre el cual aborda en detalle y hace énfasis esta tesis.

## 2.3. Modelos de proceso

Feiler P.H. et al. [31] exponen el modelo de procesos como “una representación abstracta de una arquitectura de proceso, de un diseño o definición” para el entendimiento, toma de decisiones y representaciones de los distintos procesos a través de un ciclo de vida. También añade que los modelos de proceso son elementos, cuya abstracción captura estos aspectos de un proceso relevantes para el modelado. La norma ISO 29110 y el Proceso Unificado (UP) como metodología de desarrollo de software son modelos de proceso clasificados en dos tipos (Ver 2.3.1 Clasificación de los modelos de proceso software), este trabajo recoge los elementos compatibles entre estos modelos y trata de presentar como resultado un proceso de referencia que intenta capturar un nivel de abstracción similar al UP pero acotado con los elementos de proceso de la norma ISO 29110 tratando de lograr que el producto de esta tesis pueda ser utilizable y confiable en las VSE.

### 2.3.1. Clasificación de los modelos de proceso software

El trabajo de U. Becker realiza una clasificación de los modelos de proceso en descriptivos (que describen la manera de cómo algo realiza en realidad) y prescriptivos (que describen como algo debería realizarse) [32]. O. Armbrust et al. señalan una diferencia característica entre estos 2 tipos de proceso, mientras que un modelo de proceso descriptivo es creado por la observación de los procesos ejecutados actualmente, un modelo de proceso prescriptivo por lo general señala la dirección de todos los temas relevantes para el desarrollo de una pieza de software, otra característica indicada por el mismo autor menciona que un modelo de proceso descriptivo muestra lo que la gente realiza día a día, lo que significa que deja al descubierto las prácticas que se llevan a cabo mientras que un modelo de proceso prescriptivo le indica a la

gente el que hacer, lo que implica que debe realizar algunas cosas diferentes de como lo hacen regularmente y cambiar sus hábitos lo que en ingeniería de software es una de las metas más difíciles de lograr. Un despliegue exitoso de una perspectiva de proceso significa un cambio en los hábitos de la gente [33].

## 2.4. El Proceso Unificado (UP)

El trabajo de I. Jacobson et al. describen el Proceso Unificado como “un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto” [2].

El proceso unificado posee 3 características fundamentales: está dirigido por casos de uso, los cuales representan “un fragmento de funcionalidad del sistema que proporcionan al usuario un resultado importante”[2], está centrado en la arquitectura de software que provee los cimientos más representativos del sistema y es iterativo e incremental, en donde se dividen los trabajos en partes más pequeñas o mini proyectos. Cada mini proyecto es una iteración que finalmente culmina en un incremento del software concluyendo así una versión del mismo.

El ciclo de vida del proceso unificado consta de 4 fases (inicio, elaboración, construcción y transición) en donde cada una termina con un hito y contiene un número finito de iteraciones. A lo largo del ciclo de vida se ejecutan los 5 flujos de trabajos fundamentales (requisitos, análisis, diseño, implementación y pruebas). Este trabajo considera necesario añadir la gestión de proyectos como flujo de trabajo de soporte, ya que la evaluación de UP con la ISO 29110 así lo requiere. Así cada flujo de trabajo del proceso unificado señala un proceso a seguir el cual puede ser contrastado con la sección de ingeniería de la norma, y ha sido uno de los factores por los cuales ha sido seleccionado el Proceso Unificado y que toma esta tesis para el diseño del proceso de referencia resultante.

## 2.5. La norma ISO/IEC 29110

Para apoyar las pequeñas empresas en la mejora de sus procesos internos organizacionales, ISO, a través del grupo de trabajo SC7-WG24 ha creado la norma ISO 29110. Aplicable a Very Small Entities (VSE), y fueron denominadas así por ser organizaciones de menos de 25 personas [34].

La norma establece un marco común para describir perfiles evaluables del ciclo de vida de software para ser usados en las VSE.

### 2.5.1. Visión General (TR ISO 29110-1)

Introduce los conceptos principales necesarios para la comprensión de la norma ISO 29110, aspectos de negocio, características y requisitos de VSE. Esta sección de la norma aclara la razón de ser de los perfiles específicos, documentos, estándares y guías de VSE e introduce los conceptos de proceso básico, ciclo de vida y estandarización, y la familia de documentos ISO 29110.

### 2.5.2. Perfiles (ISP)

Definen como objetivo agrupar referencias a y/o partes de otros documentos de manera formal con el fin de adaptarlos a las necesidades y características de las VSE, lo que implica producir 2 tipos de documentos para esta norma:

- *Marco de trabajo y taxonomía (ISP 29110-2)*
- *Especificaciones de perfil MPO (29110-4-m)*

### 2.5.3. Guías(TR)

Contienen directrices de aplicación (de dominio específico) sobre cómo realizar los procesos para alcanzar los niveles de madurez (por ejemplo, actividades recomendadas, medidas, técnicas, plantillas, modelos, métodos, entre otros). Hay 2 tipos de guías:

#### ***Guía de evaluación (TR29110-3)***

Describe el proceso a seguir para realizar una evaluación que determine las capacidades de proceso y/o madurez organizacional, además de ser adecuado para realizar fines de autoevaluación y es aplicable a todos los perfiles identificados de acuerdo a 29110-3.

Esta sección de la norma indica la configuración de otras normas relacionadas que sirven como parámetros para llevar a cabo una organización adecuada de la evaluación de un proceso VSE. La figura 2.2 ilustra la configuración de las normas involucradas en la elaboración de una evaluación en una VSE.

La metodología de desarrollo está acorde a principios establecidos en la norma ISO 15504-2 (evaluación de procesos). Así, la guía de evaluación de la ISO/IEC 29110-3 indica que las VSE deben cumplir el nivel 1 del estándar ISO 15504-2 para su respectiva aprobación.

#### ***Guía de ingeniería y gestión (TR29110-5)***

Proporcionan orientación sobre la implementación de un perfil al interior de una empresa, su organización indica que para cada perfil, existe un documento de guía



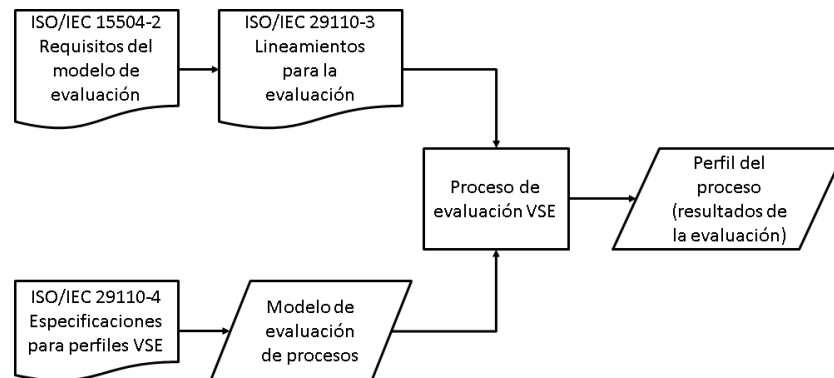


Figura 2.2: Elementos de la evaluación de procesos en VSE [1].

de ingeniería y gestión identificado como 29110-5-m-n, siendo m el número del perfil de grupo MPO al que se asocia ésta guía y n el número asignado a este perfil. Por ser dirigido a VSE, se elabora en un lenguaje más sencillo y de más fácil comprensión que se utiliza habitualmente en los estándares.

El perfil 29110-5-1-1 (Guía de ingeniería y gestión - Perfil Básico) se compone de 2 procesos: *Administración del Proyecto* y *Desarrollo de Software*, de los cuales este estudio sólo tomará la guía de ingeniería como marco de referencia para la evaluación. Esta sección de la norma cuenta con 6 subprocesos (*Software Implementation Initiation, Software Requirements Analysis, Software Architectural and Detailed Design, Software Construction, Software Integration and Test, Product Delivery*) que sirven de puntos de referencia para el análisis del Proceso Unificado y justifican la elección de este estándar para su uso en esta tesis.

## 2.6. Las normas ISO/IEC 15504 e ISO/IEC 12207

La organización ISO ha desarrollado una serie de estándares generales cuyo propósito es elaborar una forma de elaboración y evaluación adecuadas de software en la industria, aparte de la ISO/IEC 29110, ISO ha elaborado otras 2 normas que sirven de principales parámetros en la evaluación de este trabajo:

la norma ISO/IEC 12207 es el ciclo de vida de los procesos de desarrollo de software, y se divide principalmente en procesos principales, de soporte y de la organización, estableciendo un marco de trabajo común para procesos software del ciclo de vida que puede ser referenciada por la industria del software, reutiliza la terminología y rescata procesos de la ISO/IEC 15288 [35].

La norma ISO/IEC 15504 o SPICE es un modelo de evaluación de procesos para compañías de desarrollo de software el cual ofrece un marco para determinar el nivel de capacidad de un proceso de software, Según la ISO/IEC 15504-1:1998 [36] esta norma se compone principalmente de:

- **Parte 1:** Guía introductoria y conceptos
- **Parte 2:** Un modelo de referencia para procesos y capacidad del proceso
- **Parte 3:** La realización de una evaluación
- **Parte 4:** Guía para la realización de evaluaciones
- **Parte 5:** Una guía para la realización de un modelo de evaluación e indicadores
- **Parte 6:** Guía para la competencia de los evaluadores
- **Parte 7:** Guía de uso para la mejora de procesos
- **Parte 8:** Guía de uso en la determinación de la capacidad del proceso de proveedores
- **Parte 9:** Vocabulario

La norma ISO/IEC 29110-3:2011 menciona que para realizar la evaluación de un proceso en una VSE debe cumplir con el nivel de capacidad 1 de la norma ISO/IEC 15504-2 cumpliendo los objetivos de los perfiles (modelos de referencia) contemplados en la parte 5 de la norma ISO/IEC 29110. Este trabajo se centra en el cumplimiento del perfil ISO/IEC 29110-5-1-1:2012. Mas detalles de la instanciación de la evaluación de la ISO/IEC 15504 en una VSE se pueden hallar en el capítulo 3 del presente trabajo.

***Descripción de los niveles de capacidad utilizados por la ISO/IEC 29110-3 para perfiles básicos:***

- **Nivel 0:** "El proceso no esta implementado o falla en lograr su propósito", para éste trabajo se han tomado los objetivos de los subprocesos de ingeniería de requisitos de la norma.
- **Nivel 1:** "El proceso implementado cumple el propósito del proceso de referencia", asociado a éste nivel se encuentra el atributo de proceso:
  - **PA. 1.1 - Realización del proceso:** La plena realización de este atributo de proceso ofrece como resultado que el proceso logre sus salidas definidas. A continuación se ilustran los respectivos indicadores de proceso a través de practicas de gestión (MP) [37] seleccionadas en ésta evaluación:
    - **MP 1.1.1 - Identificar los productos de trabajo de entrada y salida:** aparte de su identificación se procede a verificar las características que deben tener estos productos y si cumplen o ayudan a cumplir o no el objetivo del proceso.
    - **MP 1.1.2 - Se debe asegurar que el alcance del trabajo se ha identificado para la ejecución de los procesos y de los productos de trabajo a ser usados y producidos en el proceso:** para cada proceso, las prácticas base conllevan a lograr el propósito y salidas definidas en el proceso y existe evidencia que las prácticas base son llevadas a cabo.

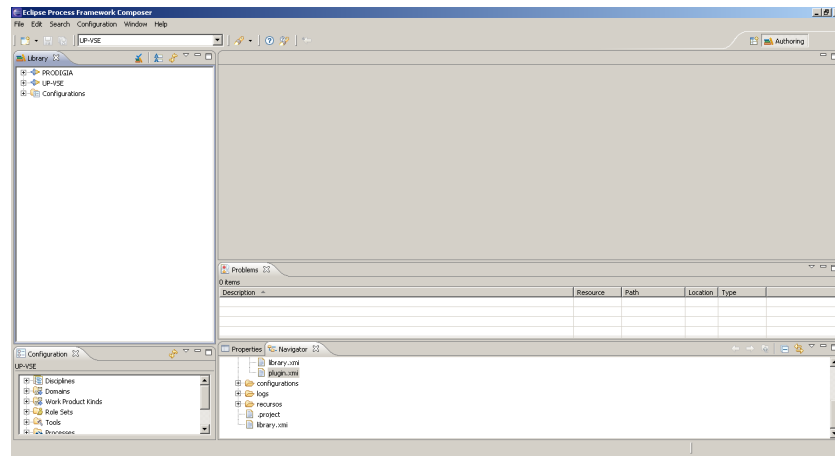


Figura 2.3: El entorno de trabajo EPF Composer.

- **MP 1.1.3 - Asegurarse que las prácticas base son implementadas, elaborando productos de trabajo que apoyan el logro de las salidas definidas del proceso:** para cada proceso, el alcance del trabajo es identificado en el proceso en ejecución.

## 2.7. Eclipse Process Framework Project (EPF composer)

EPF composer es una herramienta utilizada para la documentación de procesos de software soportando una amplia variedad de tipos de proyectos y estilos de desarrollo <sup>1</sup>.

Los objetivos del proyecto Process Framework son:

- Proveer un marco de trabajo extensible y herramientas ejemplares para métodos y autoría de procesos en la ingeniería de procesos de software, gestión de documentación, configuración y publicación de procesos.
- Proporcionar contenido ejemplar y extensible de procesos para una gama de procesos de desarrollo y gestión soportando desarrollo iterativo, ágil e incremental, aplicable a una amplia variedad de plataformas y aplicaciones de desarrollo.

EPF composer trabaja con el lenguaje SPEM 2.0, un meta-modelo de ingeniería de procesos de software utilizado como estándar para documentación<sup>2</sup>. Esta herramienta genera un archivo XML del proceso software en este meta-lenguaje el cual es útil para hacer un posterior análisis del proceso con AVISPA (ver sección 2.8). La figura 2.3 ilustra el entorno de trabajo de la herramienta EPF Composer.

---

<sup>1</sup><http://www.eclipse.org/epf/>

<sup>2</sup><http://projects.eclipse.org/projects/technology.epf>

## 2.8. Analysis and Visualization in Software Process Assessment (AVISPA)

AVISPA hace parte de un método asistido de análisis de procesos que brinda una analogía gráfica compuesta por nodos conectados entre si [38]. La estructura tecnológica de éste proyecto a grandes rasgos se soporta sobre la plataforma MOOSE<sup>1</sup> la cual fue diseñada para el análisis de software y permite la opción de instalar extensiones de funcionalidades tal como AVISPA<sup>2</sup> donde recibe un archivo XML que se obtiene como archivo exportado desde el plugin del proceso creado en EPF.

Las ventajas de utilizar esta herramienta principalmente radican en que es open source (MOOSE con licencia BSD y AVISPA con licencia MIT), arroja resultados de fácil interpretación y gran parte del conocimiento de la herramienta se encuentra en el grupo IDIS.

### 2.8.1. Salidas generadas por AVISPA

Después del procesamiento del archivo XML en notación SPEM 2.0, AVISPA produce 3 tipos de vistas: tareas, roles y productos de trabajo.

Cada vista contiene una serie de nodos rectangulares y líneas de interconexión. Específicamente en la vista de tareas, los nodos ofrecen información sobre el estado de cada sección del proceso que se necesita analizar, en efecto, cada nodo representa gráficamente una tarea y posee un conjunto de atributos que son:

- **Alto del rectángulo:** Indica el numero de productos de trabajo que la tarea recibe como entrada cuya relación es directamente proporcional a la distancia de esta dimensión
- **Ancho del rectángulo:** Indica el numero de productos de trabajo que la tarea genera como salida cuya relación es directamente proporcional a la distancia de esta dimensión
- **Color:** El color se vincula a la cantidad de roles involucrados en la tarea, la oscuridad del color tiene una relación es directamente proporcional a la cantidad de roles implicados.
- **Vínculos:** Estos indican la dependencia que hay de una tarea a otra que por lo general se asocian a los productos de trabajo del nodo de origen que se utilizan como entradas en el nodo de destino.

La figura 2.4 lo ilustra gráficamente.

---

<sup>1</sup><http://www.moosetechnology.org>

<sup>2</sup><http://www.moosetechnology.org/tools/ProcessModel>

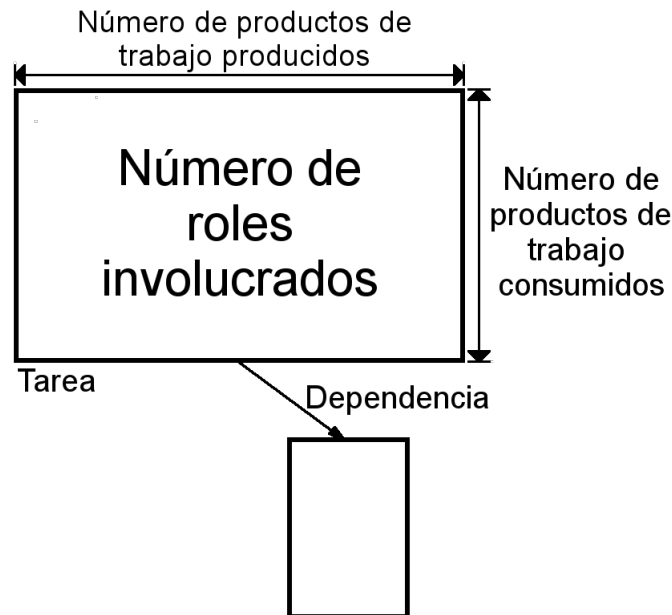


Figura 2.4: Atributos de los nodos considerados como métricas en la vista de tareas por AVISPA

### 2.8.2. Patrones de error identificados por AVISPA

AVISPA puede hallar patrones de error que generalmente se cometen al momento de definir un proceso de desarrollo de software para una entidad en particular. La tabla 2.1 ilustra estos patrones de error.

## 2.9. Trabajos relacionados

### *Comprensión e implementación de la norma*

Se han encontrado aportes con respecto a la norma como son los paquetes de despliegue (DP) [16, 17], estos toman aspectos o parte del estándar para enfocarse en un tema específico en la mejora de los procesos organizacionales de la VSE. Muchos de los paquetes de despliegue están diseñados de tal manera que sean de fácil comprensión con el fin de brindar una interiorización más rápida de los procesos de la ISO 29110. O'Connor et al. [11] presentan formalmente las DPs, donde se encuentran varios patrones sobre la adopción, brindando detalles de la incorporación del estándar ISO/IEC 29110 a través de los DPs. Esta validación se ha llevado a cabo proyectos piloto en diferentes partes del mundo, VSE de países como Canadá, Francia, Bélgica y otros más participaron en la utilización de estos paquetes con resultados satisfactorios. Aunque presenta los mecanismos de adopción, los paquetes de despliegue no presentan un modelo de proceso prescriptivo que sirva como punto de referencia o caso de estudio para una VSE. El trabajo de O'Connor et al. [24] se preocupa por el entendimiento de los temas que afectan la adopción de los estándares de procesos de software en VSE, particularmente el estándar ISO/IEC 29110 a través de entrevistas.

Patrón de error	Descripción	Localización	Identificación
No hay guías asociadas	Un elemento que no tiene guía asociada	Cualquier vista	Nodo azul
Rol sobrecargado	Un rol que se involucra demasiadas tareas	Vista de roles	Nodos que tienen mas de una desviación mas larga que la media
Rol aislado	Un rol que no colabora	Vista de roles	Un nodo que no está conectado con un borde
Tareas multipropósito	Tareas que producen muchos productos de trabajo	Vista de tareas	Nodos que tienen mas de una desviación mas alta que la media
Subproyectos independientes	Subgrafos independientes	Vista de tareas o de productos de trabajo	Subgrafos que no están conectados con un borde

Tabla 2.1: Patrones de error identificados por AVISPA

tas, cuestionarios y focus groups por medio de la interpretación de datos cualitativos y cuantitativos. Este trabajo sirve para realizar un mejor entendimiento de los temas relacionados con la adopción de los procesos del estándar en VSE.

El trabajo de O'Connor expresa la necesidad de facilitar el entendimiento de la norma en la que se centra esta tesis, este trabajo no es la excepción, y a diferencia de los anteriores trabajos, esta tesis se basa en una metodología de desarrollo de software bien conocida por la industria, el proceso unificado UP, que también tiene partes que cumplen con el estándar de calidad ISO / IEC 29110. Por tanto, el proceso UP-VSE pretende brindar un proceso concreto adaptado del proceso de desarrollo UP que satisfaga la norma ISO/IEC 29110 y que pueda ser reutilizada por una VSE para facilitar la adopción del estándar a través de la adaptación del Proceso Unificado.

### **Implementación de normas en VSE**

En el trabajo de Saliou et al. [15] brinda un aporte sobre las actividades de soporte de ingeniería de software para VSE sobre un proyecto piloto basado en web utilizando la presente norma con miras a identificar deficiencias u oportunidades de mejora de la norma ISO/IEC 29110, identificando deficiencias potenciales sobre actividades de soporte e infraestructura que hacen compleja la interiorización de la norma, análogamente, en el trabajo de O'Connor et al. [12] se explora el aspecto del compromiso de la gerencia acerca de la mejora de procesos de dos estudios similares en dos empresas irlandesas pequeñas y muy pequeñas, debido a que para muchas de las medianas y pequeñas compañías (SME) de software, la implementación de controles y estructuras para gestionar adecuadamente sus actividades de desarrollo de soft-

ware es el mayor reto. Los estudios se llevaron a cabo sobre tres factores: Mejora y evaluación de procesos, Compromiso y participación del personal y de la gestión y Planeación y objetivo del SPI, mostrando que las entidades intervenidas cambian y evolucionan sus procesos en el tiempo regularmente, son muy comprometidos en el SPI, por ende las investigaciones también muestran que las VSE pueden encontrar difícil la adopción de estándares ISO/IEC a sus necesidades de negocio y justificar su no adopción.

En el trabajo de O'Connor et al. [13] se realiza directamente la mejora de procesos (SPI) utilizando la norma sobre VSEs que, por su tamaño, no devengan grandes cantidades de dinero lo que hace que la adopción de normas, y por tanto sus mejoras sobre la competitividad se vean aplazadas, y si poseen los recursos no sienten un beneficio del retorno a la inversión al establecer procesos del ciclo de vida del software lo que conlleva a un rápido cambio de políticas para mantener sus negocios por otros 6 meses lo que termina siendo un desperdicio de recursos por mantener sus procesos actualizados. Para resolver este problema, se propone la utilización de los paquetes de despliegue (DP) que se consiguen de forma gratuita para agilizar el proceso de incorporación de la norma en la compañía. Con base en estos DPs se realizaron las pruebas con proyectos pilotos en VSE establecidas alrededor del mundo para reducir riesgos y aprender más acerca de los temas técnicos y organizacionales, un proyecto piloto implementado en una VSE en Francia señala que se han implementado correctamente los DP de los cuales han señalado que gracias a ellos han identificado potenciales procesos de infraestructura y soporte.

	<b>Deployment Packages [14, 15, 16, 17]</b>	<b>Interpretación basada en datos cualitativos y cuantitativos [24]</b>	<b>Implementación en proyectos piloto [16]</b>	<b>UP-VSE</b>
<b>Objetivo</b>	Implementar la norma ISO 29110 de manera más rápida en industrias VSE	Obtener datos sobre la adopción de estándares de procesos software en pequeñas organizaciones	Obtener aspectos específicos para la mejora de procesos organizacionales en VSE.	Implementar la norma ISO 29110 de manera más rápida en las industrias VSE aplicable también para industrias sin un ciclo de vida de desarrollo de software definido.
<b>Norma en la que se basa</b>	ISO/IEC 29110	ISO/IEC 29110	ISO/IEC 29110	ISO 29110-5-1-1 Sección de Ingeniería.
<b>Procesos que se utilizan en la implementación</b>	Todos los mencionados en la ISO/IEC 29110.	Todos los mencionados en la ISO/IEC 29110.	Todos los mencionados en la ISO/IEC 29110.	La mayoría de procesos basados en UP más la mayoría de procesos que incluye la ISO/IEC 29110-5-1-1 sección de ingeniería.
<b>Estrategia</b>	Dividir la norma en distintas secciones que son fáciles de entender y por ende implementar.	Utiliza asesoría de expertos a través de cuestionarios, entrevistas y focus groups para interpretar la norma	Hallar oportunidades de mejora a través de la ejecución de proyectos de prueba.	Utiliza un modelo de proceso software prescriptivo que acompaña la realización de los distintos procesos que sugiere la norma ISO 29110-5-1-1.
<b>Costo</b>	Medio	Alto	Alto	Medio

Tabla 2.2: Facilidades de Implementación de algunos modelos de procesos de software descriptivos.

Los anteriores trabajos muestran la importancia de la interpretación de las nor-

mas, en particular para las VSEs. Esta tesis intenta dar un paso más entregando un proceso específico que sirva de base para su reutilización y adaptación dentro de las pequeñas entidades, considerando que además de la interpretación de modelos como la ISO 29110 deben ser implementados, lo cual sigue siendo difícil para una pequeña entidad, de esta manera se ha realizado un cuadro comparativo de las diferentes implementaciones de normas sobre modelos de procesos descriptivos en relación con UP-VSE (Ver la tabla 2.2).

### ***Uso de UP con estándares de calidad en el desarrollo de software***

Alvarez et. al. [39] realizan un aporte sobre los diferentes modelos de proceso de software relevantes, en total, un estudio de 13 modelos, específicamente en modelos de ciclo de vida (como el modelo cascada) y modelos detallados de ciclo de vida (como el proceso unificado) a través de un meta-modelo de proceso de ciclo de vida, el artículo presenta recomendaciones para futuros trabajos y las limitaciones en la investigación. Lutteroth et al. [40] muestran un trabajo sobre la creación de un modelo de flujo de trabajo completo para una compañía de las TIC con el fin de alcanzar el nivel 3 de CMMI, especialmente, la compañía se centró en el esfuerzo de la integración del marco de trabajo de desarrollo del proceso unificado. Este trabajo enfatiza la necesidad del uso de UP con estándares de facto como CMMI demostrando así la importancia de esta metodología en la industria de desarrollo de software. Sägesser et al. [41] introducen un modelo de ciclo de vida iterativo (ILCM) para la compañía de tecnologías de la información Credit Suisse IT Switzerland basado en el Proceso Unificado de Rational (RUP), consideraciones como la brecha entre RUP y CMMI, en este caso nivel de madurez organizacional 3, ya que RUP no cubre todas las expectativas que exige CMMI ML3, la razón de este cambio es la necesidad de organización de procesos sobre una organización larga y compleja como Credit Suisse. ILCM ha sido aplicado alrededor de 25 proyectos alrededor de todos los departamentos de los cuales se describen 2 casos de estudio de este modelo de ciclo de vida iterativo aplicado.

Los trabajos anteriores se esfuerzan por obtener una implementación del modelo CMMI basándose en el UP. Esto refuerza el valor que tiene para las organizaciones procesos conocidos como el UP y más aún cuando vienen herramientas de soporte asociadas al proceso, por ello ha sido el proceso seleccionado en este trabajo de investigación, sin embargo dado el enfoque hacia las VSEs, el modelo utilizado por nuestra investigación es el ISO 29110.

### ***Utilización del estándar ISO en el RUP***

Jeyaraman et al. [42] presentan la experiencia en la reingeniería de una aplicación heredada en un sistema basado en la Web J2EE con modificación del Proceso Unificado de Rational (RUP) en un modelo de desarrollo in-situ a lo largo de los estándares ISO 9001 y CMM nivel 5. Esta aplicación tiene alrededor de 2500 componentes de código y un esfuerzo de 100 personas año. También demuestra que los procesos pueden ser mejorados con lecciones aprendidas en iteraciones iniciales. El trabajo anterior enfatiza en obtener una implementación de los estándares ISO basándose



en el UP. Esto refuerza el valor que tiene para las organizaciones procesos conocidos como el UP y más aún cuando vienen herramientas de soporte asociadas al proceso, por ello ha sido el proceso seleccionado en este trabajo de investigación, sin embargo dado el enfoque hacia las VSEs, el modelo utilizado por nuestra investigación es el ISO 29110.

También existen trabajos relacionados con la realización de evaluaciones utilizando la ISO 15504. La corporación Rational ha realizado una evaluación del RUP utilizando la norma ISO 15504, principalmente empleando para ello los niveles de capacidad describiendo textualmente a grandes rasgos el alcance de este modelo de ciclo de vida respecto a los objetivos de la ISO/IEC 12207 [25]. Esta evaluación toma como entradas todos los niveles de capacidad mencionados en la ISO/IEC 15504-2 y todos los atributos de proceso encontrados en la ISO/IEC 15504-5. Finalmente el resultado de la evaluación por atributos de proceso cumplidos por niveles es:

- **Nivel 1:** CUS.1, CUS.3, SUP.1, SUP.2, SUP.4, MAN.4 y ORG.2.1
- **Nivel 3:** CUS.2, ENG.2, ENG.3, ENG.4, ENG.5, SUP.5 y SUP.8
- Los demás atributos de proceso se encuentran en el nivel 0 de la ISO/IEC 15504

Estos trabajos han ilustrado la utilidad del modelo de ciclo de vida del Proceso Unificado en la industria del software y su principal necesidad de hallar oportunidades de mejora para las compañías.

#### ***Evaluaciones sobre el estandar ISO/IEC 29110-5-1-1***

El objetivo principal del trabajo realizado por M. Trujillo et. al. [43] es clarificar la brecha que existe entre MoProSoft nivel de capacidad 2 y la ISO/IEC 29110-5-1-2. Ambos modelos de proceso fueron mapeados y evaluados utilizando las métricas mencionadas en la ISO/IEC 15504-2 que determinaron el nivel de capacidad según los resultados de soporte matemático de la evaluación de éste trabajo. El artículo concluye que MoProSoft nivel 2 cubre el 85 % de las tareas y 94 % de los productos de trabajo. Esta evaluación abre las posibilidades para obtener una certificación internacional en corto tiempo en las empresas que adopten MoProSoft nivel 2 en sus procesos.

En este trabajo se basa el sistema de evaluación de la presente monografía, la cual se basa en el modelo de evaluación ISO/IEC 15504-5, haciendo uso de la ISO/IEC 12207 como norma intermediaria al mapeo propuesto en dicha sección de la norma.

El trabajo de V. Ribaud et. al. [44] ilustra como se determinó un modelo de evaluación de procesos (PAM) apto para los modelos de ciclo de vida de las VSE sobre la ISO/IEC 15504 tomando como base tanto la dimensión del proceso como del producto, este trabajo es parte fundamental para construir un PAM apto para la propuesta UP-VSE y su elaboración está ilustrada en el capítulo 3. Este trabajo pretende que tras la implementación de un perfil en una compañía ya esté aplicando el nivel 1 sin conocerlo.

**Artículos que contengan una revisión sistemática del impacto o uso de la norma en VSE**

Se encontró una investigación de Hurtado et al. [22] acerca de una implementación de CMMI usando una combinación de métodos ágiles de desarrollo de software, que representa un aporte significativo con el método utilizado para llevar a cabo dicha implementación análogamente al trabajo UP-VSE. Similarmente, el trabajo de Pino et al. [7] trata sobre una revisión sistemática hecha sobre trabajos de mejora de procesos software en micro, pequeñas y medianas empresas que muestra tendencias de estas entidades desarrolladoras de software en el uso de estándares de calidad hasta el 2006; Este trabajo no tiene registros sobre el uso de la norma ISO 29110 o cualquiera de sus partes en micro, pequeñas y medianas empresas.

Los anteriores trabajos evidencian la necesidad de la industria de mejorar sus procesos a través de estándares de renombre internacional aunque comparten en común el mismo problema en el que la mayoría de industrias manifiestan y en especial en VSE y es el entendimiento de estos estándares y, en efecto, algunos de estos artículos mencionan soluciones a estos problemas. La tabla 2.3 ilustra un cuadro comparativo con algunas de estas propuestas que intentan mejorar el entendimiento de estos estándares

	ILCM [41]	UP Modificado [42]	Scrum y XP [22]	UP-VSE
<b>Norma en la que se basa</b>	CMMI Nivel 3.	ISO/IEC 9000, ISO/IEC 9001, ISO/IEC 9004, CMMI Nivel 5.	CMMI Nivel 2.	ISO 29110-5-1-1 Sección de Ingeniería
<b>Facilidad de entendimiento</b>	Baja	Baja	Baja	Alta
<b>Dirigido a VSE</b>	No	No	No	Si
<b>Reutilización</b>	Media	Media	Alta	Media

Tabla 2.3: Implementaciones de normas a través de algunos modelos de procesos de software prescriptivos.

En las tablas 2.2 y 2.3, la primera columna son los criterios por los que han sido evaluadas cada una de las implementaciones tanto de las normas como de las metodologías de desarrollo, las demás columnas son los modelos de ciclo de vida más relevantes y anteriormente estudiados, la primera fila indica el tipo de implementaciones más importantes de la literatura, la primera fila indica los tipos de implementaciones que se han llevado a cabo, las demás filas son las características en los que se basa este estudio para calificar el grado de aproximación y definir diferencias de forma cualitativa entre cada modelo o implementación(es) de modelo(s) (columnas).

**Marco de referencia para el estudio de caso (Ver sección 5.2)**

Se han realizado varios proyectos llevados a cabo en el mundo real en donde se aplica la metodología de estudio de caso con el proceso unificado, uno de ellos es el trabajo de Renate Motschnig-Pitrik [45] que, en cooperación con una pequeña agencia de multimedia, utiliza esta metodología pero adaptada para el desarrollo de una aplicación web en donde también se enfocan en pasos y artefactos previamente seleccionados según la experiencia del autor, también discute los puntos fuertes

y débiles de cómo es percibido UP en el proyecto e intenta realizar una hipótesis generalizada y toca algunos puntos clave para una gestión de proyectos, para este caso, de 4 integrantes, y cabe resaltar que, siendo un proyecto para profesionales en pedagogía infantil, se enfocó en la lectura como parte de su metodología, evaluando por funcionalidad ofrecida los casos de uso implementando un enfoque de caja negra siguiendo la metodología propuesta por Schneider et. al. [46].

También han sido encontrados trabajos donde resulta necesario una adaptación del RUP a compañías de desarrollo de software. Borges et. al. [47] trabajan un mapeo de roles a equipos pequeños de desarrollo de software que será aplicado a un estudio de caso de un equipo certificado en CMMI nivel 3 definiendo una serie de roles base, en donde, para que sean considerados esenciales deben cumplir uno de tres condiciones mencionadas en el trabajo de Monteiro et. al. [48], para luego ser mapeados con los roles del RUP (39 en total), quedando reducidos a 13 de los cuales se han definido cuales pueden ser integrados entre RUP y los roles base y cuáles no, indicando en algunos casos (con una X en el mapeo) cuales roles deben ser absolutamente restringidos en una acumulación de los mismos en una sola persona.

Por otro lado, el proceso unificado es una metodología de desarrollo de software estricta en sus artefactos y en sus procesos, sin embargo, es necesario adaptar el proceso unificado a las necesidades particulares de una entidad desarrolladora de software, en donde, el trabajo de Hanssen et. al [49] que consiste en la adaptación del proceso unificado a proyectos según en el contexto en el que se realicen modificando así aspectos como la organización y el tipo de proyecto aplicado a una compañía noruega de consultoría con 50 empleados resultado exitosa su adaptación llevando su aprendizaje a partir de su estrategia KISS (Keep It Simple Stupid). Este trabajo concluye que el proceso unificado, a pesar de ser una metodología de desarrollo (por lo tanto, de más bajo nivel de abstracción que un estándar de desarrollo de software), sigue siendo un proceso que está en alto nivel y por lo tanto necesita ser adaptado a las organizaciones y proyectos por su contexto.

### **Conclusiones**

Debido a que la norma ISO / IEC 29110 es una norma relativamente nueva, se justifica el empeño por continuar esta investigación, por las características y necesidades de las VSEs y especialmente las regionales, constituyendo este esfuerzo como un punto de referencia y de soporte para proyectos e investigaciones futuras sobre aspectos de la calidad de estas entidades. Como se ha visto anteriormente, existen distintos métodos de aproximación tanto a la norma, como a UP, pero ambos tienen factores que los distinguen y los apartan como 2 modelos de ciclo de vida de software distintos. Existe otra problemática que ha sido en muchos casos el factor común con respecto a la adopción de normas en VSE y es su baja aceptación y prioridad que estos estándares tienen para ser implementados, principalmente esto se debe a su bajo presupuesto principalmente en aspectos como el tiempo, el personal y su bajo poder económico [24]. Existen otras razones como escasos requerimientos de cliente, nivel de detalle y de dificultad del proceso de la norma que actualmente está siendo implementado y la prioridad del desarrollo de software sobre la inversión de recursos en la gestión de la calidad del software [24].

## Capítulo 3

# Modelo de evaluación sobre el estándar ISO/IEC 29110-5-1-1 para los procesos basados en el Proceso Unificado

### 3.1. Introducción

Este capítulo ilustra el modelo de evaluación basado en la norma ISO/IEC 29110-5-1-1:2012[18] y su respectiva ejecución sobre el Proceso Unificado. Esta evaluación tiene como objetivo realizar un análisis del cubrimiento entre el Proceso Unificado y el estándar ISO/IEC 29110-5-1-1 con el fin de hallar elementos útiles del proceso para la implementación del modelo UP-VSE.

El modelo de evaluación fue definido de acuerdo a la sección ISO/IEC 29110-3, una guía de evaluación adjunta al grupo de normas de la ISO/IEC 29110 el cual referencia aspectos de diversas normas como la ISO/IEC 12207 y la ISO/IEC 15504-5 los cuales han sido adaptados a las exigencias y capacidades de las VSE.

La norma ISO/IEC 15504 propone una guía que especifica entradas, la ejecución y las salidas obtenidas. Se ha desarrollado una plantilla de evaluación en base al mapeo entre las normas ISO/IEC 29110-5-1-1 y la ISO/IEC 12207 en contraste con las actividades de los modelos de proceso involucrados en este proyecto (en el orden: UP, UP-VSE y PRODIGIA).

A parte de contar con la evaluación de actividades, este modelo de evaluación considera aspectos aparte de los mencionados en las guías de evaluación ISO/IEC 29110-3 e ISO/IEC 15504 tal como la evaluación de productos de trabajo o plantillas y equivalencias entre roles de los modelos de proceso respecto a las ISO/IEC 29110-5-1-1:2012.

Como ha sido mencionado anteriormente, el modelo de evaluación de la ISO/IEC 15504 propone una calificación haciendo uso de una escala de 1 a 5 sobre el proceso a evaluar cuya unidad indica su nivel de capacidad a través de atributos de proceso que son utilizados como indicadores dentro de la evaluación. Dado a que el proceso de evaluación mencionado en la ISO/IEC 29110-3 referencia al modelo de la ISO/IEC 15504 donde un perfil se considera alcanzado si el proceso cumple por lo menos el nivel 1 de capacidad respecto a un perfil básico de la ISO/IEC 29110-5 [1], la evaluación utiliza la misma escala definida en la ISO/IEC 15504 el cual se basa en el trabajo de [43] cuyo proceso de evaluación fue adaptado e instanciado en este trabajo (ver sección 3.2.1).

Las siguientes secciones ilustran el modelo de evaluación y su respectiva ejecución sobre los 3 modelos de proceso involucrados en esta tesis.

## 3.2. Evaluación del Proceso Unificado

El Proceso Unificado de I. Jacobson et. al. [2] es un modelo de proceso reutilizable y adaptable de software bien conocido por la industria y ampliamente aceptado por la comunidad académica [8]. Este trabajo utiliza esta ventaja del UP para definir un proceso lo suficientemente liviano pero completo para las VSE siguiendo el estándar ISO/IEC 29110-5-1-1 con lo cual se busca ahorrar recursos en su interpretación, capacitación e implementación para lo cual ha sido necesario realizar una evaluación que determine el grado de cobertura del Proceso Unificado respecto a la norma ISO/IEC 29110-5.

### 3.2.1. Guía de evaluación respecto a la norma ISO/IEC 29110-5-1-1

El Modelo de Evaluación del Proceso (PAM) que propone la ISO/IEC 29110:2012 se basa en la ISO/IEC 15504 y por lo tanto es un modelo de capacidad del proceso bi-dimensional. La dimensión de la capacidad define un conjunto de atributos de proceso en niveles de capacidad. Los atributos de proceso son las características del proceso que se pueden medir y que pueden brindar resultados sobre su capacidad. Los indicadores de la capacidad del proceso son las variables de los atributos de proceso y principales fuentes de información sobre la capacidad del proceso que han sido considerados dentro de la evaluación. La evidencia de los indicadores de la capacidad del proceso soporta el juicio del grado de completitud del atributo de proceso [1]. En el diseño de la evaluación se han tomado aportes que pueden ser encontrados en los trabajos de R. Motschnig-Pitrik [25] y G. Schneider et. al.[44].

#### 3.2.1.1. Planeación de la evaluación

Se ha realizado un plan de evaluación liviano conforme a las especificaciones del modelo de evaluación de la ISO/IEC 29110-3 para los perfiles básicos [1]. La subsec-

ción 6.3 de esta parte de la norma especifica que: "Los requisitos para el cumplimiento de los perfiles VSE pueden ser derivados de las respectivas partes ISO/IEC 29110-4 e ISO/IEC 29110-5". Como mínimo, todos los elementos obligatorios del perfil VSE, definidos en la ISO/IEC 29110-4 necesitan ser considerados en la evaluación, así por ejemplo, los procesos evaluados del perfil VSE básico necesitan alcanzar el nivel de capacidad 1 definido en la ISO/IEC 15504-2. Esto significa que el proceso implementado logra su propósito de proceso y sus salidas definidas"(Ver figura 2.2).

### 3.2.1.2. Modelo de evaluación de la ISO/IEC 15504

#### **Generalidades**

En el proceso de instanciación de la ISO/IEC 15504 se ha logrado establecer una configuración lógica de los elementos que participan en la evaluación, la figura 3.1 lo ilustra. En las siguientes subsecciones se ilustrarán estos elementos.

#### **Marco de trabajo y alcance de la ISO/IEC 15504**

Como ha sido mencionado anteriormente, se tomará en cuenta el nivel de capacidad 1 para cumplir con el perfil básico, por lo tanto ésta se acota a algunas categorías y atributos de proceso de la ISO/IEC 15504 y se conserva la escala de evaluación para los atributos de proceso encontrados en la ISO/IEC 15504-5 [37]:

- **Dimensión del proceso:** Para las siguientes evaluaciones se escogen las categorías del ciclo de vida del proceso
  - **Grupo de procesos de Ingeniería (ENG):**
    - ENG.1: *Proceso de análisis y diseño de los requisitos del sistema.*
    - ENG.2: *Análisis de los requisitos del software.*
- **Dimensión de la capacidad:**
  - **Niveles de capacidad:** Siguiendo el modelo de evaluación ISO/IEC 29110-3, los niveles de capacidad a evaluar son 0 y 1 descritos en la sección 2.6.
  - **Escala de calificación de los atributos de proceso:** Se utiliza la escala de la ISO/IEC 15504-2 [50].

#### **Medición en la dimensión de la capacidad:**

Dado a que la evaluación se hará en el primer nivel de capacidad y que por lo tanto solo se incluirá un solo atributo de proceso (ver sección anterior) se vinculan los respectivos indicadores de capacidad del proceso mencionados en la sección B.2.1 de la norma ISO/IEC 15504-5 [37].

#### **Medición en la dimensión del proceso:**

Fue necesaria la realización de un mapeo entre la ISO/IEC 12207:2008 y la ISO/IEC 29110-5-1-1 para asegurar equivalencias en la evaluación de la ISO/IEC 15504. (ver tabla 3.1 <sup>1</sup>).

---

<sup>1</sup>Los nombres de estas actividades se han acertado, se hallan completos en sus respectivas normas.

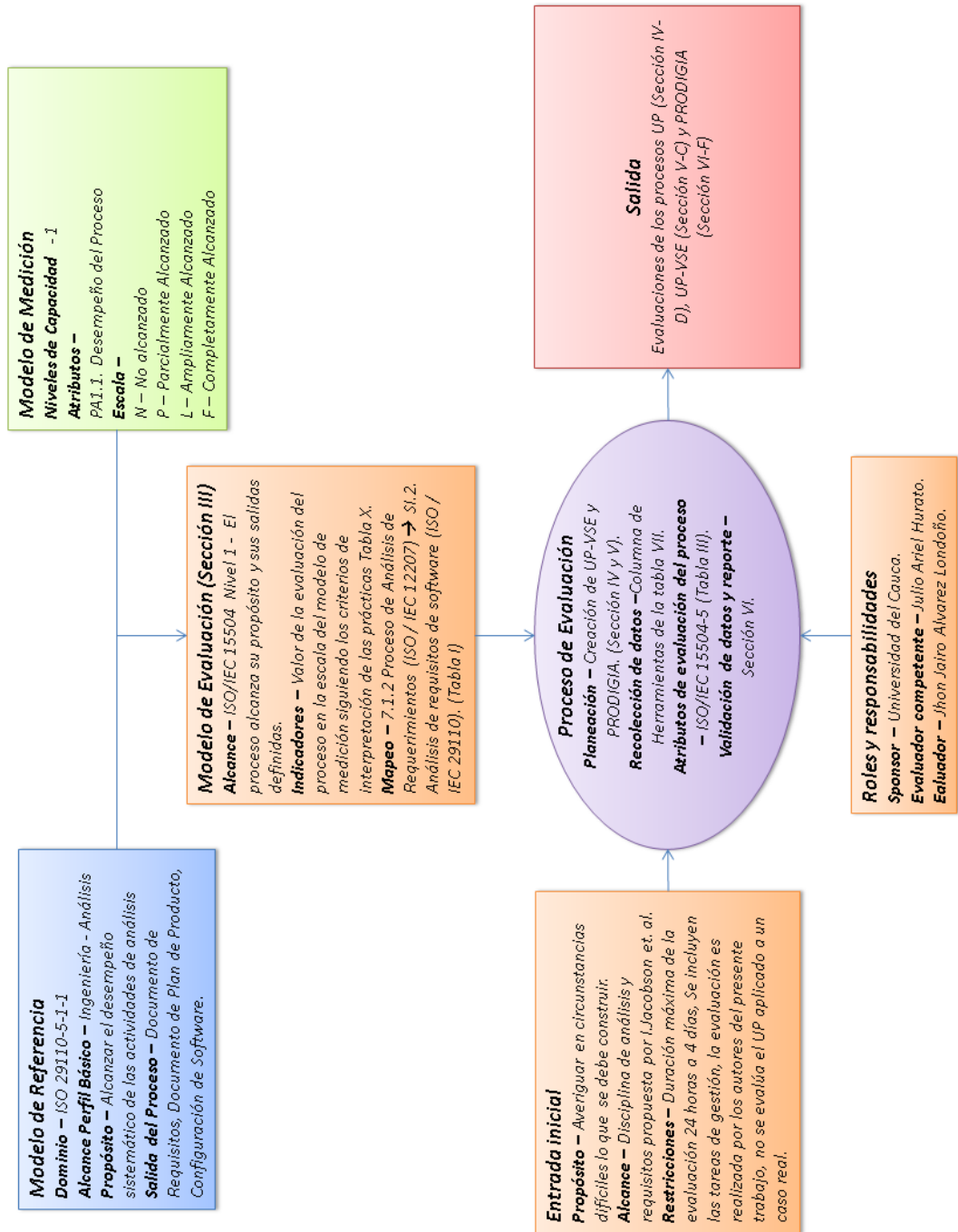


Figura 3.1: Elementos instanciados de la evaluación en ISO/IEC 15504-2.

ISO/IEC 12207	ISO/IEC 29110-5-1-1
7.1.1.3.1 Estrategia de implementación de software. 7.1.1.3.1.4 El implementador debe desarrollar planes para conducir las actividades de los procesos de implementación de software. <sup>1</sup> (Específicamente la asignación de roles de trabajo)	SI. 1 Inicio de la implementación software SI.1.1 Revisar el plan de proyecto actual con los miembros del equipo de trabajo. <sup>1</sup>
7.1.1.3.1.3 El ejecutor deberá seleccionar las herramientas necesarias para la realización de las actividades de los procesos de implementación de software y apoyo. <sup>1</sup> (Se vuelven opcionales el uso de los elementos que sugiere esta actividad con el fin de cumplir el mapeo)	SI. 1.2 Establecer o actualizar el entorno de implementación
7.1.2.3.1.1 Análisis de requisitos de software 6.3.1.3.2.1*. Preparar los planes para la ejecución del proyecto. <sup>1</sup> Subsección e: asignación de responsabilidades y 6.2.4.3.3.7* Otorgar a los equipos a realizar sus roles asegurándose de que estos hayan: (Subsección a) tenido un entendimiento de sus roles en el proyecto. <sup>1</sup>	SI. 2 Análisis de requisitos de software SI. 2.1 Asignar tareas a los miembros del equipo de trabajo en conformidad con su rol respectivo basándose en el plan de proyecto concurrente.
7.1.2.3.1.1.1 Establecer y documentar el reporte de especificación de requisitos. <sup>1</sup> (sin las características de calidad mencionadas en éste punto de la norma).	SI. 2.2 Documentar o actualizar los requisitos de software
7.1.2.3.1.2 Evaluar los requisitos de software. <sup>1</sup> (sin consideración de los criterios mencionados en éste punto de la norma).	SI. 2.3 Validar y obtener aprobación de la especificación de requisitos
7.1.3.3.1 Diseño arquitectónico de Software y 7.1.4.3.1 Diseño detallado de Software 6.3.1.3.2.1* y 6.2.4.3.3.7*	SI. 3 Identificación de componentes de software. SI.3.1* ↔ SI. 2.1* SI. 3.2 Entender la especificación de requerimientos.
7.1.3.3.1.1 El implementador deberá transformar los requisitos de software en la arquitectura en la que se identifican los componentes software. <sup>1</sup>	SI. 3.3 Documentar o actualizar la identificación de componentes de software. <sup>1</sup>
7.1.3.3.1.2 El implementador deberá desarrollar y documentar un diseño de interfaces externas de alto nivel del item de software. <sup>1</sup>	
7.1.4.3.1.1 El implementador deberá desarrollar un diseño detallado para cada item del componente software. <sup>1</sup>	
7.1.5.3.1 Construcción de software 6.3.1.3.2.1* y 6.2.4.3.3.7*	SI. 4 Construcción de software. SI.4.1* ↔ SI. 2.1* SI. 4.2 Entender la identificación de componentes de software. 4.3 Construir o actualizar los componentes software
7.1.5.3.1.1 El implementador deberá desarrollar y documentar: (Subsección a) cada unidad de software y de base de datos y (Subsección b) procedimientos y datos para pruebas de cada unidad de software y base de datos. <sup>1</sup>	SI. 4.4 Establecer o actualizar casos y procedimientos de prueba para cada unidad e integración de software basándose en la especificación de requisitos y la identificación de componentes de software. <sup>1</sup>
7.1.5.3.1.1.1 El implementador debe desarrollar un plan de integración para unir los componentes de software. <sup>1</sup>	SI. 4.5 Probar los componentes de software. SI. 5 Integración de software y pruebas. SI.5.1* ↔ SI. 2.1*
7.1.5.3.1.2 El implementador debe probar cada unidad de software y base de datos. <sup>1</sup>	
7.1.6.3.1 Integración de software, 7.2.8.3.2 Resolución de problemas y 7.2.2.3.3 Control de la configuración 6.3.1.3.2.1* y 6.2.4.3.3.7*	SI. 5.2 Entender los casos y procedimientos de prueba. Establecer o actualizar el entorno de pruebas. <sup>1</sup> SI.5.3 Integrar el software usando los componentes y actualizar los casos y procedimientos de prueba. SI.5.4 Llevar a cabo las pruebas de software usando los casos y procedimientos de prueba para integración. <sup>1</sup>
6.1.2.3.4.5 Desarrollo del plan de proyecto incluyendo: (Subsección b): entorno de pruebas. <sup>1</sup>	
7.1.6.3.1.2 Integrar los componentes software y realizar las pruebas acorde al plan de integración. <sup>1</sup>	
7.2.8.3.1.1 Un proceso de resolución de problemas debe ser establecido. <sup>1</sup>	
7.2.8.3.2.1 El informe de problemas deberá ser reportado, se utiliza desde la detección del problema, a través de la investigación y el análisis hasta su resolución. <sup>1</sup>	
7.2.2.3.3.1 Debe llevarse a cabo la identificación de los cambios, su análisis y evaluación, aprobación y negación del cambio y de la petición de cambio. <sup>1</sup>	SI. 5.6 Incorporar la especificación de requisitos y el software a la configuración de software SI. 6 Entrega del producto. SI.6.1* ↔ SI. 2.1*
6.1.1.3.6 Aceptación del adquisidor y 7.2.2.3.3 Control de la configuración 6.3.1.3.2.1* y 6.2.4.3.3.7*	
6.1.1.3.6.1 El adquisidor debe preparar la aceptación basándose en unos criterios y estrategia de ítems haciendo uso de casos, datos, entorno y procedimientos de prueba. <sup>1</sup>	SI. 6.2 Revisar para la comprensibilidad de la configuración de software.
6.1.1.3.6.2 El adquisidor debe conducir la revisión y pruebas de aceptación del producto software a entregar. <sup>1</sup>	6.3 Realizar la entrega al gestor de proyecto y apoyarla acorde al plan del proyecto.

\*Tareas repetidas, (en algunas se omite su descripción).

Tabla 3.1: Equivalencia entre procesos y tareas involucradas en el mapeo



ISO/IEC 12207	UP I. Jacobson et.al. [2] / UP-VSE
7.1.1.3.1 Estrategia de implementación de software	13.2 Comienzo de la fase de inicio
7.1.2.3.1 Análisis de requisitos de software	7 Disciplina de ingeniería de requisitos
7.1.3.3.1 Diseño arquitectónico de software y 7.1.4.3.1 Diseño detallado de software	8 Disciplina de análisis y 9 Disciplina de diseño
7.1.5.3.1 Construcción de software y 7.1.6.3.1 Integración de software	10 Disciplina de implementación
7.2.8.3.2 Resolución de problemas	11 Disciplina de pruebas

Tabla 3.2: Relación de subprocesos entre la norma ISO/IEC 12207 y UP

Valores de cobertura (C)	Porcentaje	Puntaje asignado (Q)
T - Totalmente	85 - 100 %	1.0
L - Ampliamente	50 - 85 %	0.7
P - Parcialmente	15 - 50 %	0.3
N - No logrado	0 - 15 %	0.0

Tabla 3.3: Rangos de capacidad y puntajes asignados

Para continuar la evaluación entre la ISO/IEC 12207 previamente mapeada con los procesos de la ISO/IEC 29110-5-1-1, se procede a determinar la equivalencia de procesos con el Proceso Unificado. La tabla 3.2 ilustra los subprocesos presentes en la norma ISO/IEC 12207 y el UP que harán parte de la evaluación con UP.

#### **Métricas y reglas de cobertura:**

Las métricas definidas en la ISO/IEC 15504-5 fueron usadas para determinar los valores de cobertura esenciales para este trabajo y basándose en los niveles de Cobertura (C). A cada valor en C se le ha sido asignado un puntaje utilizado para definir un nivel Cuantitativo (Q). La tabla 3.3 ilustra la relación entre los valores de la evaluación según la ISO/IEC 15504-5 y el puntaje asignado en el trabajo de M. Trujillo et. al.[43].

Tomando los indicadores de capacidad de proceso filtrados (Ver sección Medición de la capacidad) junto a los subprocesos mapeados (Ver sección Medición del proceso) y utilizando la ISO/IEC 12207 como un modelo de referencia de procesos intermediario se procede a definir la evaluación:

- El nivel de cobertura (C) y el cuantitativo (Q) aplica para las tareas ( $C_T$  ó  $Q_T$ ), productos de trabajo ( $C_W$  ó  $Q_W$ ) y el proceso ( $C_P$  ó  $Q_P$ )
- Para hallar el nivel de cobertura de un proceso se aplica la siguientes formulas generales:

$$C_T(Q_T[Tareas]) = C_T \left( \frac{\sum Q_T[C_T\{T_i\}]}{|T|} \right)$$

y

$$C_W(Q_W[Productos_de_trabajo]) = C_W \left( \frac{\sum Q_W[C_W\{W_i\}]}{|W|} \right)$$

en donde:

- $T$  y  $W$ : Es el conjunto de tareas y productos de trabajo del proceso respectivamente.
  - $T_i$  y  $W_i$ : Es la  $i$ -ésima tarea y el  $i$ -ésimo producto de trabajo del proceso respectivamente.
  - $|T|$  y  $|W|$ : Es el número total de las tareas y productos de trabajo del proceso respectivamente.
  - $\sum Q_T[C_T\{T_i\}]$ : Es la suma de cada uno de los niveles cuantitativos (puntuajes) de las tareas del proceso.
  - $\sum Q_W[C_W\{W_i\}]$ : Es la suma de cada uno de los niveles cuantitativos (puntuajes) de los productos de trabajo del proceso.
- Un ejemplo de la aplicación de esta fórmula puede ser encontrado en el trabajo de M. Trujillo et. al.[43].
  - Cada puntaje tanto de tareas y productos de trabajo se evaluaron por aparte por lo tanto darán distintas perspectivas de  $C_P$  y  $Q_P$ .
  - No se considera el tiempo en la ejecución tanto de tareas como de artefactos en la evaluación.

#### **Metodología de evaluación:**

Basándose en la estructura metodológica propuesta en la ISO/IEC 15504 [37], se ha diseñado un modelo de evaluación siguiendo una serie de tareas:

- *Análisis de los modelos de ciclo de vida*: esta actividad involucra las siguientes tareas: (i) obtener trabajos sobre los modelos a evaluar, (ii) analizar su estructura y (iii) buscar las plantillas de los productos de trabajo asociadas a cada modelo de ciclo de vida.
- *Diseño de la evaluación*: esta actividad involucra las siguientes tareas: (i) definir la dirección de la evaluación, (ii) realizar el mapeo de la evaluación para la ISO/IEC 15504, en este caso entre la ISO/IEC 12207 y la ISO/IEC 29110-5-1-1 (iii) diseñar la plantilla de evaluación (tareas relacionadas con la captura de requisitos de la norma ISO/IEC 29110-5-1-1 y los distintos modelos de ciclo de vida involucrados en conjunto con las plantillas asociadas a cada modelo si existen), (iv) definir escalas de evaluación y (v) verificar la coherencia y dirección de la evaluación y de los resultados.
- *Ejecución de la evaluación*: esta actividad involucra las siguientes tareas: (i) realizar la equivalencia entre procesos de la ISO/IEC 12207 y los modelos de ciclo de vida involucrados en las evaluaciones (ii) comparar los modelos de ciclo de vida de la evaluación conforme a las métricas en la sección "Métricas y reglas de cobertura" siguiendo las prácticas escogidas en "Descripción de los niveles de capacidad", (iii) llenar los campos de la plantilla en blanco con los valores de cobertura mencionados en "Métricas y reglas de cobertura", (iv) resolver discrepancias entre evaluadores y corregir la evaluación y (v) verificar y validar los resultados obtenidos de esta actividad.

- *Presentación y análisis de las salidas de la evaluación.*

Específicamente, la ejecución de cada una de las 3 evaluaciones ha sido realizada como se describe a continuación:

- Se toman los subprocesos en el orden que aparecen en la norma para la evaluación, y por cada subproceso se realizan los siguientes pasos:
  - Se asigna un valor de evaluación (Ver sección Medición en la dimensión del proceso) que indica el nivel alcanzado de la implementación de las familias y derivaciones del UP propuestas en este trabajo sobre una tarea o artefacto específico del subproceso escogido de la norma siguiendo las buenas prácticas (MP) mencionadas en la ISO/IEC 15504-5 mencionadas en la sección "Descripción de los niveles de capacidad" del presente trabajo. Para las tareas se evalúan en base a su significado semántico y los productos de trabajo corroborando su existencia con las plantillas del RUP y las características de cada producto de trabajo según la norma ISO/IEC 29110-5-1-1.
  - Al final de la evaluación de cada subproceso se socializan y documentan los resultados y se realizan todas las correcciones pertinentes.
- La mayoría de tareas de algunos subprocesos de la norma contienen aspectos relacionados con la gestión de proyectos, para este caso, estos subprocesos han tomados como un solo conjunto para la evaluación y se ejecutan los mismos pasos del punto anterior.

### 3.2.1.3. Respetto a los artefactos de la ISO/IEC 29110-5-1-1

Como se menciona en la norma ISO/IEC 29110-5-1-1 cada tarea contiene productos de entrada y de salida, por otro lado la sección 9 de la presente norma indica la descripción de cada uno de los productos. Con base en estas sentencias se procede a realizar la evaluación de cada producto de trabajo mencionado en la ISO/IEC 29110-5-1-1 respecto a su descripción siguiendo también las buenas prácticas (MP) de los atributos de proceso de la ISO/IEC 15504-5. Por otro lado, los artefactos involucrados en el proceso de captura de requisitos de la ISO/IEC 29110-5 que se tomarán en cuenta para las evaluaciones son:

- **Plan de proyecto:** Presenta como el proceso y las actividades se ejecutaran para asegurar la culminación exitosa del proyecto y la calidad de los productos de trabajo.
- **Documento de especificación de requisitos:** Aquí se identifican los requisitos de software.
- **Identificación de componentes software:** Información textual y gráfica de la estructura del software y sus relaciones.

Disciplina	Numeración en I. Jacobson et. al. [2]	Nombre de la tarea
Ingeniería de requisitos	7.4.1.	Encontrar los actores y casos de uso.
	7.4.2.	Priorizar casos de uso.
	7.4.3.	Detallar un caso de uso.
	7.4.4.	Prototipar la interfaz de usuario.
	7.4.5.	Estructurar el modelo de casos de uso.
Análisis	8.6.1.	Análisis de la arquitectura.
	8.6.2.	Analizar un caso de uso.
	8.6.3.	Analizar una clase.
	8.6.4.	Analizar un paquete.
Diseño	9.5.1.	Diseño de la arquitectura.
	9.5.2.	Diseñar un caso de uso.
	9.5.3.	Diseñar una clase.
	9.5.4.	Diseñar un subsistema.
Implementación	10.5.1.	Implementación de la arquitectura.
	10.5.2.	Integrar sistemas.
	10.5.3.	Implementar un subsistema.
	10.5.4.	Implementar una clase.
	10.5.5.	Realizar prueba de unidad.
Pruebas	11.5.1.	Planificar prueba.
	11.5.2.	Diseñar prueba.
	11.5.3.	Implementar prueba.
	11.5.4.	Realizar prueba de integración.
	11.5.5.	Realizar prueba de sistema.
	11.5.6.	Evaluar prueba.

Tabla 3.4: Numeración de tareas en I. Jacobson et. al. [2]

- **Casos y procedimientos de prueba:** Identificación de los elementos que necesitan ser probados.
- **Reporte de pruebas:** Documenta la ejecución de pruebas.

Se han omitido los detalles de cada artefacto tomados en cuenta para la evaluación del UP y sus derivados tomados en cuenta en éste trabajo (UP-VSE y PRODIGIA). Más información puede ser encontrada en la ISO/IEC 29110-5-1-1 [18].

### 3.2.2. Ejecución y resultados del proceso de evaluación con UP

En ésta sección se presenta una evaluación entre los subprocesos de la norma ISO/IEC 29110-5-1-1 y los principales flujos de trabajo del UP tomando como intermediario la ISO/IEC 12207 con el fin de hallar los elementos de proceso faltantes del UP para cumplir con la norma, la evaluación se hace conforme a la guía de evaluación de la norma y bajo el marco de mediciones en la sección 3.2.1.

#### 3.2.2.1. Los flujos de trabajo del UP

Los pasos y actividades del Proceso Unificado según I. Jacobson et. el. [2] contienen una numeración que se conservará a lo largo de éste trabajo que será utilizada como identificador de cada tarea con fines de evaluación y documentación. Finalmente la tabla 3.4 ilustra esta configuración.

### 3.2.2.2. Actividades adicionales del UP involucradas en la evaluación

Adicionalmente a los flujos principales de trabajo del Proceso Unificado, la fase de inicio ilustra como se realiza la planificación del proyecto incluyendo la asignación de roles. por ejemplo la sección 13.2 del trabajo de I. Jacobson et. al [2] ilustra cómo sucede la planificación en el UP dejando evidencia de cobertura en la asignación de roles en el que la norma ISO/IEC 29110-5-1-1 lo enuncia como requisito en sus subprocesos según lo estipulado en la ISO/IEC 15504. Las secciones posteriores del trabajo enuncian como se relacionaron estos subprocesos (ver sección 3.2.2.4).

### 3.2.2.3. Observaciones durante la ejecución de la evaluación

Durante la realización de la evaluación algunas tareas de la norma han requerido de una sección de gestión de proyectos en el UP, para solucionar este requerimiento, se añadió esta disciplina de soporte que contiene todos los aspectos de gestión del Proceso Unificado propuestos en I. Jacobson et. al. [2].

### 3.2.2.4. Presentación y análisis de los resultados de la evaluación

#### ***Evaluación de las tareas de UP:***

La tabla 3.5 ilustra los resultados obtenidos de la evaluación del UP siguiendo los parámetros establecidos (Ver sección 3.2.1). Esta tabla contrasta el mapeo realizado entre la ISO/IEC 29110-5-1-1:2012 y la ISO/IEC 12207 en la tabla 3.1 (primeras dos columnas de izquierda a derecha) y cada una de las tareas del UP (primeras cuatro filas de arriba hacia abajo) en la evaluación propuesta por la ISO/IEC 29110-3 utilizando los parámetros del modelo de evaluación propuesto en el capítulo 3, esto significa que cada tarea del mapeo (en relación a la norma ISO/IEC 29110-5-1-1) ha sido validada con cada tarea del Proceso Unificado obteniendo los resultados de dicha tabla.. Obteniendo los resultados de  $Q_T$  se procede a utilizar las formulas previamente establecidas para hallar la cobertura de las tareas de UP, esto es:

$$Q_T[Tareas] = \frac{\sum C_T\{T_i\}}{|T|}$$

$$Q_T[Tareas] = \frac{(1*17)+(0.7*2)+(0.3*3)}{22}$$

$$Q_T[Tareas] = 0.87727$$

$$C_T(Q_T[Tareas]) = T = \text{Totalmente alcanzado.}$$

UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras de Software Basado en el Proceso Unificado

		Disciplinas del UP																								Q <sub>T</sub>								
		C <sub>T</sub>												Pruebas (11.5)																				
		Gestión del proyecto*						Ingeniería de requisitos (7.4)						Análisis (8.6)						Diseño (9.5)						Implementación (10.5)								
		16.4.5		13.2		13.3.2		16.4.1		17.3.3		.1		.2		.3		.4		.5		.1		.2		.3		.4		.5		.6		
Subprocesos ISO 12207 (ISO 29110-5-1-1)	Tareas ISO 12207 (ISO 29110-5-1-1)																																	
	7.1.1.3.1 Estrategia de implementación de software (SI. 1)	T		P																														1
7.1.2.3.1 Análisis de requisitos de software (SI. 2)	7.1.1.3.1.4 (SI. 1.1)																																	0.3
	7.1.1.3.1.3 (SI. 1.2)	T																																1
7.1.2.3.1 Análisis de requisitos de software (SI. 2)	6.3.1.3.2.1 y 6.2.4.3.3.7 (SI. 2.1)																																	1
	7.1.2.3.1.1 (SI. 2.2)																																	1
7.1.3.3.1 Diseño arquitectónico de software y 7.1.4.3.1 Diseño detallado de software (SI. 3)	7.1.2.3.1.2 (SI. 2.2, SI. 2.3)																																	0.3
	6.3.1.3.2.1 y 6.2.4.3.3.7 (SI. 3.1)	T																																1
7.1.5.3.1 Construcción de software (SI. 4)	7.1.3.3.1.1 (SI. 3.2 y SI. 3.3)																																	1
	7.1.3.3.1.2 (SI. 3.3)																																	1
7.1.6.3.1 Integración de software. 7.2.8.3.2 Resolución de problemas y 7.2.2.3.3 Control de la configuración (SI. 5)	7.1.4.3.1.1 (SI. 3.3)																																	1
	6.3.1.3.2.1 y 6.2.4.3.3.7 (SI. 4.1)	T																																1
7.1.6.3.1 Integración de software. 7.2.8.3.2 Resolución de problemas y 7.2.2.3.3 Control de la configuración (SI. 5)	7.1.5.3.1.1 (SI. 4.2, SI. 4.3 y SI. 4.4)																																	1
	7.1.5.3.1.2 (SI. 4.4)																																	1
6.1.1.3.6 Aceptación del adquiridor y 7.2.2.3.3 Control de la configuración (SI. 6)	7.1.5.3.1.2 (SI. 4.5)																																	1
	6.3.1.3.2.1 y 6.2.4.3.3.7 (SI. 5.1)	T																																1
6.1.1.3.6 Aceptación del adquiridor y 7.2.2.3.3 Control de la configuración (SI. 6)	6.1.2.3.4.5 (SI. 5.2)																																	1
	7.1.6.3.1.2 (SI. 5.2 y SI. 5.3)																																	1
6.1.1.3.6 Aceptación del adquiridor y 7.2.2.3.3 Control de la configuración (SI. 6)	7.2.8.3.1.1 (SI. 5.4)																																	1
	7.2.8.3.2.1 (SI. 5.5)																																	1
6.1.1.3.6 Aceptación del adquiridor y 7.2.2.3.3 Control de la configuración (SI. 6)	7.2.2.3.3.1 (SI. 5.6)	L																															0.7	
	6.3.1.3.2.1 y 6.2.4.3.3.7 (SI. 6.1)	T																																1
6.1.1.3.6 Aceptación del adquiridor y 7.2.2.3.3 Control de la configuración (SI. 6)	6.1.1.3.5.1 (SI. 6.2)			L																														0.7
	6.1.1.3.6.2 (SI. 6.3)																																	

\* El Proceso Unificado de I. Jacobson et. al [2] no contiene una disciplina de gestión de proyectos definida formalmente pero se especifica con fines de organización.

Tabla 3.5: Evaluación de las tareas de UP respecto al estándar ISO/IEC 29110-5-1-1:2012

Artefactos		$C_W$	$Q_W$
ISO/IEC 29110-5-1-1	UP I. Jacobson et. al. [2]		
Plan del proyecto	Plan de desarrollo de software	T	1
Documento de especificación de requisitos	Modelo de casos de uso, Documento de casos de uso, Prototipo de interfaz de usuario	T	1
Identificación de componentes software	Documento de arquitectura de software[51]	T	1
Casos y procedimientos de prueba	Guías para la ejecución de pruebas, Casos de prueba[51]	T	1
Reporte de pruebas	Resumen de las pruebas de evaluación[51]	T	1

Tabla 3.6: Evaluación de los artefactos del UP y UP-VSE respecto al estándar ISO/IEC 29110-5-1-1:2012

#### **Evaluación de los artefactos de UP:**

Como se ha mencionado en anteriores secciones de éste trabajo, los artefactos del RUP han sido descargados para complementar la carencia de plantillas que soporten al UP de Jacobson et. al.[2]. Una serie de plantillas gratuitas han sido descargadas de [51] y tomadas en cuenta dentro de la evaluación. La tabla 3.6 muestra la calificación otorgada a los distintos artefactos de UP. Cada plantilla ha sido evaluada respecto a la descripción hecha en la ISO/IEC 29110-5-1-1. Se procede a utilizar las mismas formulas de la anterior evaluación para calcular la cobertura de los productos de trabajo, esto es:

$$Q_T[\text{Productos\_de\_trabajo}] = \frac{\sum Q_T[C_T\{T_i\}]}{|T|}$$

$$Q_T[\text{Productos\_de\_trabajo}] = \frac{1*5}{5}$$

$$Q_T[\text{Productos\_de\_trabajo}] = 1$$

$$C_T(Q_T[\text{Productos\_de\_trabajo}]) = T = \text{Totalmente alcanzado.}$$

ISO/IEC 29110-5-1-1:2012		UP I. Jacobson et. al.[2]
Customer	CUS	Cliente
Project Manager	PM	Jefe de proyecto
Work Team	WT	Analista de sistemas
		Especificador de casos de uso
		Diseñador de interfaces de usuario
		Arquitecto
		Ingeniero de casos de uso
		Ingeniero de componentes
		Ingeniero de pruebas
		Integrador de sistemas
		Ingeniero de pruebas de integración
		Ingeniero de pruebas de sistema

Tabla 3.7: Equivalencia de roles entre ISO/IEC 29110-5-1-1 y UP

### 3.2.3. Consideraciones adicionales

#### 3.2.3.1. Equivalencia de roles

En el orden de completar el mapeo, la tabla 3.7 ilustra una equivalencia entre los roles de la ISO/IEC 29110-5-1-1 y el Proceso Unificado de I. Jacobson et. al.[2] tomando en cuenta sus habilidades y competencias como principal criterio.

#### 3.2.3.2. Acerca de la ejecución del proceso de evaluación en UP

Debido a que esta evaluación fue realizada por los autores de esta tesis, es posible esperar un cierto nivel de subjetividad al momento de su ejecución por lo que podría afectar la confiabilidad de los datos obtenidos. Este proceso de evaluación tuvo como esfuerzo aproximado un total de 142 horas / hombre cuya duración fue de 3 meses aproximadamente.

## 3.3. Análisis de los resultados

El proceso unificado tomado como fuente el trabajo de I. Jacobson et. al. [2] es un modelo de proceso ampliamente utilizado y se evidencia en la evaluación por la gran cantidad de elementos de proceso que cubren la mayoría de aspectos de la norma ISO/IEC 29110-5-1-1.

Este modelo propende por una fuerte documentación técnica, abarcando todos los aspectos necesarios para llegar a un correcto tratamiento de los requisitos, modelado, implementación y entrega de los artefactos correspondientes.

A pesar de su fuerte documentación en el aspecto del desarrollo de software, realizando un análisis mas detallado de los resultados se han podido evidenciar las



siguientes falencias:

- **En relación a la proporción de industria:** El Proceso Unificado presenta una cantidad y tamaño en sus elementos de proceso que están sobredimensionados para las VSE. Por ende se hace necesario realizar una reducción para la propuesta de desarrollo de software UP-VSE.
- **Carece de plantillas:** La documentación exclusiva de Jacobson et. al. [2] no contiene plantillas de los artefactos que este modelo propone, por lo que se utilizaron plantillas gratuitas del RUP que contienen los campos necesarios que necesitan UP, UP-VSE y PRODIGIA para la evaluación de artefactos (ver [51] para descargarlas)
- **Descuida algunos aspectos de soporte:** El Proceso Unificado, a pesar de tener una gran fortaleza en aspectos de desarrollo técnico, carece de una formalización adecuada de procesos que brindan soporte según la norma ISO/IEC 29110-5-1-1, algunos de ellos son:
  - Tareas definidas para la verificación de requisitos.
  - Tareas sobre la formalización del control de la configuración de software.

Las anteriores características reflejan una serie de variables de utilidad para la definición del proceso UP-VSE, esto ayuda a obtener un espectro mas amplio del Proceso Unificado encontrando fortalezas y debilidades para tratar de lograr un equilibrio aproximado en la propuesta que propone una orientación mas cercana a las VSE.

## Capítulo 4

# UP-VSE

### 4.1. Introducción

El modelo de proceso de referencia propuesto por este trabajo se ha denominado UP-VSE en el que principalmente se conserva la misma filosofía del Proceso Unificado, sin embargo, se han adaptado sus actividades a gran parte de las orientaciones propiciadas por la norma ISO/IEC 29110-5-1-1:2012 teniendo en cuenta los acercamientos teóricos y prácticos a través del estudio de caso.

UP-VSE ha sido creado con el fin de representar una aproximación de un modelo de proceso de gran trayectoria tanto académica [8] como industrial [52] como lo es el UP hacia las necesidades de las VSE y brinda una amplia flexibilidad sobre su configuración lo cual permite que sea adaptable a las necesidades de las organizaciones VSE sin que se pierda su esencia como Proceso Unificado.

La definición de este modelo de proceso ha sido documentado en el lenguaje SPEM 2.0 bajo el entorno de programación Eclipse Process Framework Composer (EPF) como herramienta CASE usada como soporte para su especificación formal.

### 4.2. Características del proceso

- **Guiado por casos de uso, centrado en la arquitectura, iterativo e incremental:** Al igual que el trabajo I. Jacobson et. al. [2], UP-VSE cuenta con éstas 3 características que ilustran partes del ciclo de vida del proceso desde una vista mas general mostrando tanto su flujo de ejecución como de la interrelación de los artefactos mas importantes. La iteración brinda una vista mas precisa de lo que se está trabajando en un instante de tiempo dentro del proyecto.
- **Utiliza disciplinas y fases como vistas del proceso:** Las disciplinas en UP-VSE ilustran lo que el usuario realizará para cumplir con el proceso, las fases brindan una vista de gestión, ilustran en que parte del proceso se encuentra el proyecto y los principales entregables al final de cada una.

- **Aborda flexibilidad en la elaboración de sus artefactos:** Para tomar acciones sobre la naturaleza cambiante y única de cada proyecto, UP-VSE puede especializarse a diferentes tipos de proyectos a través de la elaboración opcional de algunos artefactos gracias a la filosofía iterativa e incremental que permite la planificación de la iteración antes de su comienzo y las vistas de proceso por fases y disciplinas del Proceso Unificado.
- **Simple:** Al contener pocos roles y tareas acordadas con los participantes, UP-VSE es un proceso fácil de entender y contiene artefactos mínimos para la realización del proceso. También se incluye un lenguaje simple sobre la explicación de cada tarea y ejemplos sencillos de los artefactos más importantes.
- **Mismas actividades, diferentes fases:** Existen tareas que se repiten en diversas fases, éste proceso aprovecha esta ventaja a su favor, evitando el re-trabajo de memorizar e interiorizar nuevas actividades y configuraciones de los elementos del proceso.
- **Aproxima a la norma ISO/IEC 29110-5-1-1:** Debido a la reutilización de algunos de los elementos de proceso de la ISO/IEC 29110-5-1-1, UP-VSE es un modelo de proceso que pretende orientarse y adaptarse a las necesidades de las VSE. Por otra parte, conserva algunos de sus elementos de proceso como sus roles excepto el rol de equipo de trabajo que se divide en más roles tal y como en el Proceso Unificado de I. Jacobson et. al. [2], en colaboración de algunos artefactos y buenas prácticas dado a que considera todos los subprocesos de éste perfil de la norma.

### 4.3. Construcción del proceso UP-VSE

La documentación principal del Proceso Unificado para esta tesis fue realizada por I. Jacobson et. al. [2]. UP ha sido documentado manteniendo un modelo de proceso paralelo en todas las disciplinas al igual que UP-VSE, pero difieren en la configuración de sus respectivos elementos de proceso. Esto es, que la documentación de I. Jacobson et. al. [2] provee una configuración secuencial de actividades y tareas en disciplinas, sin embargo, UP-VSE lo hace a través del ciclo de vida.

El modelo de proceso UP-VSE ofrece una serie de plantillas sobre las cuales realizar los distintos artefactos para el proyecto software [51] además de diversos ejemplos para lograr una adecuada documentación [53].

Las secciones posteriores ilustran la configuración de esta propuesta de modelo de ciclo de vida.

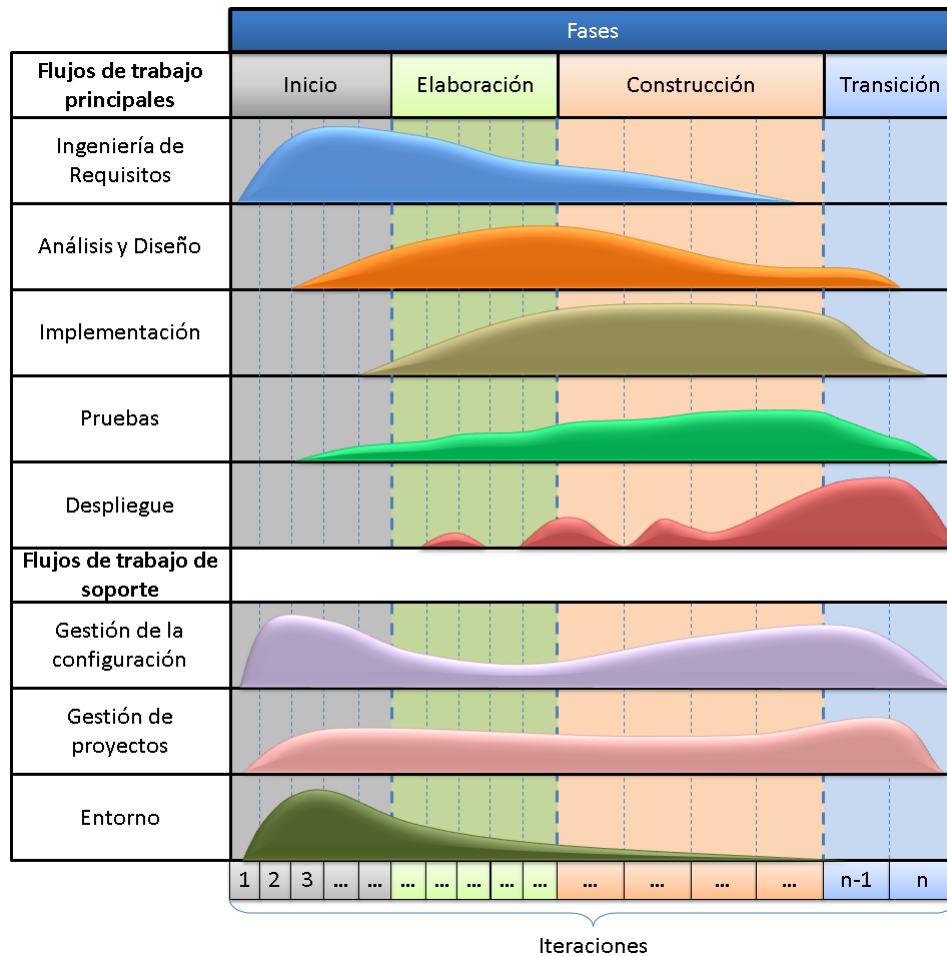


Figura 4.1: Ciclo de vida de UP-VSE

#### 4.4. Ciclo de vida de UP-VSE

El ciclo de vida de UP-VSE brinda una visión sobre el esfuerzo invertido en la ejecución del proyecto dividido en disciplinas, y, al igual que el Proceso Unificado de I. Jacobson et. al. [2] consta de 4 fases (**inicio**, **elaboración**, **construcción** y **transición**). La figura 4.1 ilustra detalladamente el ciclo de vida en 2 vistas (disciplinas y fases). Esta vista del ciclo de vida brinda información general sobre el estado del desarrollo en el proyecto a través del sobre los cuales se pueden construir parámetros y criterios para futuros cambios y mejoras sobre el proceso.

UP-VSE está dividido en 2 tipos de disciplinas o flujos de trabajo: los **flujo de trabajo principales** los cuales abordan aspectos técnicos de la construcción y ejecución del software y los **flujos de trabajo de soporte** que se encargan del seguimiento y control del proyecto sobre sus actividades básicas, artefactos entre otros. Para más información sobre las disciplinas de UP-VSE ver la sección 4.4.2.

#### 4.4.1. El proceso UP-VSE:

Las fases de UP-VSE, al igual que el Proceso Unificado son **Inicio**, **Construcción**, **Transición** y **Pruebas**, la figura 4.2 ilustra detalladamente su configuración.



Figura 4.2: Fases de UP-VSE

1. **La fase de Inicio:** esta fase pone en marcha el proyecto realizando el análisis del negocio sobre el que se justifica la continuación o no del proyecto de desarrollo. Debido a la inherente inestabilidad en el inicio del proyecto se pueden presentar diversas formas de comenzar su ejecución. La figura 4.3 ilustra la fase de inicio que propone UP-VSE.



Figura 4.3: Fase de Inicio

- 1.1. **Actividad Inicializar el proyecto:** En esta actividad se definen diversos artefactos como la construcción de la visión, la preparación del entorno para el proyecto y las primeras estrategias que garantizan continuidad en su desarrollo. La figura 4.4 ilustra la configuración de esta actividad.

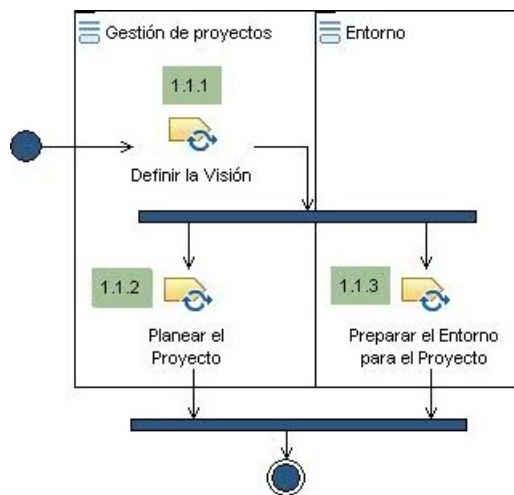


Figura 4.4: Actividad Inicializar el proyecto

- 1.2. **Iteraciones de la fase de inicio:** Principalmente en estas iteraciones se explora la viabilidad del proyecto en términos de la lógica del negocio y de la solución software. La figura 4.5 ilustra las actividades que componen las iteraciones<sup>1</sup>.

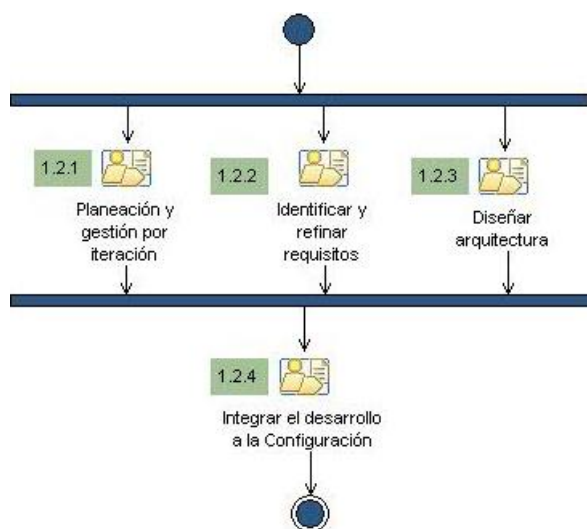


Figura 4.5: Iteraciones de la fase de inicio

<sup>1</sup>La diferentes actividades en paralelo ilustradas en esta figura delegan al practicante del proceso a organizar su propia configuración de las mismas

1.2.1. **Actividad Planeación y gestión por iteración:** Esta actividad consiste en establecer parámetros de entrada para la ejecución de la iteración y velar por su gestión, ejemplos de estos parámetros son el numero de tareas a realizar sobre la construcción de software, tiempo estimado de la iteración sin entrar en retrasos y otras que puedan ser consideradas al momento de su ejecución. La figura 4.6 ilustra éste proceso.

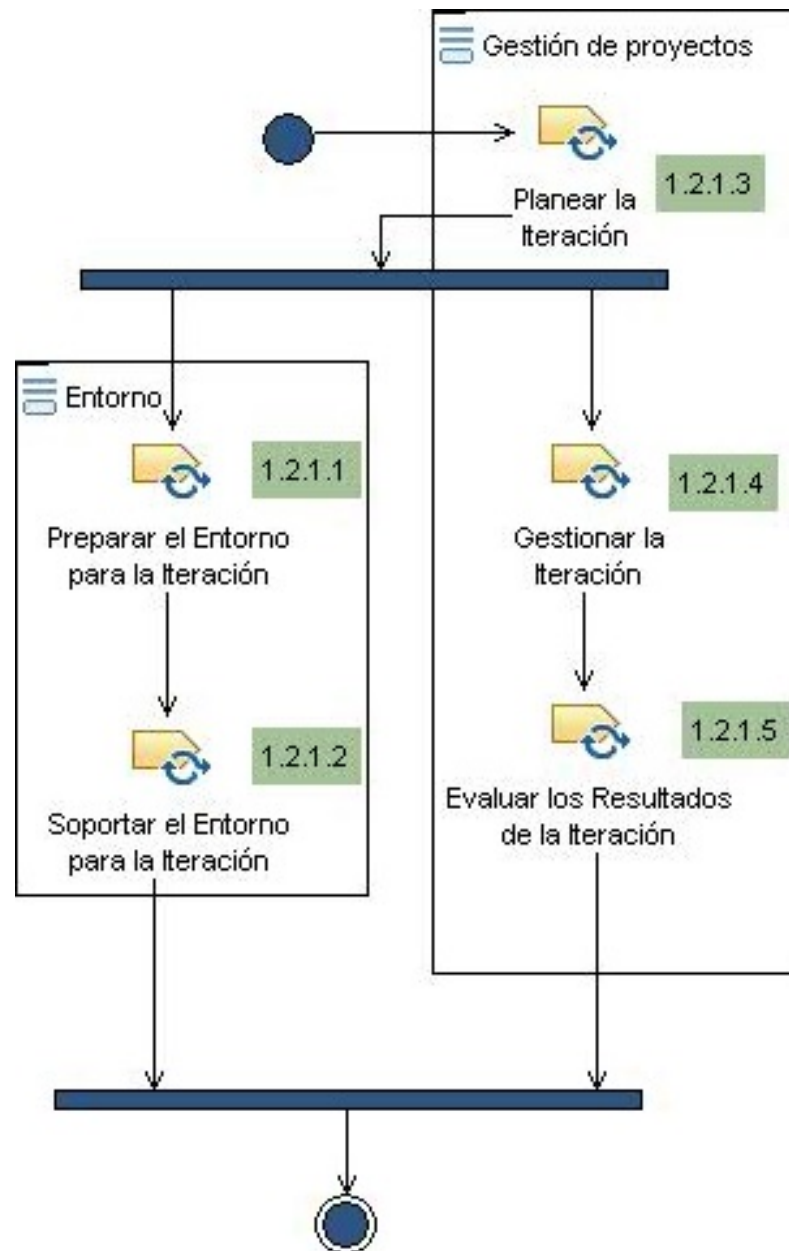


Figura 4.6: Actividad planeación y gestion por iteración

1.2.2. **Actividad Identificar y refinar requisitos:** Basándose en la filosofía de la centralización sobre los casos de uso, UP-VSE también hace uso de esta práctica, sin embargo, UP-VSE trata los casos de uso y su respectivo modelo como partes del documento de especificación de requerimientos mencionado en la norma ISO/IEC 29110-5-1-1. La figura 4.7 ilustra la configuración de tareas contenidas en esta actividad.

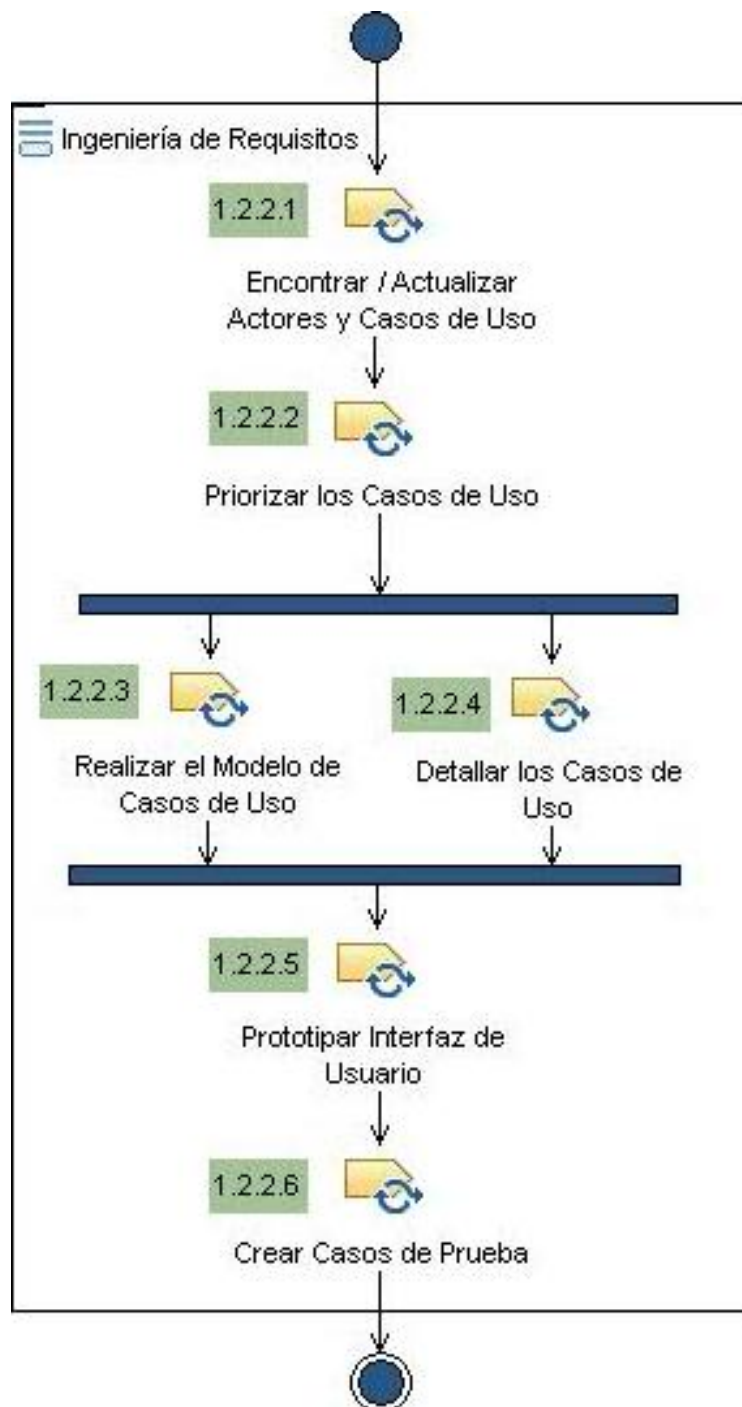


Figura 4.7: Actividad Identificar y refinar requisitos.



1.2.3. **Actividad Diseñar la arquitectura:** Dependiendo de los modelos que se necesiten realizar al interior de una organización UP-VSE, se propone en esta actividad un conjunto de tareas sencillas que pretenden adaptarse a los modelos de software que pueden ser utilizados en el proyecto en general. La figura 4.8 ilustra las de esta actividad.

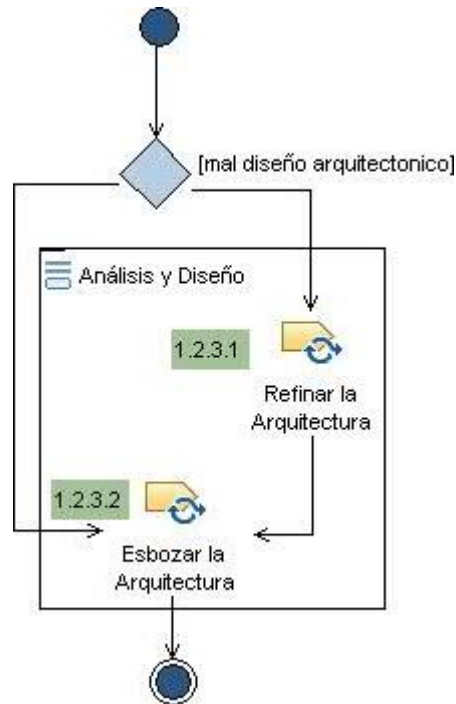


Figura 4.8: Actividad Diseñar arquitectura.

1.2.4. **Actividad integrar componentes desarrollados a una nueva versión de la solución:** Esta actividad vela por la integración de los componentes desarrollados en la iteración a una nueva versión de la solución. La figura 4.9 ilustra el contenido de esta actividad.

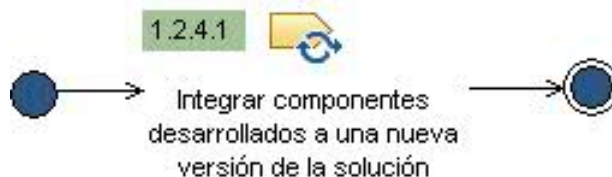


Figura 4.9: Actividad Integrar el desarrollo a la configuración

2. **La fase de Elaboración:** Al igual que el Proceso unificado de I. Jacobson et. al. [2], la fase de elaboración define la línea base de la arquitectura junto con los riesgos más importantes. La figura 4.10 ilustra el contenido de esta fase.

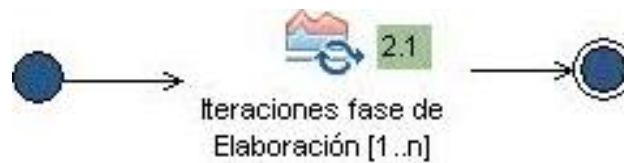


Figura 4.10: Fase de Elaboración

- 2.1. **Iteraciones de la fase de elaboración:** Aquí comienza a reforzar conceptos de desarrollo que pudieron haber quedado iniciados o sin terminar en la fase de inicio, se contemplan aspectos de pruebas y desarrollo de software por incrementos. La figura 4.11 ilustra la configuración de tareas de estas iteraciones<sup>1</sup>.

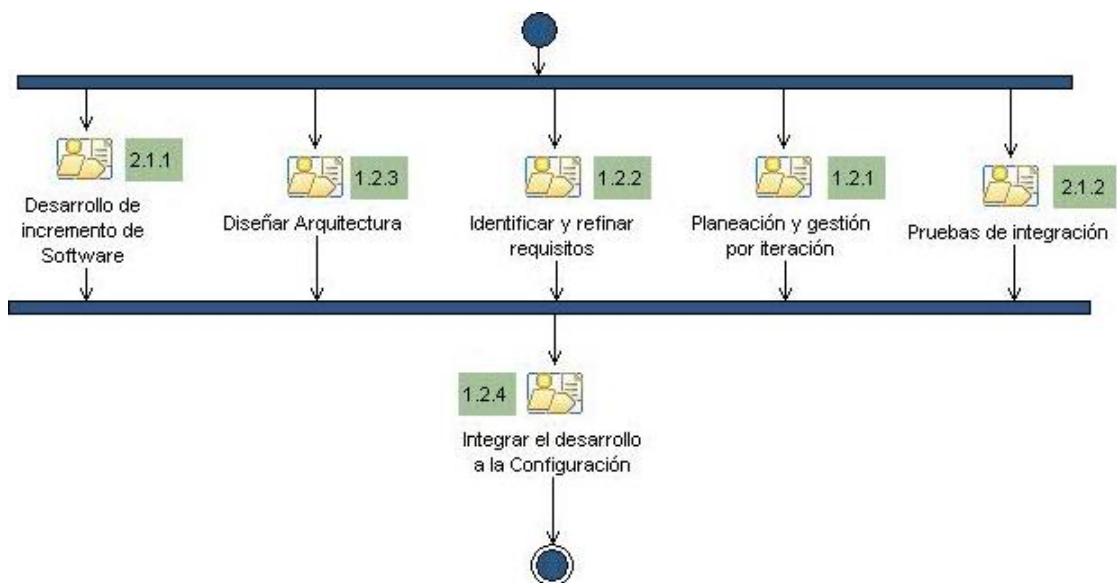


Figura 4.11: Iteraciones de la fase de Elaboración

2.1.1. **Actividad Desarrollo de incremento de software:** Esta actividad consiste en la construcción de software obteniendo como resultado un incremento por componente desarrollado. La figura 4.12 ilustra la configuración de tareas de esta actividad.

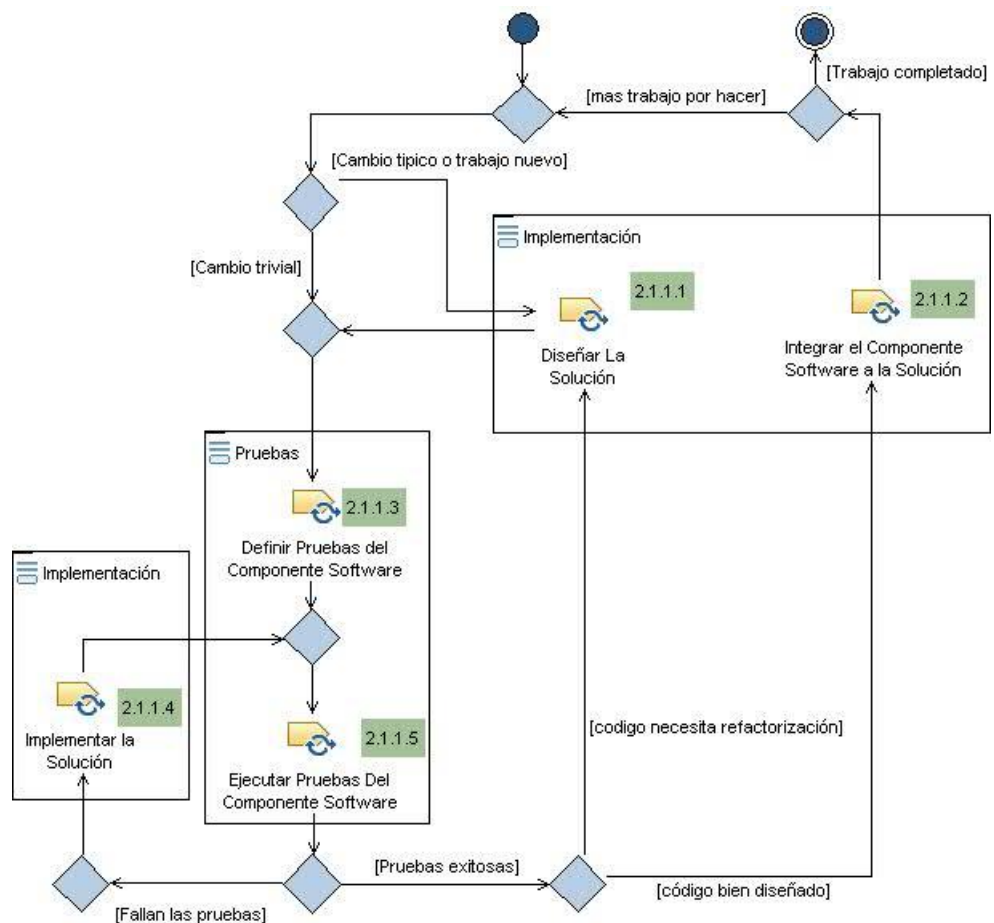


Figura 4.12: Actividad Desarrollo de incremento de software

2.1.2. **Actividad Pruebas de integración:** Esta actividad define una serie de tareas cuyo objetivo consiste en realizar las pruebas de integración sobre el o los componentes desarrollados, si la integración no conlleva gran dificultad se puede continuar con la su respectiva creación y ejecución de pruebas de integración y por último su implementación. Si por el contrario la solución en la integración de componentes de software no es tan trivial es recomendable elaborar una estrategia coordinada con el equipo de desarrollo diseñando una solución para la integración. La figura 4.13 ilustra este proceso.

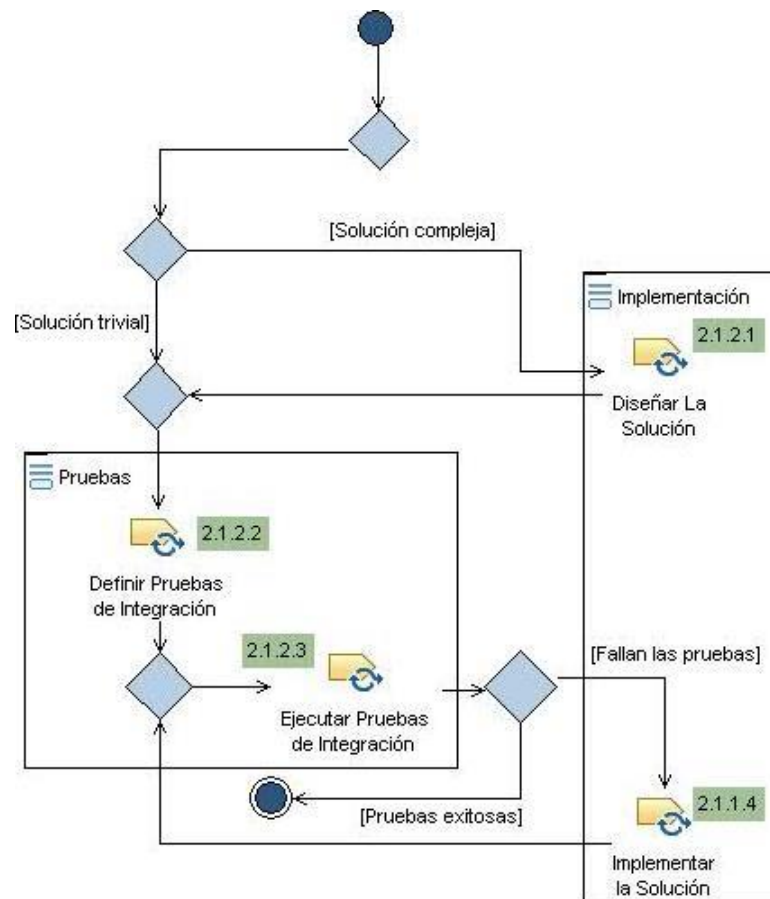


Figura 4.13: Actividad Pruebas de integración

3. **La fase de Construcción:** El objetivo de la fase de construcción consiste en que, a su término, se obtenga la capacidad operativa inicial del software, al igual que el Proceso Unificado de I. Jacobson et. al. [2]. La figura 4.14 ilustra la configuración de esta fase.

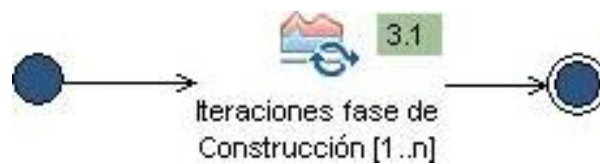


Figura 4.14: Fase de Construcción

- 3.1. **Iteraciones de la fase de construcción:** Estas iteraciones contienen los mismos pasos que las iteraciones de la fase de elaboración, la figura 4.11 lo ilustra<sup>1</sup>.

4. **La fase de Transición:** Finalmente la fase de transición completa la versión del producto tal como lo enunciado en el trabajo de I. Jacobson et. al. [2] una vez el producto haya alcanzado todos los requisitos y cumpla con las expectativas del cliente se realiza la entrega del producto implementando el sistema en el sitio operativo del cliente. La figura 4.15 ilustra el contenido de esta fase.



Figura 4.15: Fase de Transición

4.1. **Iteraciones de la fase de transición:** Debido a que es necesaria la planeación de la finalización del proyecto, se define un plan de despliegue de software y se disminuyen las actividades de captura de requisitos y de modelado de los componentes de software. La figura 4.16 ilustra la red de actividades de estas iteraciones<sup>1</sup>.

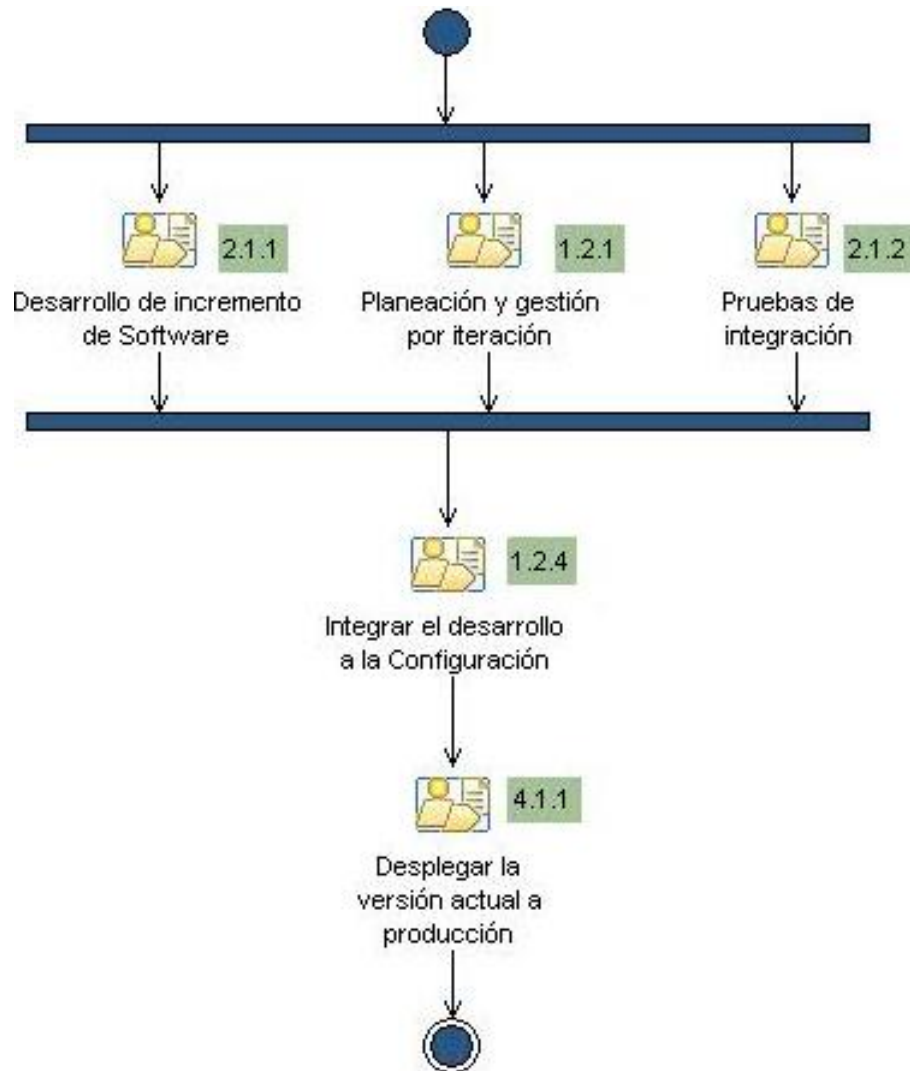


Figura 4.16: Iteraciones de la fase de Transición

4.1.1. **Actividad Desplegar la versión actual a producción:** Esta actividad se encarga de realizar entregas sobre el producto software, adaptando el despliegue a las necesidades particulares de cada proyecto basándose en el análisis del modelo de negocio y realizando la respectiva validación y verificación de lo realizado. La figura 4.17 ilustra en detalle la configuración de tareas de esta actividad.

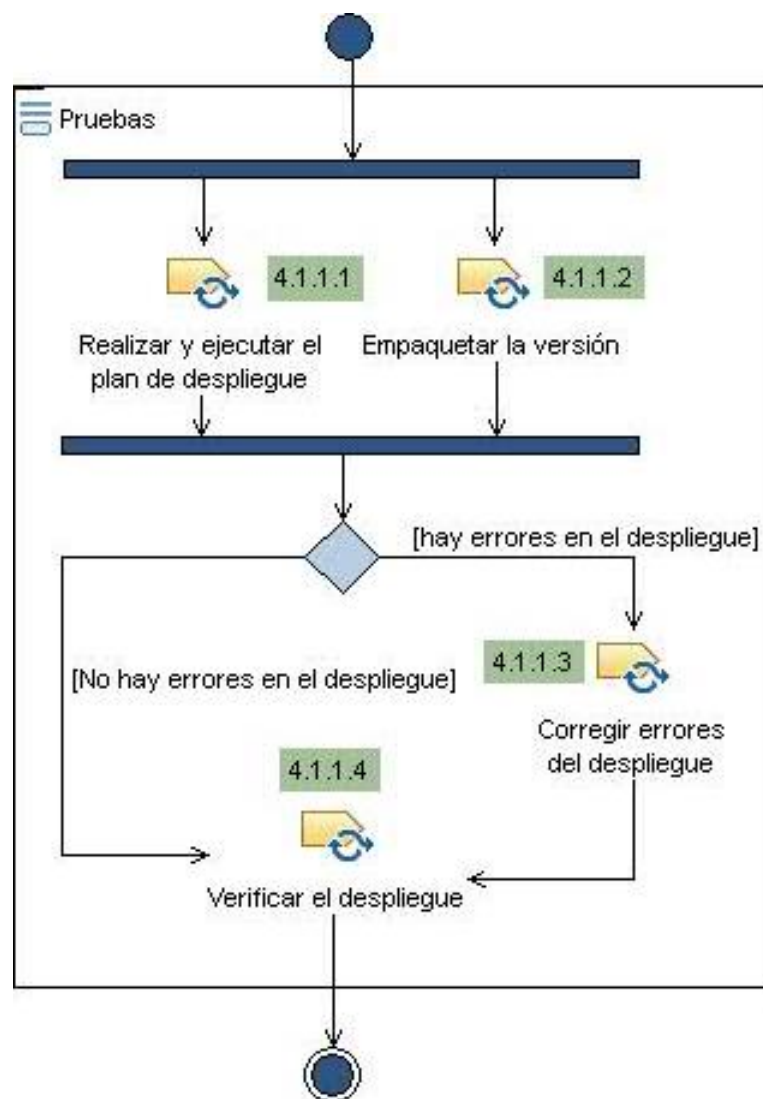


Figura 4.17: Actividad Desplegar la versión actual a producción

#### 4.4.2. Disciplinas de UP-VSE

Al igual que el Proceso Unificado de I. Jacobson et. al. [2], el conjunto de tareas del UP-VSE descritas anteriormente se encuentran agrupadas en disciplinas que ayudan a dividir la carga cognoscitiva del proceso ejecutadas en paralelo (figura 4.1). Las disciplinas o flujos de trabajo en UP-VSE se dividen en:

- **Disciplinas principales:** Cubren los aspectos del desarrollo de software, principalmente la captura de requisitos pasando por su desarrollo y terminando en la producción y se asegura de la creación de los artefactos básicos del software. Estas disciplinas son:
  - **Requisitos:** Inicia el proyecto y realiza la elicitación y procesamiento de requisitos en funcionalidades de software ampliamente detalladas.
  - **Análisis y Diseño:** Se encarga de la investigación de la solución software, de las bases teóricas que soporten su funcionamiento y del modelado de la solución en planos arquitectónicos que brinden detalles exactos de su estructura.
  - **Implementación:** Se encarga de llevar los conceptos y planos a código fuente manteniendo buenas prácticas de programación con el objetivo asegurar la escalabilidad y mantenibilidad del código software.
  - **Pruebas:** Las pruebas mantienen el código libre de errores de todo tipo, aseguran de que el código fuente sea funcional, cumpla los requisitos pactados desde la disciplina de ingeniería de requisitos y aseguran la trazabilidad con los artefactos involucrados en la dirección del desarrollo.
  - **Despliegue:** Se encarga de asegurar el paso del estado de desarrollo al estado de producción del software, se encarga de culminar las pruebas, las validaciones y verificaciones de software y el proyecto en general.
- **Disciplinas de soporte:** Ofrecen acompañamiento al proceso y complementan la entrega de artefactos adicionales que se necesitan para una culminación mas detallada del desarrollo de software. Estas disciplinas son:
  - **Gestión de la configuración:** Soporta la calidad del producto software en cualquier etapa de su desarrollo a través de un estricto control de cambios realizados sobre el mismo y de la disposición de las versiones del producto en todo el tiempo de desarrollo de software.
  - **Gestión de proyectos:** Se encarga de mantener las tareas desarrolladas al día en el cronograma de actividades, en la administración eficiente de recursos y en la culminación exitosa del proyecto.
  - **Configuración del entorno:** Se refiere a las herramientas y cambios en la configuración del proceso conforme a las exigencias en el momento de la ejecución del proyecto.



#### 4.4.3. Criterios para la definición UP-VSE

La definición del proceso UP-VSE fue determinada por una serie de características que principalmente variaban en incluir aspectos de:

- **La norma ISO/IEC 29110-5-1-1:** debido a que la finalidad del proceso UP-VSE consiste en aportar un modelo de ciclo de vida para las VSE.
- **El Proceso Unificado:** principalmente el proceso UP-VSE trata de conservar los 3 aspectos mas importantes del UP (iterativo e incremental, guiado por casos de uso, centrado en la arquitectura) y que conserve también la construcción de modelos arquitectónicos en UML[54]
- **Buenas prácticas de las metodologías ágiles (opcional):** El proceso UP-VSE también se elaboró pensando en la necesidad de incorporar buenas prácticas metodológicas ágiles. Permite el uso de reuniones de pie de 15 minutos máximo de SCRUM a modo de planeación de la iteración controlando el desarrollo incremental, estas prácticas no están sujetas a su ejecución y se mencionan en esta tesis a modo de sugerencia.

A su vez, UP-VSE ha pasado por un conjunto de filtros que determinaron que aspectos de ambos modelos de proceso debería conservar, estas condiciones en orden de importancia están sujetas a modificar elementos del proceso que:

- **Brindaran construcción inmediata:** Se conservaron subprocesos tanto de la norma como del UP que brindaran una pronta construcción del software siguiendo buenas prácticas y sin que aporten una documentación demasiado extensa principalmente de las plantillas.
- **No aporten a alcanzar la norma ISO/IEC 29110-5-1-1:2012:** Dado que las VSE tratan de alcanzar una certificación de esta norma, los elementos de proceso que no brinden una construcción rápida de software y que no aporten a alcanzar la norma ISO/IEC 29110-5-1-1:2012 pueden ser excluidos del proceso
- **No aporten a un desarrollo rápido:** se pretende que los artefactos que se produzcan en este modelo de proceso sean los mínimos para lograr un desarrollo adecuado en un software. Temporalmente los elementos de proceso que produzcan artefactos de poca utilidad para un proyecto permitan ser pasados por alto.

El proceso UP-VSE se enfoca en la elaboración de artefactos utiles para un proyecto de desarrollo de software por lo que descuida aspectos estructurales del UP y de ISO/IEC 29110-5-1-1.

## 4.5. Evaluación de UP-VSE con el proceso de evaluación

Al igual que en las anteriores evaluaciones se procede a realizar la evaluación utilizando los mismos elementos de proceso de la sección 3.2.1 como parámetros.

### 4.5.1. Ejecución y resultados del proceso de evaluación con UP-VSE

#### 4.5.1.1. Clasificación de tareas por disciplinas de UP-VSE

Cada actividad dentro de UP-VSE se le ha asignado un indicador numérico con fines de evaluación y documentación. La tabla 4.1 ilustra la numeración de estas actividades.

Disciplina	Numeración en UP-VSE	Nombre de la tarea
Ingeniería de requisitos	1.2.2.1.	Encontrar / Actualizar actores y casos de uso.
	1.2.2.2.	Priorizar casos de uso.
	1.2.2.3.	Realizar el modelo de casos de uso.
	1.2.2.4.	Detallar los casos de uso.
	1.2.2.5.	Prototipar interfaz de usuario.
	1.2.2.6.	Crear casos de prueba.
Análisis y Diseño	1.2.3.1.	Refinar la arquitectura.
	1.2.3.2.	Esbozar la arquitectura.
Implementación	1.2.3.1.	Diseñar la solución.
	2.1.1.2.	Integrar los componentes de software a la solución.
	2.1.1.4.	Implementar la solución.
Pruebas	2.1.1.3.	Definir pruebas del componente software.
	2.1.1.5.	Ejecutar pruebas del componente software.
	2.1.2.2.	Definir pruebas de integración.
	2.1.2.3.	Ejecutar pruebas de integración.
Despliegue	4.1.1.1.	Realizar y ejecutar el plan de despliegue.
	4.1.1.2.	Empaquetar la versión.
	4.1.1.3.	Corregir errores en el despliegue.
	4.1.1.4.	Verificar el despliegue.
Gestión de la configuración	1.2.4.1.	Incluir la documentación realizada en la monografía.
Gestión de proyectos	1.1.1.	Definir la visión.
	1.1.2.	Planear el proyecto.
	1.2.1.3.	Planear la iteración.
	1.2.1.4.	Gestionar la iteración.
	1.2.1.5.	Evaluar los resultado de la iteración.
Configuración del entorno	1.1.3.	Preparar el entorno para el proyecto.
	1.2.1.1.	Preparar el entorno para la iteración.
	1.2.1.2.	Soportar el entorno para la iteración.

Tabla 4.1: Numeración de tareas de UP-VSE

#### 4.5.1.2. Evaluación de las tareas de UP-VSE

Los resultados de la evaluación se presentan en la tabla 4.2.



La tabla 4.2 contrasta el mapeo realizado entre la ISO/IEC 29110-5-1-1:2012 y la ISO/IEC 12207 en la tabla 3.1 (primeras dos columnas de izquierda a derecha) y cada una de las tareas del proceso UP-VSE (primeras cinco filas de arriba hacia abajo) en la evaluación propuesta por la ISO/IEC 29110-3 utilizando los parámetros del modelo de evaluación propuesto en el capítulo 3, esto significa que cada tarea del mapeo (en relación a la norma ISO/IEC 29110-5-1-1) ha sido validada con cada tarea del proceso UP-VSE obteniendo los resultados de dicha tabla.

Al igual que en las evaluaciones de UP y PRODIGIA, se realiza la evaluación de las tareas del proceso UP-VSE siguiendo los parámetros indicados en 3.2.1. Esto es:

$$Q_T[Areas] = \frac{\sum Q_T[C_T\{T_i\}]}{|T|}$$

$$Q_T[Areas] = \frac{(1*19)+0.7+0.3}{22}$$

$$Q_T[Areas] = 0.901$$

$$C_T(Q_T[Areas]) = T = \text{Totalmente alcanzado.}$$

#### 4.5.1.3. Evaluación de los productos de trabajo de UP-VSE

Dado a que los artefactos son los mismos definidos para la evaluación del proceso unificado y las plantillas del RUP encontradas en [51], la tabla 3.6 contiene la misma evaluación de artefactos del UP-VSE.

#### 4.5.1.4. Relación de los roles de UP-VSE e ISO/IEC 29110-5-1-1

La tabla 4.3 ilustra la relación entre roles de UP-VSE y sus respectivos equivalentes de la norma ISO/IEC 29110-5-1-1.

ISO/IEC 29110-5-1-1:2012		UP-VSE
Customer	CUS	Cliente
Project Manager	PM	Jefe de proyecto
Work Team	WT	Analista de sistemas
		Analista del Sistema
		Analista de Pruebas
		Arquitecto de Software
		Diseñador de Interfaz de Usuario
		Diseñador de Pruebas
		Especialista en Herramientas
		Especialista en Pruebas
		Especialista en Requisitos
		Implementador de Software
		Integrador
		Jefe del Control de Cambios
Revisor Técnico		

Tabla 4.3: Equivalencia de roles entre ISO/IEC 29110-5-1-1 y UP-VSE

#### 4.5.1.5. Acerca de la ejecución del proceso de evaluación en UP-VSE

Al igual que en la evaluación de UP (ver sección 3.2). También es posible esperar un cierto nivel de subjetividad al ejecutar este modelo de evaluación con UP-VSE debido a que esta evaluación fue realizada por los autores de esta tesis por lo que podría afectar la confiabilidad de los datos obtenidos. Este proceso de evaluación tuvo como esfuerzo aproximado un total de 115 horas / hombre cuya duración fue de 3.5 meses aproximadamente.

### 4.6. Evaluación de UP-VSE con AVISPA

La evaluación con AVISPA (Analysis and Visualization in Software Process Assessment) [38] se realiza como un segundo método de validación dentro de la evaluación del proceso UP-VSE.

Debido a que SPEM 2.0 ha sido usado como el meta-lenguaje en la definición de UP-VSE, AVISPA ha sido seleccionada como herramienta de evaluación dado a que se utiliza como un asistente de validación a través de representaciones visuales de los elementos de proceso y patrones de error, identificando anomalías y proporcionando indicaciones para solucionarlas [38], para más información consultar la sección 2.8.

El análisis de los datos generados por AVISPA se limitó a encontrar nodos o subproyectos desconectados de la red de ejecución principal, cada nodo o subproyecto que se encuentra aislado del flujo principal ha sido anotado en la tabla de errores (ver tabla 4.4) y conectado al proceso.

#### 4.6.1. Resultados del análisis de UP-VSE con AVISPA

A través del tiempo se han mejorado las versiones de UP-VSE gracias a los datos generados por AVISPA, a continuación se ilustra el historial de mejoras presentadas sobre algunas versiones del proceso, en la siguiente sección se ilustra el resultado de la primera y última versión:

##### 4.6.1.1. Vistas obtenidas por AVISPA del proceso UP-VSE

Las vistas generadas por AVISPA para la evaluación del modelo de proceso UP-VSE arrojan una evaluación desde las distintas partes del proceso: evaluación de roles, de artefactos y de tareas.

La evaluación está organizada por versiones las cuales representaron cambios significativos del proceso o han sido versión alfa <sup>1</sup>. La figuras 4.18, 4.19 y 4.20 ilustran

---

<sup>1</sup>“La versión alfa es la primera versión del programa, la cual es enviada a los verificadores para probarla”[55]

las evaluaciones de las tareas, los roles y los artefactos respectivamente de UP-VSE versión 0.5.

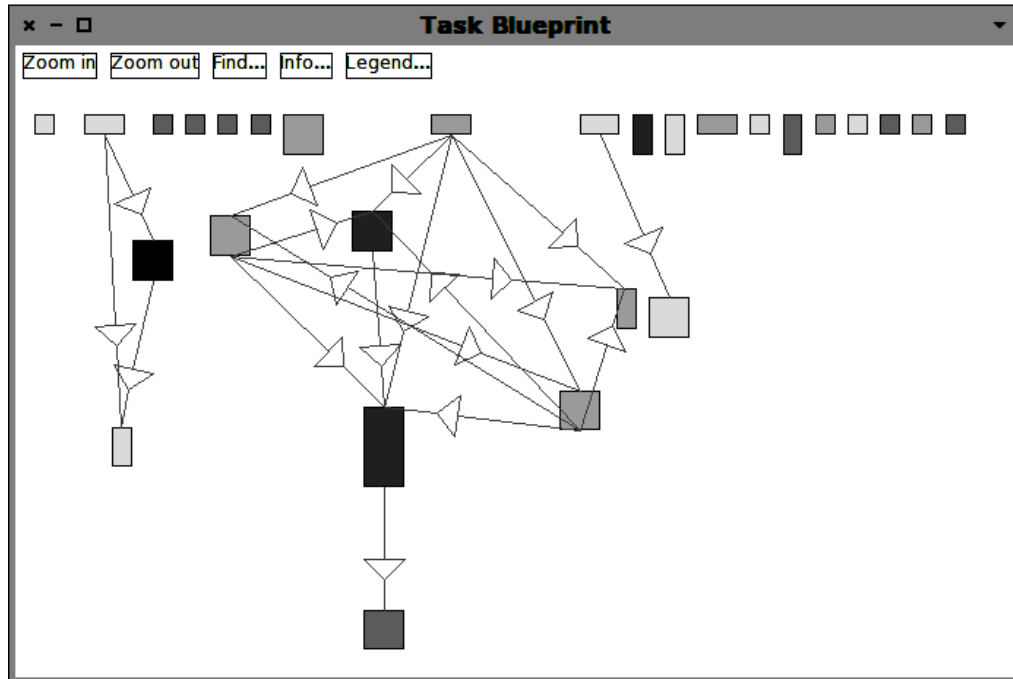


Figura 4.18: Evaluación de las tareas de UP-VSE 0.5 en AVISPA

La figura 4.18 ilustra gráficamente la configuración de tareas obtenidas de UP-VSE v. 0.5 de la cual se puede inferir una gran cantidad de nodos desconectados de la red principal, esto es que se presenta un alto índice de tareas que no aportan valor al proceso.

La figura 4.20 similar al gráfico de tareas arrojado por AVISPA (fig. 4.18) e ilustra la configuración de artefactos de trabajo de UP-VSE v. 0.5. Se infiere de estos resultados que, al observar los nodos desconectados, estos artefactos tampoco aportan valor al proceso.

La figura 4.19 ilustra la configuración gráfica de roles obtenida por AVISPA del proceso UP-VSE v. 0.5. La gran cantidad de nodos interconectados en esta gráfica ilustra un proceso con alta colaboración entre roles.

La elaboración de cada una de las tareas hizo traza de la filosofía iterativa e incremental del Proceso Unificado, en conjunto con artefactos y tareas adicionales con el ISO/IEC 29110-5-1-1:2012 pero reduciendo el peso de la elaboración de artefactos que propone UP en unos pocos pero que aproximan ser los suficientes para lograr obtener un desarrollo genérico para la pequeña industria de desarrollo de software tal como se aprecia en el número de artefactos de trabajo de UP-VSE arrojados por AVISPA.

En efecto, existieron así mismo algunos aspectos de mejora del proceso concluidos por observación de datos arrojados por esta herramienta, especialmente en la

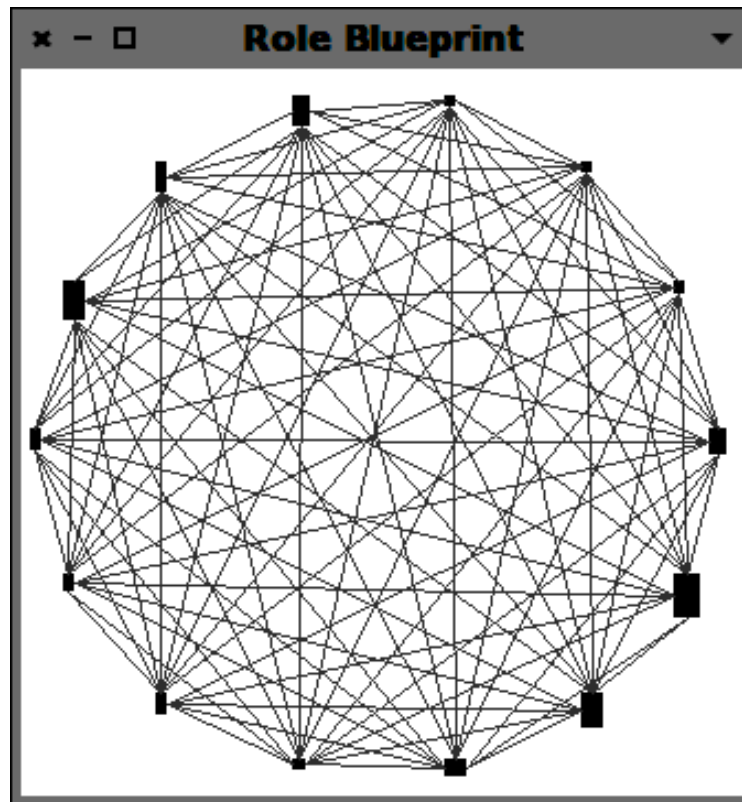


Figura 4.19: Evaluación de los roles de UP-VSE 0.5 en AVISPA

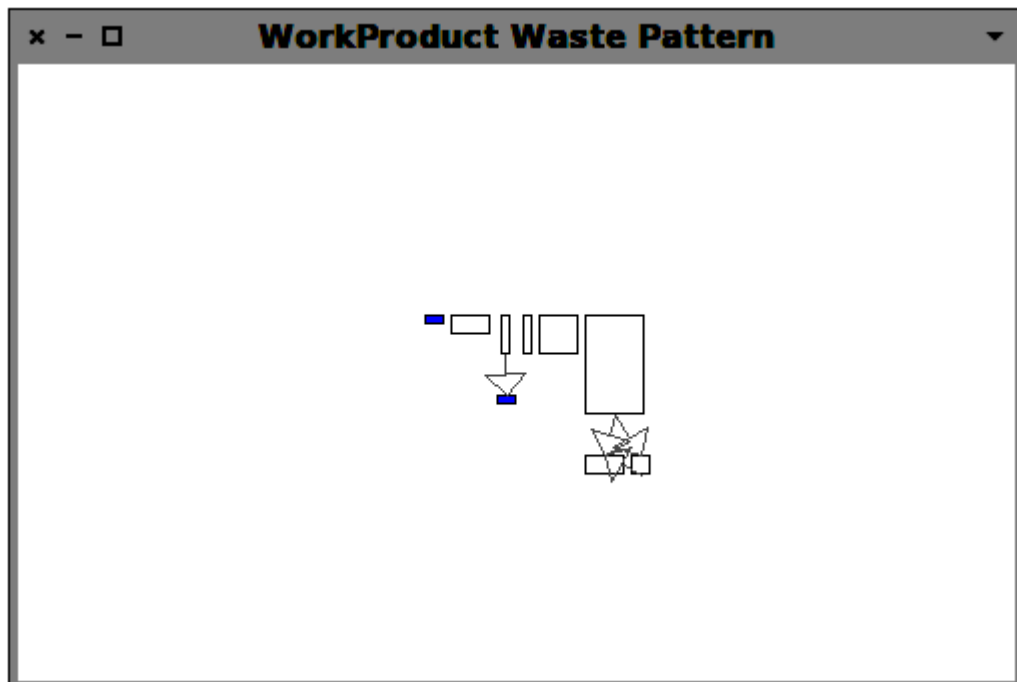


Figura 4.20: Evaluación de los artefactos de UP-VSE 0.5 en AVISPA

vista de roles, ya que proporciona una distribución cuya responsabilidad trata de distribuirse a través de un patrón de roles colaboradores de los cuales pueden hacerse cargo de ciertas tareas cuando los roles principales de su desarrollo no pueden hacerlo. De igual forma también se ha evidenciado que para la cantidad de roles, productos de trabajo y tareas, UP-VSE es un proceso relativamente pequeño abarcando así los aspectos básicos de la norma con elementos de proceso de UP pensando en las necesidades de desarrollo de las VSE.

#### **4.6.2. Acciones realizadas según los errores identificados a través de AVISPA**

Gran parte de los análisis obtenidos de los datos recolectados tras la evaluación de AVISPA (ver sección 4.6.1.1) recaen principalmente en los múltiples nodos sin conexión o serie de subgrafos independientes, las siguientes series de acciones han sido realizadas para obtener un mejor resultado de los nodos y sus interrelaciones, sin embargo pueden persistir errores de especificación que podrían dificultar la comprensión del modelo. Es ideal desarrollar un solo grafo de ejecución del proceso que obtenga múltiples proyectos independientes (grafos desconectados), es decir, que estas partes separadas no aportan al flujo de ejecución del proceso. Este problema suele presentarse en avanzadas fases de su construcción. La tabla 4.4 ilustra las acciones realizadas ante los análisis de los datos arrojados por AVISPA del proceso UP-VSE version 0.5.

Existieron errores encontrados respecto a la herramienta así también como a violaciones sobre las precondiciones de su uso tras su ejecución con UP-VSE, estas consistieron en:

- Bug o error lógico respecto a la estructuración de productos de trabajo específicamente los artefactos "Modelo de casos de uso", contenido en el documento de "casos de uso", que no permitía una salida adecuada de los datos de AVISPA sobre UP-VSE.
- Bug o error lógico sobre la repetida ejecución de lectura de AVISPA con los xml en SPEM 2.0 generados por eclipse arrojando la excepción "SubscriptOutOfBounds: <x>" la variable "x", se ha observado que esta variable corresponde con el número de importaciones del archivo xml del proceso al entorno de trabajo MOOSE y se muestra con  $x \geq 10$ . La figura 4.21 ilustra esta excepción.
- Bug o error lógico sobre la actualización de los planos generados por AVISPA que se presenta cada vez que se realizaba un cambio en el proceso en EPF y al importarlo a MOOSE no se veía reflejado los cambios realizados. La solución temporal hallada consiste en guardar el entorno de trabajo y reiniciar el programa.



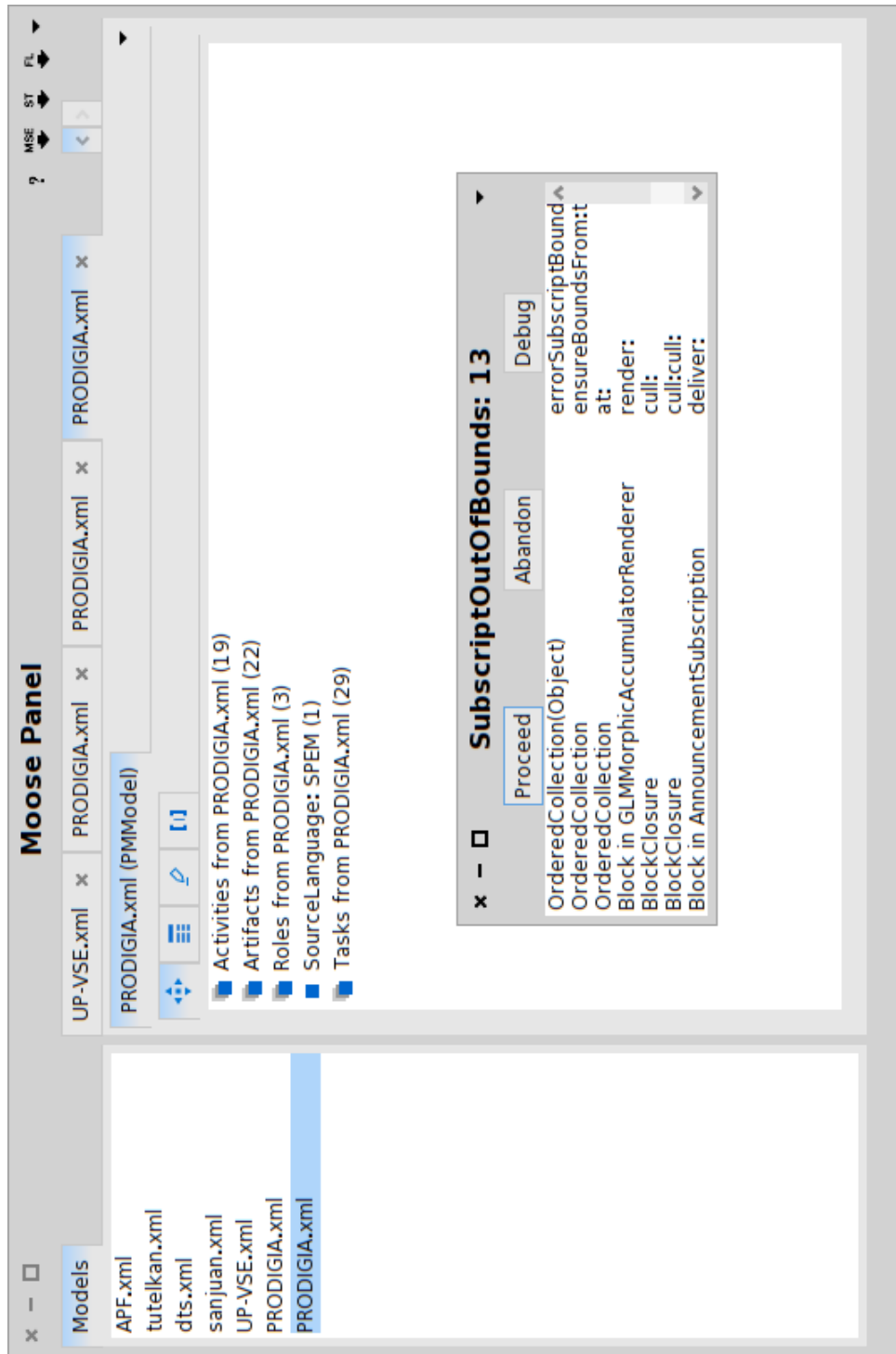


Figura 4.21: Excepción "SubscriptOutOfBounds" de AVISPA

Id Error	Nombre del Elemento o elementos	Patrón de Error	Estado del Error	Descripción del Problema	Solución al Problema
1	Modelo de casos de uso	Subproyectos independientes (Artefactos)	Comprobado	El elemento "Modelo de casos de uso" se encontró desconectado	Se asoció este artefacto como entrada del "Documento de diseño", a través de las modificaciones de 2 tareas en particular: "Esbozar la arquitectura", e "Implementar la solución", donde se añadió este artefacto como entrada y la tarea "Refinar la arquitectura", donde se añadió el mismo elemento mas los "Casos de uso", otorgando como artefacto de salida el "Documento de diseño". Se interconectaron estos artefactos asociando el "Plan de proyecto", como entrada del "Plan de iteración", y se asociaron estos 2 elementos como entradas para el documento de casos de uso colocando estos 2 artefactos como entradas de la tarea "Encontrar y actualizar casos de uso".
2	Plan de proyecto, Plan de iteración	Subproyectos independientes (Artefactos)	Comprobado	Los elementos "Plan de proyecto", y "Plan de iteración", se encontraban desconectados	Existieron 2 tareas que no estaban interconectadas entre si por lo que generaban subproyectos independientes luego de algunas acciones de corrección que no tenían entradas ni salidas ("Integrar componentes desarrollados a una nueva versión de la solución", y "Preparar el entorno para el proyecto")
3	Integrar componentes desarrollados a una nueva versión de la solución, Preparar el entorno para el proyecto	Subproyectos independientes (Múltiples tareas)	Comprobado	Los artefactos nombrados no eran requeridos por ningún otro producto o entregable	Estos artefactos fueron asociados como entradas de la tarea "Empaquetar la versión". Estos artefactos fueron asociados como entradas principales de las tareas "Ejecutar Pruebas de Integración", y "Empaquetar la Versión".
4	Manual técnico, Manual de instalación y Manual de usuario	Sin guía asociada (Artefactos)	Comprobado	El artefacto "Script de pruebas", no era requerido por ningún otro producto de trabajo del proceso	En total se han encontrado 3 errores de tipo subproyectos independientes y 4 errores de tipo sin guía asociada en los artefactos.
5	Script de pruebas	Sin guía asociada (Artefactos)	Comprobado		

En total se han encontrado 3 errores de tipo subproyectos independientes y 4 errores de tipo sin guía asociada en los artefactos.

Tabla 4.4: Acciones correctivas sobre resultados arrojados por AVISPA del proceso UP-VSE

### 4.6.3. UP-VSE con correcciones, visualización con AVISPA

El resultado final del proceso de corrección en AVISPA otorgó los siguientes resultados para los artefactos y tareas de UP-VSE, las figuras 4.22, 4.23 y 4.24 los ilustran. Finalmente los roles no han sido modificados.

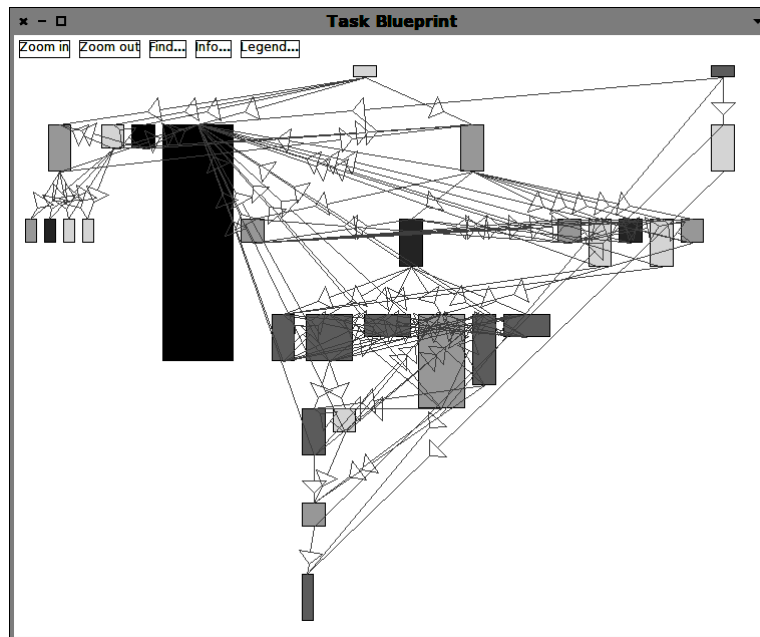


Figura 4.22: Evaluación de las tareas de UP-VSE alpha en AVISPA

Las figuras 4.22 y 4.24 ilustran el resultado de la corrección de UP-VSE v. 0.5 respecto a los nodos independientes en cada una de estas gráficas en las cuales se evidencia una relación de todas las tareas y productos de trabajo respectivamente entrelazados entre si y a sus respectivas redes principales.

Respecto a la figura 4.23 donde se ilustra la gráfica obtenida por AVISPA en la configuración de roles del proceso UP-VSE en donde se observa un aumento de responsabilidades y productos de trabajo asignados por rol, sin embargo se conserva la misma interconexión de roles en UP-VSE v 0.5 en la versión actual de este proceso propuesta.

El siguiente capítulo se ilustrarán aspectos de la evaluación de UP-VSE por medio de un estudio de caso y como se han mejorado estos aspectos de la propuesta de modelo de proceso tomando los resultados de los datos arrojados por la ejecución del estudio de caso con PRODIGIA y la ejecución del modelo de evaluación de la ISO/IEC 29110-3.

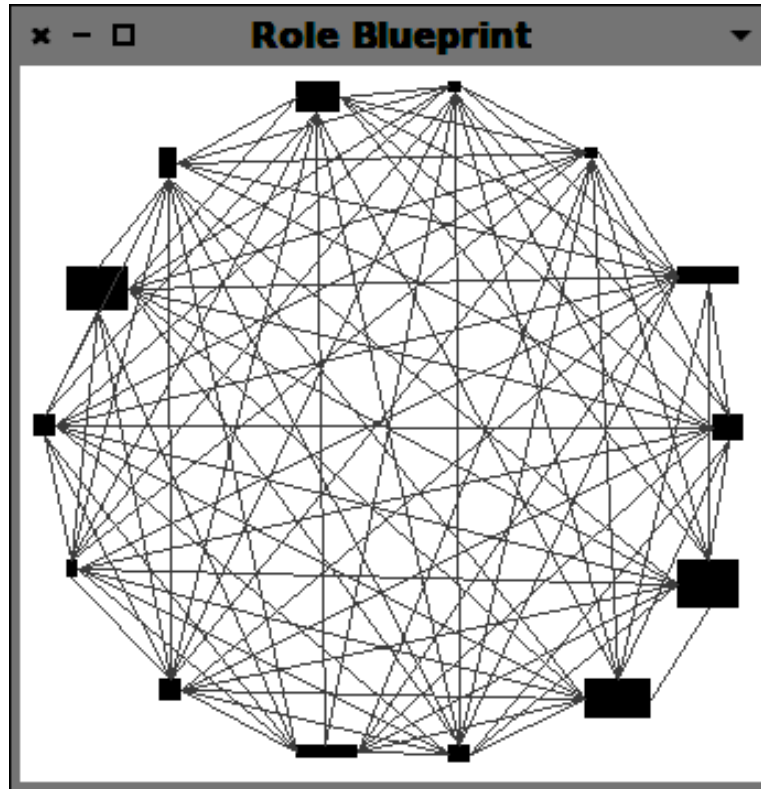


Figura 4.23: Evaluación de las roles de UP-VSE alpha en AVISPA

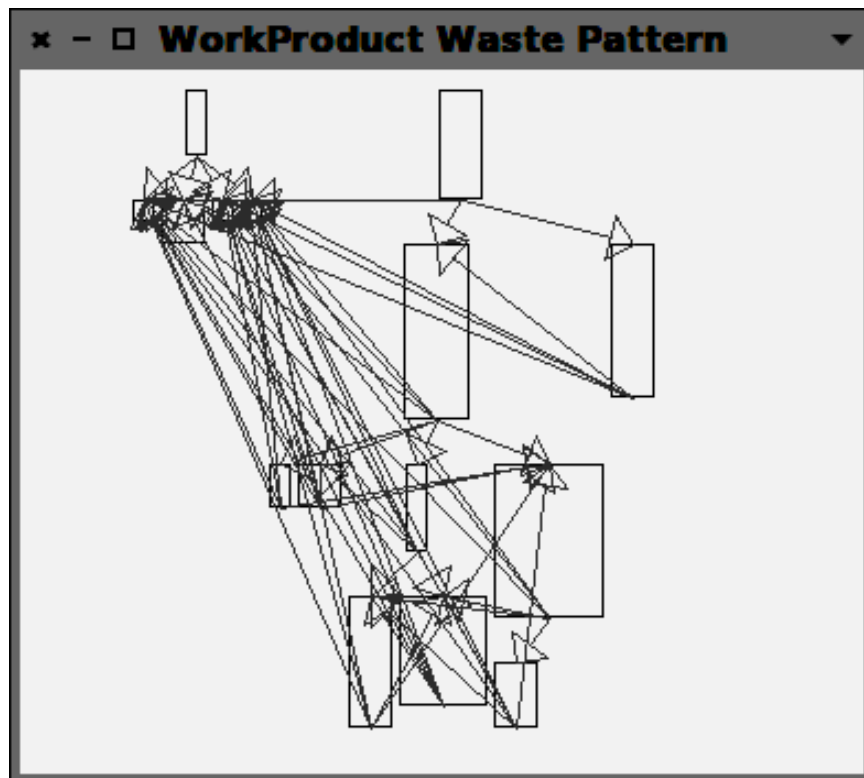


Figura 4.24: Evaluación de los artefactos de UP-VSE alpha en AVISPA

## Capítulo 5

# Evaluación Incremental de UP-VSE

### 5.1. Introducción

La evaluación incremental de fue la estrategia seleccionada en esta tesis para determinar, refinar y evaluar el modelo UP-VSE.

El modelo de UP-VSE ha sido filtrado por una serie de 3 evaluaciones las cuales han permitido construir y refinar el modelo propuesto. Parte de esta evaluación corresponde a la evaluación respecto al modelo ISO/IEC 29110-3, así como la verificación del modelo de proceso a través de AVISPA.

La otra parte de la evaluación ha sido realizada mediante un estudio de caso que se desarrollará a lo largo de este capítulo que se divide principalmente en el estudio de caso PRODIGIA y en las acciones tomadas tras cada resultado obtenido de cada una de las evaluaciones.

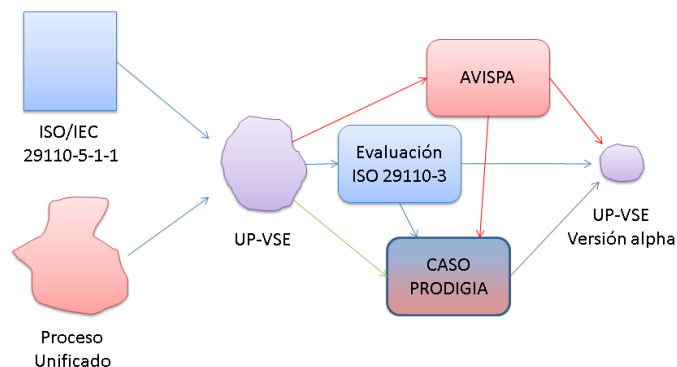


Figura 5.1: Proceso de evaluación de UP-VSE.

## 5.2. Evaluación a través del estudio de caso PRODIGIA

El estudio de caso del proceso UP-VSE gira a entorno a la gestión de un proceso denominado PRODIGIA (PROceso de Desarrollo Iterativo del Grupo de Automática Industrial) [56] el cual pretende solventar las necesidades manifestadas por sus integrantes acerca de la organización de la documentación y de la centralización de la información a través de la web.

El estudio de caso ha sido seleccionado por diversos motivos, principalmente sobre la disponibilidad de recursos entre ellos el entorno universitario en el cual ha sido llevado a cabo, ventajas de disponibilidad de recurso humano y posibilidad de orientación mas personalizada acerca de la ejecución adecuada del proceso y su direccionamiento y especialización a los proyectos de desarrollo de software particulares.

El estudio de caso de PRODIGIA se compone de dos grandes núcleos, el primero consiste en el proceso documentado, formalizado y ejecutado (ver en los anexos el apéndice A). El segundo consiste en el desarrollo de la herramienta de soporte que cumple con las necesidades de gestión de la información mencionadas por el grupo (ver sección 5.3.3.6). El respectivo desarrollo del estudio de caso se encuentra en la sección 5.3.

El proceso de construcción de la propuesta UP-VSE consistió en obtener una configuración lógica inicial de elementos de proceso entre la ISO/IEC 29110-5-1-1 y el Proceso Unificado de I. Jacobson et. al. [2], esta version inicial de UP-VSE ha pasado por tres filtros, el primero fue la evaluación con la norma ISO/IEC 29110-3, el segundo fue con AVISPA y por último se filtró con el estudio de caso PRODIGIA que a su vez fue evaluado con los primeros 2 filtros. Finalmente se obtuvo la versión de entrega de UP-VSE. La figura 5.1 ilustra este proceso. La sección 5.3.3.9 ilustra la evaluación de UP-VSE con respecto a la norma ISO/IEC 29110-5-1-1:2012, e ilustra las ventajas y desventajas del uso de este modelo de proceso.

## 5.3. Desarrollo del caso:

### 5.3.1. Resumen

El reciente desarrollo del estándar de calidad ISO/IEC 29110 y el gran número de entidades desarrolladoras de software que están catalogadas como micro y pequeñas empresas [7] han resaltado la necesidad de definir herramientas que ilustren caminos correctos para la adopción de prácticas que optimicen los recursos destinados al desarrollo de software. Este estudio de caso orientó hacia la definición una instancia de UP-VSE como un Proceso Unificado para proyectos cuyas entidades no exceden mas de 2 personas particularmente con integrantes del grupo de Automática Industrial para lo cual se ha especializado el proceso hacia la construcción de simuladores.

### 5.3.2. Introducción

El presente documento ilustra el diseño, la conducción y reporte de un estudio de caso tipo exploratorio en el marco del proyecto UP-VSE.

El estudio de caso es una metodología de investigación para ingeniería de software. Los casos de estudio son llevados a cabo en el mundo real, en donde se espera que los datos obtenidos en el estudio sean consistentes. La validez de un caso de estudio depende en gran medida del seguimiento del proceso, la ejecución, así como el análisis de los resultados. Este caso de estudio ha sido conducido siguiendo la metodología de estudio de caso propuesto por Runeson et. al. [29].

### 5.3.3. Diseño del estudio de caso

A continuación se describe el diseño del caso, es decir sus objetivos, las preguntas de investigación asociadas, la definición y selección de las unidades de análisis, los indicadores y procesos de medición y los instrumentos utilizados.

#### 5.3.3.1. Objetivo

Determinar el subconjunto de elementos del proceso UP (tareas, roles y productos de trabajo) que son aplicables al contexto VSE a través de una aplicación empírica del UP en una pequeña entidad que no cuenta con un proceso de desarrollo definido.

#### 5.3.3.2. Preguntas de investigación

Este estudio de caso buscó responder a las siguientes dos preguntas de investigación:

- ¿Qué elementos del proceso UP-VSE (tareas, roles y productos de trabajo) son aplicables a una VSE?
- ¿Qué tanto, el conjunto de elementos proceso, de UP-VSE satisface la norma ISO29110-5-1-1?

#### 5.3.3.3. Descripción del caso

La entidad de desarrollo fue un grupo de investigación en el cual como parte de su trabajo, debían desarrollar aplicaciones de software de simulación quirúrgica. El grupo de desarrollo que clasifica como VSE (ver sección 2.5 [34]) estaba conformado por 4 estudiantes, 1 de maestría y 3 de pregrado. El caso se refirió a un proyecto “Diseño y construcción de un ambiente virtual para simulador quirúrgico utilizando interfaces hápticas”. En éste caso el grupo con acompañamiento definió su propio

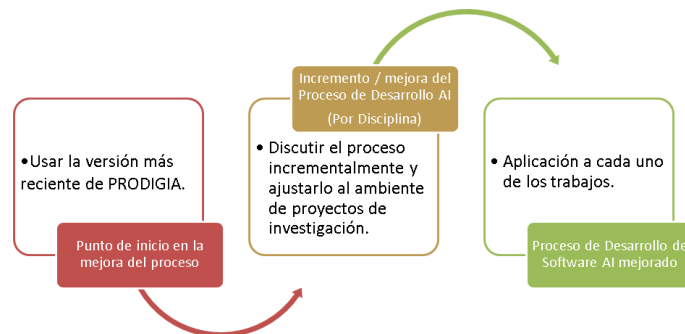


Figura 5.2: Estrategia de trabajo para el estudio de caso.

proceso a partir de la especificación bibliográfica del UP como se muestra en la figura 5.2. El desarrollo del producto se realizó en diferentes plataformas utilizando como base tecnologías como QT y Virtual Matlab VTK a través de Visual Studio, utilizando prototipos de brazos robóticos texturizados a 3D de otros proyectos de investigación llevados a cabo previamente con miras a realizar incisiones internas en los órganos del cuerpo humano sin realizar extensos cortes de la epidermis para acceder a su interior.

#### 5.3.3.4. Selección del caso

La selección de éste caso obedece a dos criterios, el primero es que éste es un caso típico en el que se requiere definir un proceso basado en el Proceso Unificado. El segundo, es la disponibilidad de un escenario real en la misma facultad, lo que facilita la interrelación y aporta un trabajo interdisciplinario valioso para las partes.

La VSE de este estudio de caso, el grupo AI, ha sido seleccionado debido a la disponibilidad de una entidad cuya necesidad principal fuera la mejora de documentación de sus procesos y proyectos ya que notaron que muchos de ellos no podían ser continuados debido a la falta de documentación y de organización del código y por tanto de documentación asociada que ilustrara apropiadamente como ha sido construido el software.

Otro factor que obedece a la selección de esta VSE consistió en el uso de elementos que ya estaban presentes en el modelo de proceso tomado como punto de partida para la construcción de la propuesta de la presente tesis: el Proceso Unificado. La documentación de la VSE de trabajos previos ha ilustrado que hubo un avance por parte de este grupo para estandarizar sus procesos de documentación tal como la construcción de artefactos como diagramas de clase y de casos de uso, por lo que facilita la enseñanza sobre la ejecución de procesos de software, específicamente de PRODIGIA.



UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras de Software Basado en el Proceso Unificado

---

Indicador	Métricas	Fuentes de información	Herramientas
Subconjunto de elementos del proceso con mayor nivel de aceptación y de calidad. (SUP)	NAE - Nivel de aceptación del proceso por parte del equipo de desarrollo. NSE - Nivel de calidad en la aplicación de los productos de trabajo.	Usuarios finales del software, Desarrolladores y gestores del sistema quirúrgico	Entrevistas, encuestas, repositorio del proyecto (Herramienta de gestión documental), y la plantilla de reporte de errores de AVISPA.
Grado de Cumplimiento del proceso UP respecto a la norma ISO 29110-5-1	GA <sub>ISO</sub> - Grado de aplicación de la práctica ISO 29110-5 en el proyecto	Usuarios finales del software, Desarrolladores y gestores del sistema quirúrgico, El producto software	Modelo de Proceso Prodigia. Repositorio del proyecto respecto a la norma ISO 29110-5-1. Protocolo de Evaluación (Mapping)
Adopción del proceso.	NCP - Nivel de cumplimiento del proceso por parte del equipo de desarrollo	Verificación del Proceso	Herramienta de gestión documental. Descripción de los procesos en UP y repositorio del proyecto
Aplicabilidad	E - Esfuerzo	Artefactos de Seguimiento del Grupo de Proceso	Se procesaron las minutas de reunión y se consulta el repositorio del proyecto (Herramienta de gestión documental).

Tabla 5.1: Métricas del estudio de caso

### 5.3.3.5. Hipótesis

#### ***Hipótesis***

Para el caso PRODIGIA, UP-VSE el Proceso Unificado será lo suficientemente útil para las VSE en la implementación del estándar ISO/IEC 29110-5-1-1.

#### ***Hipótesis Nula***

La hipótesis nula planteada ante este método es que: el UP-VSE requiere una transformación considerable para llegar a ser útil a las VSE en la implementación del estándar ISO29110-5, por lo cual será necesario usar complementos de otros modelos de proceso.

### 5.3.3.6. Indicadores e instrumentos de recolección

Para el estudio de caso, se han definido diversos métodos e indicadores sobre los cuales obtener datos para el proceso UP-VSE, principalmente de la herramienta de soporte de documentación de PRODIGIA, del modelo de evaluación realizado a partir de la ISO/IEC 29110-3 y de AVISPA.

#### ***Métricas***

En la tabla 5.1 se han definido un conjunto de indicadores e instrumentos de recolección que se usaran en el caso de estudio.

El estudio se hizo sobre un proyecto de desarrollo de software acerca del control de brazos electrónicos para intervenciones quirúrgicas evitando grandes cortes en la piel para acceder a los órganos, el software ya está hecho, pero carece de patrones y

documentación adecuada que describa cómo se comporta, como puede ser instalado y usado, y como puede ser entendido por otras personas que vayan a trabajar sobre el código.

***Instrumentos:***

Para este estudio de caso también se diseñaron / usaron los siguientes instrumentos:

- Entrevistas y encuestas: que fueron realizadas al equipo de desarrollo del software para procedimientos quirúrgicos con miras a conocer el grado de aceptación, participación y satisfacción del proceso que se ha acordado.
- Artefactos: Utilizando principalmente las monografías de los estudiantes graduados del grupo AI que trabajaron con software, indicaron una forma previa del trabajo del grupo y como podría adaptarse un proceso (PRODIGIA) a sus necesidades.
- El modelo de evaluación ISO/IEC 29110-3: Brinda información sobre la cobertura de la norma sobre el proceso PRODIGIA ilustrando posibles falencias del desarrollo de artefactos, el modelo se encuentra explicado en el capítulo 3 y la evaluación aplicada en la sección 5.3.3.9.
- El análisis de modelos de procesos de software a través de AVISPA: Puede brindar errores del proceso en documentación del modelo y configuración de los elementos del proceso PRODIGIA que han sido reportados en la plantilla de verificación de errores de AVISPA, esta plantilla se encuentra en la carpeta de anexos digitales.
- Herramienta de soporte a la documentación, Anexo C: La herramienta de soporte puede brindar información sobre la forma de trabajo de los estudiantes del grupo AI a través de su monografía. Específicamente la documentación de los estudiantes que realizaron desarrollo de software. Debido a la necesidad de compartir documentación de tesis entre el tesista y el tutor de tesis se diseñó un sistema de gestión documental.
- La definición de un proceso colaborativo cuya herramienta de soporte fue el entorno colaborativo de apoyo a la mejora de procesos de software en pequeñas organizaciones de software COLLAB, Anexo F: De la herramienta de soporte COLLAB se obtuvieron datos sobre el acceso al repositorio interno de documentos y notificaciones de uso de la herramienta y por tanto de parte del esfuerzo invertido en el proyecto.

Para mas información de los instrumentos de recolección ver anexo B.

### **5.3.3.7. Herramientas de soporte para el estudio de caso**

#### ***Diseño e implementación de la herramienta de soporte de PRODIGIA***

El desarrollo de la herramienta de soporte se ha realizado utilizando un sistema gestor de contenidos (CMS) en el lenguaje de programación PHP conocido como Drupal que puede ser descargado de su página oficial<sup>1</sup>, la versión de Drupal de la herramienta de soporte es 7.24 lanzada en la fecha 20-11-2013 cuya arquitectura está documentada en el anexo C y consiste en una herramienta de gestión documental con el objetivo de facilitar la comunicación tutor de tesis - tesista. La instancia de la versión actual de la tesis se encuentra en línea<sup>2</sup>.

Una de las ventajas más significativas de Drupal, consiste en el ahorro significativo de recursos destinados al desarrollo al trabajar con módulos previamente codificados que extienden funcionalidades puede ofrecer esta herramienta. Así para proyectos pequeños con pocos requisitos sin muchas especificaciones, permite pasar directamente de la ingeniería de requisitos a la construcción de software, en donde, los recursos consumidos en modelar planos arquitectónicos se reducen a la investigación de módulos que satisfagan las necesidades del desarrollador evitando codificar las funcionalidades ya implementadas.

En la construcción de ésta herramienta se ha seguido un proceso de ingeniería de requisitos derivado del Proceso Unificado usando artefactos como los casos de uso y su respectivo modelo en conjunto con estrategias de validación y verificación tal como reuniones periódicas con el objetivo de dirigir correctamente el desarrollo de la herramienta.

La figura 5.3 ilustra una captura de pantalla de la herramienta de soporte.



Figura 5.3: Herramienta de gestión documental de PRODIGIA

### ***COLLAB y el uso de Thinklets para el soporte de la elaboración del proceso PRODIGIA***

<sup>1</sup><http://www.drupal.org/>

<sup>2</sup><http://jhonalvarez.hol.es/PRODIGIA>

## UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras de Software Basado en el Proceso Unificado

Para tratar de mitigar la falta de tiempo por parte de los integrantes del equipo de procesos y mejorar el uso del tiempo en la organización de ideas para la definición del proceso, se implementó el uso de un proceso colaborativo basado en “Thinklets”<sup>1</sup> y de una herramienta diseñada por el grupo IDIS denominada COLLAB [57], diseñada con el propósito de soportar colaborativamente la creación de procesos de software en pequeña y mediana empresa. La instancia de COLLAB [57] utilizada para el desarrollo inicial de PRODIGIA se encuentra en línea<sup>2</sup>.

La herramienta de soporte COLLAB entro rápidamente en desuso, debido a que no tenía ningún método de comunicación automática de cambios ni había colaboración de parte de los demás usuarios del sistema para estar pendientes de la herramienta ante alguna novedad y por ende, el proceso colaborativo también dejó de implementarse completamente.

El anexo F contiene una información detallada de la elaboración del proceso colaborativo y de la ejecución en el estudio de caso, este artículo aún está en proceso de elaboración.

La figura 5.4 ilustra una captura de pantalla de la herramienta.



Figura 5.4: Instancia de COLLAB para ser utilizado en PRODIGIA

### 5.3.3.8. Ejecución del caso

Este estudio de caso, fue conducido en colaboración con integrantes del grupo de investigación AI (Grupo de Automática Industrial) con el objeto de resolver dificultades y brindar una mejor gestión de la documentación de los proyectos al interior

<sup>1</sup> **Thinklet:** Es la unidad más pequeña de capital intelectual requerido para crear un patrón de colaboración simple, repetible y predecible a través de personas que trabajan por un objetivo.

<sup>2</sup> <http://jhon-jairo-alvarez-londono.alwaysdata.net/collab/>

de su grupo de investigación. Este estudio de caso se ha dividido en dos partes, la primera consistió en la definición del proceso donde se llevaron a cabo reuniones para entender sus dinámicas y observar patrones de desarrollo que puedan servir como "buenas" prácticas, la segunda parte consistió en la ejecución del proceso con estudiantes de pregrado del mismo grupo de investigación.

**El proceso PRODIGIA**

El proceso PRODIGIA es un modelo de proceso específicamente adaptado a las necesidades del desarrollo de software y de documentación de tesis de grado basado en Proceso Unificado de I. Jacobson et. al. [2].

El ciclo de vida del proceso PRODIGIA, al igual que el Proceso Unificado está dividido en fases y disciplinas. Las fases del proceso PRODIGIA (En este orden: Formulación, Especificación, Programación y Entrega) muestran una vista del tiempo transcurrido del proceso conformada por iteraciones cuyas actividades (Análisis, Diseño, Programación, Pruebas, Documentación y Planificación), conocidas en UP como disciplinas, se ejecutan en paralelo. La figura 5.5 ilustra la distribución de esfuerzos del proceso y contrasta disciplinas y fases.

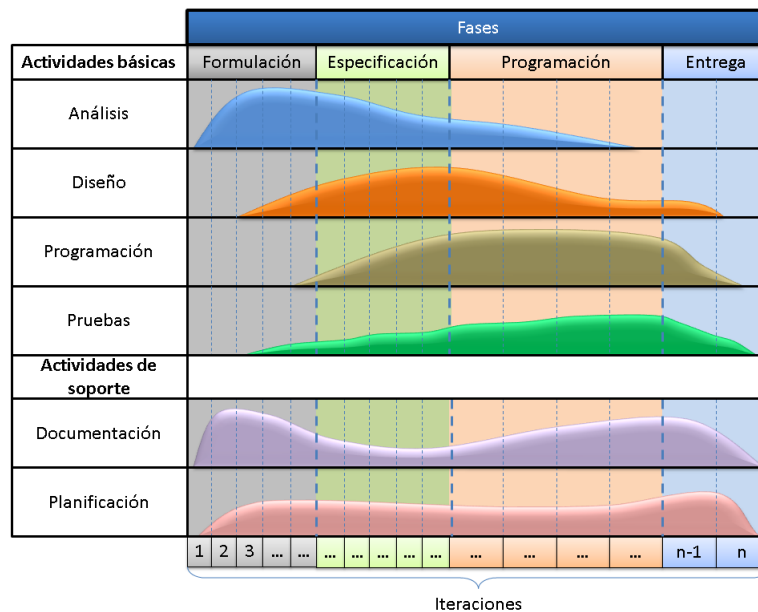


Figura 5.5: Ciclo de vida de PRODIGIA

El anexo A contiene una documentación mas detallada e ilustra la configuración de los elementos del proceso.

**Reutilización de artefactos de UP-VSE en PRODIGIA**

Para el diseño del proceso PRODIGIA se han definido ciertos artefactos que pueden servir de insumos tanto para este proceso como para la propuesta UP-VSE, La tabla 5.2 ilustra un porcentaje aproximado de artefactos que han sido reutilizados en el modelo de proceso empleado en esta propuesta.

**Reuniones iniciales y dinámica para la creación el proceso PRODIGIA**

UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras de Software Basado en el Proceso Unificado

Artefactos		Porcentaje de reutilización
UP-VSE	PRODIGIA	
Casos de uso	Documento de análisis de requisitos	100
	Casos de uso priorizados	
Modelo de casos de uso	Diagrama de casos de uso	100
Documento de diseño	Diagrama de clases	50
	Diagrama de comunicación entre objetos	
	Modelo de diseño del sistema	
Manual de instalación	Manual de instalación Manual de usuario	66.6
Manual de usuario		
Manual técnico		
Software	Código software incompleto	90
Componente software	Código software completo	
Caso de prueba Script de prueba	Artefacto a probar	66.6
	Artefacto corregido	
	Artefacto de prueba	
Plan de proyecto	Formato A y la visión del producto	15
Configuración de software	Ninguno	0
Plan de despliegue		
Plan de iteración		
Prototipos de interfaz de usuario		

Para ponderar el porcentaje que determinó la utilidad total de los artefactos reutilizados se multiplicó cada porcentaje por el número de artefactos de UP-VSE que calificaron en dicho porcentaje y se divide por el número de artefactos de UP-VSE, finalmente la ponderación arrojó como resultado que un **90.4%** de los artefactos de UP-VSE fueron reutilizados en PRODIGIA.

Tabla 5.2: Reutilización de artefactos de UP-VSE en PRODIGIA

Las primeras aproximaciones para la definición del proceso PRODIGIA fueron hechas usando el Proceso Unificado de I. Jacobson et. al. [2] iniciando con el proceso de captura de requisitos. Cada subproceso fue tomado y adaptado a las necesidades del grupo AI.

La dinámica de trabajo consistía en explicar como funciona la disciplina de UP ilustrando el flujo de ejecución del proceso, luego, los integrantes de procesos lo analizan para definir su propio proceso, se resolvieron dudas y se recibieron sugerencias y se llegó a un consenso de la creación del proceso PRODIGIA por actividades principales, al final fue documentada la versión del proceso convirtiendo las actividades principales en una configuración de elementos de proceso que ilustraran el ciclo de vida completo del proceso.

**Evaluaciones realizadas en el estudio de caso:**

Durante el caso se realizaron varias evaluaciones las cuales fueron:

- Con respecto a la norma ISO/IEC 29110-3: conducida por el Dr. Julio Ariel Hurtado quien supervisaba el trabajo realizado por el estudiante de pregrado Jhon Jairo Álvarez quien realizaba la evaluación del proceso
- Con respecto a AVISPA: al igual que la anterior evaluación también ha sido dirigida por el Dr. Julio Ariel Hurtado orientando el trabajo del estudiante de pregrado Jhon Jairo Álvarez quien realizó el análisis de los resultados obtenidos

de esta herramienta.

- Con respecto a los participantes del estudio de caso: que son integrantes del grupo AI adscrito a la Universidad del Cauca quienes desarrollaron software en base al proceso cuyo director de tesis, el Dr. Oscar Andrés Vivas Albán colaboró liderando a los estudiantes de maestría Ing. Luis Daladier Guerrero quien colaboró en la definición del proceso PRODIGIA e Ing. Karin Correa Arana que ejecuto este proceso donde ambos participaron en el llenado de sus respectivas encuestas de definición y ejecución de PRODIGIA respectivamente. Al igual que las anteriores evaluaciones, también ha sido liderada por el Dr. Julio Ariel Hurtado y ejecutada por el estudiante de pregrado Jhon Jairo Álvarez.

### ***Proceso seguido para la definición de PRODIGIA***

Una serie de reuniones fueron definidas para la creación, definición y documentación de PRODIGIA:

1. **Reuniones de gestión de procesos:** Estas reuniones fueron realizadas con tesis de maestría y tutores de tesis de grado interesados en la definición del proceso de desarrollo. En paralelo, fueron definidas una serie de reuniones de las cuales se encuentra un beneficio común en el desarrollo de un sistema gestor que ayude a organizar mejor el proceso. Reuniones posteriores se concentraron en definir una base inicial del proceso que respondiera a las necesidades más urgentes de formalización de la documentación de los requisitos de software para su posterior construcción sobre el resto de disciplinas. Para más información de los aspectos más importantes de las reuniones, el anexo D ilustra los puntos acordados en dichas reuniones.
2. **Reuniones con tutores de tesis de la línea de investigación de Ingeniería Automática del grupo IA:** Luego de la fase de formalización inicial del proceso, se dieron unas charlas a los tutores de tesis de estudiantes del grupo IA en las cuales se ilustraron ventajas del uso del proceso y como puede realizar una mejor gestión de la documentación utilizando un software subido en la plataforma (ver Anexo C), como las reuniones han sido concretadas para llevar un control sobre los tesis, se han mejorado aspectos de gestión del proceso a pesar de que el proceso se enfoca más en la construcción del software. La figura D.2 (en anexos) ilustra como ha sido llevada a cabo esta fase.
3. **Trabajo de formalización del proceso PRODIGIA, desarrollo de la herramienta de soporte documental y reuniones para validación, verificación y transición del software:** En el anexo D se ilustran algunas de las reuniones que han sido llevadas a cabo para la corrección tanto del modelo de proceso como de la herramienta de gestión de la documentación bajo los requisitos del grupo IA acordando especificaciones de la herramienta en los servidores de la Universidad del Cauca.

### ***Proceso seguido para la ejecución de PRODIGIA***



Figura 5.6: Reuniones para la socialización de proceso y de la herramienta de soporte al grupo AI.

Cada una de las reuniones ha contribuido a una socialización de la forma de trabajo y de las prácticas realizadas para el mejoramiento tanto del proceso como de las prácticas de programación y documentación llevadas a cabo. Como parte del proceso de validación y verificación de este modelo de proceso, se ha realizado lo siguiente:

1. **Reuniones de socialización del proceso:** Aquí se impartieron conceptos para la ejecución inicial del proceso, se validó también su acogida ante los tesisistas y su flexibilidad ante casos específicos de los proyectos. Se capturaron e interpretaron los datos de desarrollo del estudio de caso y se tomaron las respectivas acciones correctivas y preventivas.
2. **Utilización de la herramienta de gestión y asesoramiento continuo sobre el proceso:** Una vez realizada la gestión de la comunicación del conocimiento sobre PRODIGIA, los estudiantes de pregrado y los tutores de tesis, acompañados por el asesoramiento continuo sobre el proceso pueden comenzar a desarrollar sus respectivos proyectos de grado. Las reuniones de socialización tanto de la herramienta como del proceso han venido siendo constantes, la figura 5.6 ilustra la última reunión de socialización del proceso y de la herramienta de soporte.
3. **Reuniones de socialización de avances con expertos en el área:** Cada avance fue socializado con expertos en sus respectivas áreas de procesos y desarrollo de simuladores, luego contrastado con el proceso para hallar oportunidades de mejora basándose en prácticas de construcción de software mejores a las que se llevaban a cabo y sin cambiar dramáticamente el proceso para evitar re-ingeniería de los elementos realizados en la ejecución del mismo.
4. **Captura de datos y mejora del proceso:** A través de encuestas realizadas a cada rol (Su diseño se encuentra en el Anexo B y los resultados en la sección 5.3.4) se obtuvieron variables importantes en la continuación de la elaboración de PRODIGIA. A lo que repercutió en cambios sobre la elaboración y configura-



UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras de Software Basado en el Proceso Unificado

Disciplina	Numeración en PRODIGIA	Nombre de la tarea
Análisis	1.1.1.	Reunión con el director de tesis.
	1.1.2.	Realización de objetivos y planteamiento del problema.
	1.1.3.	Realización del cronograma, presupuesto y acta de propiedad.
	2.1.1.	Realizar el formato A y la visión del producto.
	2.2.2.1.	Elaborar actores y casos de uso.
	2.2.2.2.	Priorizar casos de uso.
	2.2.2.3.	Detallar casos de uso.
	2.2.2.4.	Corregir y verificar casos de uso.
Diseño	2.2.1.1.	Construcción del modelado matemático.
	2.2.3.1.	Escoger los casos de uso a diseñar.
	2.2.3.2.	Mejorar casos de uso a elegir.
	2.2.3.3.	Realizar, corregir o mejorar el modelo de diseño del sistema.
	2.2.3.4.1.	Realizar, corregir o mejorar el diagrama de clases.
Programación	2.2.3.4.2.	Realizar, corregir o mejorar el diagrama de comunicación entre objetos.
	2.2.4.1.	Entender los diagramas construidos.
	2.2.4.2.	Realizar correcciones a los diagramas de software.
	2.2.4.3.	Construir el software basado en los modelos creados.
Pruebas	4.2.1.	Liberar el código software.
	2.2.6.1.	Diseñar prueba.
	2.2.6.2.	Ejecutar prueba.
	2.2.6.3.	Corregir prueba.
	2.2.6.4.	Corregir defectos en el artefacto.
Documentación	2.2.6.5.	Documentar prueba.
	2.2.5.1.	Incluir la documentación realizada en la monografía.
	2.2.5.2.	Realizar o corregir el artículo.
	2.2.5.3.	Documentar el manual de instalación.
	2.2.5.4.	Documentar el manual de usuario.
	4.2.2.	Liberar el artículo.
4.2.3.	Liberar la monografía.	

Tabla 5.3: Numeración de tareas de PRODIGIA

ción del proceso y sobre la documentación asociada, los posteriores capítulos del presente trabajo ilustran detalladamente en que consistieron.

5. **Ejecución de la versión actualizada:** Se concretó con los tesisistas que ejecutan el proceso un trabajo de transición del mismo basándose en reuniones cortas de capacitación y en ejemplos previos.

**5.3.3.9. Evaluación de PRODIGIA respecto a la ISO/IEC 29110-3 del perfil ISO/IEC 29110-5-1-1:2012 y usando AVISPA**

Usando los elementos de evaluación de procesos en VSE de la sección 3.2.1, se procede a realizar la evaluación de PRODIGIA (ver sección 5.3.4.1).

Al igual que la evaluación de UP-VSE con AVISPA, se utiliza la misma metodología de evaluación empleada para evaluar el proceso PRODIGIA (ver sección 4.6 para más información). El análisis de los datos generados por AVISPA se limitó a encontrar nodos o subproyectos desconectados de la red de ejecución principal, cada nodo o subproyecto que se encuentra aislado del flujo principal ha sido anotado en la tabla de errores (ver tabla 5.9) y conectado al proceso.

### 5.3.4. Resultados

#### 5.3.4.1. Ejecución y resultados del proceso de evaluación respecto a la ISO/IEC 29110-3 sobre PRODIGIA:

##### ***Los flujos de trabajo de PRODIGIA***

A diferencia de UP, las actividades en PRODIGIA no mantienen un orden de ejecución al interior de las disciplinas, sin embargo lo hacen a través del ciclo de vida. Ver tabla 5.3. Esta tabla ilustra una numeración asignada a cada tarea del proceso PRODIGIA cuyo objetivo consiste en asignar un identificador único con fines de evaluación y documentación.

#### 5.3.4.2. Presentación y análisis de los resultados respecto al modelo de evaluación de la ISO/IEC 29110-3

##### ***Evaluación de las tareas de PRODIGIA:***

Al igual que la evaluación del Proceso Unificado, en PRODIGIA se han seguido los parámetros de evaluación establecidos en 3.2.1. Los resultados se ilustran en la tabla 5.4. Esta tabla contrasta el mapeo realizado entre la ISO/IEC 29110-5-1-1:2012 y la ISO/IEC 12207 en la tabla 3.1 (primeras dos columnas de izquierda a derecha) y cada una de las tareas del proceso PRODIGIA (primeras cinco filas de arriba hacia abajo) en la evaluación propuesta por la ISO/IEC 29110-3 utilizando los parámetros del modelo de evaluación propuesto en el capítulo 3, esto significa que cada tarea del mapeo (en relación a la norma ISO/IEC 29110-5-1-1) ha sido validada con cada tarea del proceso PRODIGIA obteniendo los resultados de dicha tabla.

Similarmente a la evaluación de UP, se procede a hallar el  $Q_T$  utilizando las formulas de la sección 3.2.1.2:

$$Q_T[Tareas] = \frac{\sum Q_T[C_T\{T_i\}]}{|T|}$$

$$Q_T[Tareas] = \frac{(1*14)+(0.3*4)}{22}$$

$$Q_T[Tareas] = 0.6909$$

$$C_T(Q_T[Tareas]) = L = \text{Ampliamente alcanzado.}$$



UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras de Software Basado en el Proceso Unificado

Artefactos		$C_W$	$Q_W$
ISO/IEC 29110-5-1-1	PRODIGIA [56]		
Plan del proyecto	Anteproyecto de grado [56]	P	0.3
Documento de especificación de requisitos	Monografía [56]	T	1
Identificación de componentes software	Monografía [56]	T	1
Casos y procedimientos de prueba	Monografía [56]	P	0.3
Reporte de pruebas	Monografía [56]	P	0.3

Tabla 5.5: Evaluación de los artefactos de PRODIGIA respecto al estándar ISO/IEC 29110-5-1-1:2012

ISO/IEC 29110-5-1-1:2012		PRODIGIA [56]
Customer	CUS	Stakeholder del estudio de caso
Project Manager	PM	Tutor de tesis
Work Team	WT	Tesista

Tabla 5.6: Equivalencia de roles entre ISO/IEC 29110-5-1-1 y PRODIGIA

**Evaluación de los artefactos de PRODIGIA:**

Los artefactos que han sido creados y acordados con los participantes del estudio de caso sobre los cuales se han extraído una serie de plantillas junto con el proceso documentado disponible en [56]. La tabla 5.5 muestra la calificación otorgada a los distintos artefactos de PRODIGIA. Cada plantilla ha sido evaluada respecto a la descripción hecha en la ISO/IEC 29110-5-1-1, esto es:

$$Q_T[\text{Productos\_de\_trabajo}] = \frac{\sum Q_T[C_T\{T_i\}]}{|T|}$$

$$Q_T[\text{Productos\_de\_trabajo}] = \frac{(1*2)+(0.3*3)}{5}$$

$$Q_T[\text{Productos\_de\_trabajo}] = 0.58$$

$$C_T(Q_T[\text{Productos\_de\_trabajo}]) = L = \text{Ampliamente alcanzado.}$$

**5.3.4.3. Consideraciones adicionales respecto al modelo de evaluación**

**Equivalencia de roles:**

Al igual que en la evaluación del Proceso Unificado, PRODIGIA también considera una equivalencia de roles a los mencionados en la ISO / IEC 29110-5-1-1. La tabla 5.6 lo ilustra.

#### 5.3.4.4. Acerca de la ejecución del proceso de evaluación en PRODIGIA

Al igual que en la evaluación de UP y UP-VSE (ver sección 3.2 y 4.5). También es posible esperar un cierto nivel de subjetividad al ejecutar este modelo de evaluación con PRODIGIA debido a que esta evaluación fue realizada por los autores de esta tesis por lo que podría afectar la confiabilidad de los datos obtenidos. Este proceso de evaluación tuvo como esfuerzo aproximado un total de 119 horas / hombre cuya duración fue de 2 meses aproximadamente.

En el estudio de caso se vuelve persistente el uso de malas prácticas para el desarrollo de software atribuido a varios posibles factores:

- **Primero:** obtenido a través de las preguntas de las encuestas consiste en que el proceso PRODIGIA no es fácil de interiorizar y ejecutar inclusive con asesoramiento de la ejecución de artefactos.
- **Segundo:** por medio de charlas y frecuentes asesorías, se resaltó la necesidad de satisfacer objetivos secundarios centrando el esfuerzo del desarrollo de software del proyecto PRODIGIA en base a la herramienta (software) de soporte y no al proceso.
- **Tercero:** Existieron constantes retrasos de los estudiantes para entregar artefactos del proceso debido a que el tiempo que ellos tienen para entregar sus artefactos es mayor que el tiempo de desarrollo del proyecto. Se asume que este riesgo se materializó debido a que el proyecto no tenía mayor control sobre los estudiantes y la falta de apoyo del tutor de tesis por incentivar el uso del proceso mas que de la herramienta de soporte.
- **Cuarto:** Otro inconveniente que se ha presentado en todos los proyectos del estudio de caso es el comienzo de la ejecución del proceso sin iniciar el proyecto desde su inicio, para lo que se acudió al análisis de los artefactos ya realizados para ver si cumplían o no con PRODIGIA. Es evidente que de este análisis se infiere que pudieron haber surgido cambios significativos en el tiempo de ejecución del proceso que no se pudieron observar al momento de su realización en un punto mas avanzado de dichos proyectos.

#### 5.3.4.5. Vistas obtenidas por AVISPA del proceso PRODIGIA

AVISPA ha generado una serie de vistas del modelo de proceso PRODIGIA ilustrando una serie de falencias de sus primeras versiones.

En el modelo de negocio fue necesario una entrega temprana del proceso al grupo, por lo que en esa versión no se arroja vista de roles ya que ha sido inherente que todas las tareas han debido ser ejecutadas por el tesista.

Las figuras 5.7, 5.8 y 5.9 ilustran los resultados de artefactos, tareas y roles respectivamente obtenidos por AVISPA.

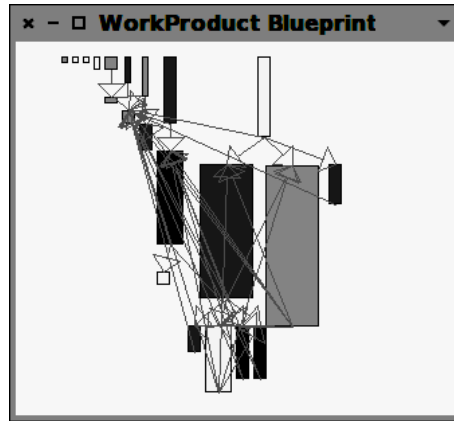


Figura 5.7: Configuración de artefactos de PRODIGIA v. 0.3 arrojada por AVISPA.

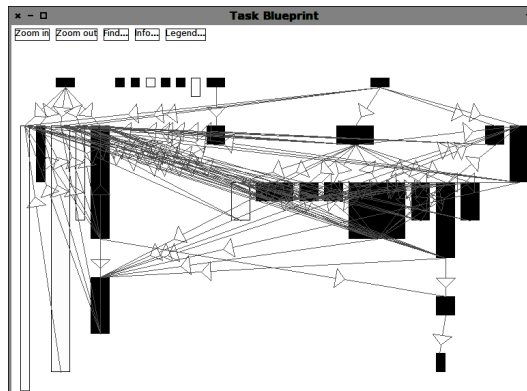


Figura 5.8: Configuración de tareas de PRODIGIA v. 0.3 arrojada por AVISPA.

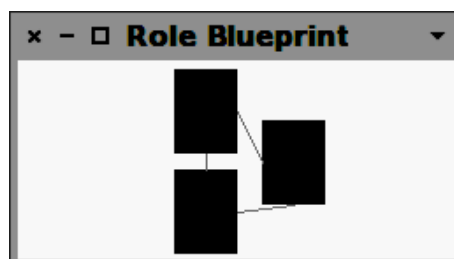


Figura 5.9: Configuración de roles de PRODIGIA v. Alpha arrojada por AVISPA.

Los siguientes errores detectados tras el respectivo análisis de las figuras anteriores consisten en:

- Nodos no conectados con el proyecto tanto en la vista de tareas como en la de artefactos.
- Algunas tareas contienen esfuerzo no relativo respecto a los roles, principalmente el esfuerzo mantiene una proporción mayor al número de roles asignados para la realización de la tarea.

La siguiente sección ilustra las acciones llevadas a cabo respecto a los principales errores encontrados en el PRODIGIA.

Los resultados de cada encuesta (para consultar su diseño ver anexo B) han sido capturados e interpretados para obtener cada una de las métricas de la sección 5.3.3.6:

**Resultados de la encuesta general para tesis:** La tabla 5.7 ilustra el resultado de esta encuesta. Esta tabla contiene una serie de preguntas divididas en 2 partes. El primer grupo de preguntas refieren a la completitud de cada tarea del proceso PRODIGIA respecto al proyecto realizado, cada tarea esta identificada con el sufijo numérico propio dentro del proceso. Las preguntas desde E1 hasta E5 son el segundo grupo de preguntas, que contienen su respectiva descripción en el pie de la tabla al igual que la respuesta R1 a la pregunta E5. Cada pregunta de la encuesta tuvo un rango de calificación numérica de 0 a 5 a excepción de la pregunta E5 que consistió en un campo abierto sobre la opinión tanto de la elaboración del proceso como de la encuesta.

**Resultados de la encuesta para los tutores de tesis y equipo de procesos:** La tabla 5.8 ilustra el resultado de esta encuesta.

Al igual que en la encuesta anterior cada pregunta de la encuesta tuvo un rango de calificación numérica de 0 a 5 a excepción de la pregunta 4 en la encuesta para tutores y la 5 en la encuesta para el equipo de definición de procesos que consistieron en campos abiertos sobre la opinión tanto de la elaboración del proceso como de la encuesta.





### **5.3.5. Análisis de los resultados**

#### **5.3.5.1. Respecto al estudio de caso**

Tras la realización del estudio de caso se han determinado las siguientes conclusiones:

- PRODIGIA es un proceso que refleja una gran complejidad tanto en su interiorización como en su ejecución en base a las encuestas obtenidas sobre la definición y puesta en marcha del proceso.
- El modelo de proceso al parecer debe contar con un asesor especialmente a las personas que ejecutan el proceso en sus fases iniciales debido a que constantemente deben ser informados sobre como se puede llevar a cabo y como se ve aplicado a su proyecto en particular.
- La herramienta de soporte ha favorecido a la comunicación entre el tutor de tesis y el tesista, pero basándose en las necesidades de los integrantes del grupo de investigación, lo cual ha desviado el diseño de la herramienta hacia el proceso que es el objetivo que se pretendía alcanzar con su desarrollo.

#### **5.3.5.2. Respecto a las encuestas**

Se realizaron un total de 3 encuestas para 4 participantes, de los cuales 3 las respondieron, las 3 encuestas de satisfacción del proceso destinadas para el equipo de procesos, el tesista que ejecutó el proceso y el tutor de tesis respectivamente. En cada encuesta se respondían preguntas de forma cuantitativa donde el usuario asigna un valor de calificación de 0 a 5. De esta forma se obtuvo que el integrante del equipo de procesos, Esp. Luis Daladier Guerrero calificó el proceso con un promedio de 3.25, el tesista lo valoró con un promedio de 3.75 y el tutor de tesis consideró un promedio de 4.67. Si se promedian estos valores se obtiene una calificación del proceso en 3.89.

A partir de la encuesta para tesistas que, al requerir otros materiales distintos del flujo de ejecución del proceso que ilustra el modelo SPEM 2.0 revela que quien lo ejecutó no está familiarizado con la interpretación de de este tipo de modelos ni con la navegación propuesta por la página web generada del proceso en EPF.

Para los integrantes que participaron en el proceso PRODIGIA la utilidad que consideraron del proceso fue del 77.8% a partir de la calificación total del proceso respecto a las encuestas, lo que podría significar una amplia utilidad y reutilización de los elementos del proceso PRODIGIA en las VSE.

### 5.3.5.3. Respecto al modelo de evaluación de la ISO/IEC 29110-3

Los resultados indican que norma ISO/IEC 29110-5-1-1:2012 alcanzó un 69.09% de cobertura del proceso PRODIGIA, lo que en la escala definida en la ISO/IEC 15504 indica que el proceso se encuentra ampliamente alcanzado de un total de 22 tareas obtenidas tras el mapeo de la norma ISO/IEC 29110-5-1-1:2012 con la ISO 12207 (ver tabla 3.1) de las cuales 14 se encuentran totalmente alcanzadas y 4 parcialmente alcanzadas.

De estos datos se concluye que la ISO/IEC 29110-5-1-1:2012 no alcanza el proceso tan ampliamente como lo realiza el Proceso Unificado de I. Jacobson et. al. [2], ni siquiera tanto como la propuesta UP-VSE, pero esto es debido a que el proceso PRODIGIA se encuentra construido específicamente a trabajos cuyos integrantes son de 1 o 2 tesisistas máximo quienes realizaron todos los artefactos de su tesis.

También se concluye que la propuesta PRODIGIA no fue totalmente alcanzada por la norma ISO/IEC 29110-5-1-1:2012 debido a que los proyectos contemplados por este proceso no abarcan únicamente el desarrollo de software. En efecto, el proceso también debe gestionar la documentación propia del ámbito académico en el cual se ejecutó y sus ambientes controlados los cuales son:

- Requisitos parcial o ampliamente claros desde el comienzo de la ejecución del proyecto, por lo que el esfuerzo invertido a la ingeniería de requisitos se centra más en documentación que en la captura.
- Tiempos controlados, ya que generalmente los proyectos software por lo general pueden tener un cambio de cronograma lo que repercute en ampliación o recorte del tiempo destinado para el desarrollo del producto
- Interés en mostrar resultados rápidos, ya que el proceso, al estar desarrollado por pocos integrantes no priorizan la necesidad de manejar buenas prácticas como crear un repositorio amplio de control de versiones o diseñar planos arquitectónicos de software. Se observó que centran sus esfuerzos en la necesidad de construcción de código funcional por lo que algunas tareas fueron omitidas y/o modificadas.

A pesar de las diferencias entre un ambiente académico y uno industrial, los resultados obtenidos indican que el proceso PRODIGIA conserva un 69.09%<sup>1</sup> de aplicabilidad para las VSE según el modelo de evaluación de la ISO 29110-3.

---

<sup>1</sup>Este dato se obtuvo a partir de la fórmula de evaluación de tareas de la tabla 5.4 en la subsección 5.3.4.2

#### **5.3.5.4. Respetto a los datos obtenidos por AVISPA**

Al igual que la evaluación de UP-VSE, se han ordenado en una serie de acciones y patrones de error las falencias halladas tras el análisis del proceso PRODIGIA.

La tabla 5.9 ilustra las acciones realizadas ante los errores detectados en los datos arrojados por AVISPA del proceso PRODIGIA version 0.3.

Así mismo también existieron errores encontrados respecto a la herramienta así también como a violaciones sobre las precondiciones de su uso tras su ejecución con PRODIGIA, estas consistió en un bug o error lógico respecto al nombrado de tareas específicamente la tarea "Documentar el manual de usuario", que carecía de un modelo adecuado para ser reconocido por la herramienta. Su solución consistió en cambiar el nombre al estándar manejado internamente (Upper Camel Case).

La tabla de errores encontrados por AVISPA (tabla 5.9) ilustra que de 30 tareas 5 fueron detectadas como subproyectos independientes y de 21 artefactos 3 fueron observados bajo este mismo patrón de error. Según este análisis se concluye que AVISPA tuvo una efectividad del 16.6% sobre las tareas y del 14.2857% sobre los artefactos de PRODIGIA. Los roles no han causado variabilidad alguna en este análisis, dado que su configuración se ha mantenido igual tanto antes como después de esta evaluación.

Estos datos reflejan una mejoría de la efectividad del proceso PRODIGIA en un 15.6862% lo que implica un mayor margen de eficiencia y reutilización de los elementos de este proceso en una VSE.

#### **5.3.5.5. PRODIGIA con correcciones en base a AVISPA**

Las figuras 5.10 y 5.11 muestran el antes y el después de artefactos y tareas respectivamente. Los roles no han sido modificados (ver figura 5.12).

Id Error	Nombre del Elemento o elementos	Patrón de Error	Estado del Error	Descripción del Problema	Solución al Problema
1	"Corregir defectos en el artefacto"	Subproyectos independientes (Tareas)	Comprobado	La tarea "Corregir defectos en el artefacto" no tiene artefactos de entrada ni de salida	Asociar como entradas de esta tarea los artefactos "Artefacto a probar", y "Artefacto de prueba checklist", y de salida el artefacto "Artefacto corregido".
2	"Corregir prueba"	Subproyectos independientes (Tareas)	Comprobado	La tarea "Corregir prueba", no tiene artefactos de entrada ni de salida	Asociar como entradas de esta tarea los artefactos "Artefacto a probar", y "Artefacto de prueba checklist" de salida el artefacto "Artefacto de prueba checklist", corregido.
3	"Documentar prueba"	Subproyectos independientes (Tareas)	Comprobado	La tarea "Documentar prueba", no tiene artefactos de entrada ni de salida	Asociar como entradas de esta tarea los artefactos "Artefacto a probar", "Artefacto corregido", y "Artefacto de prueba checklist", y de salida el artefacto "Monografía (incompleta)", con los artefactos de entrada debidamente corregidos.
4	"Ejecutar prueba"	Subproyectos independientes (Tareas)	Comprobado	La tarea "Ejecutar prueba", no tiene artefactos de entrada ni de salida	Asociar como entradas de esta tarea los artefactos "Artefacto a probar", y "Artefacto de prueba checklist", y de salida el artefacto "Artefacto de prueba checklist", con cada prueba ejecutada y documentada.
5	"Liberar el artículo"	Subproyectos independientes (Tareas)	Comprobado	La tarea "Liberar el artículo", tiene artefactos de entrada / salida sin una ubicación adecuada	Cambiar el artefacto "Artículo completo", de entrada opcional a salida e incluir el artefacto "Artículo incompleto", en la realización de la monografía y el artículo.
6	"Anteproyecto de grado"	Subproyectos independientes (Artefactos)	Comprobado	El artefacto "Anteproyecto de grado", no está conectado con el proyecto	Incluir el artefacto "Anteproyecto de grado completo", como entrada de la tarea "Incluir la documentación realizada en la monografía".
7	"Código software completo"	Subproyectos independientes (Artefactos)	Comprobado	El artefacto "Código software completo", no está conectado con el proyecto	Cambiar el artefacto "Código software completo", de entrada opcional a salida en la tarea "Liberar código software".
8	"Monografía completa"	Subproyectos independientes (Artefactos)	Comprobado	El artefacto "Monografía completa", no está conectado con el proyecto	Cambiar el artefacto "Monografía completa", de entrada opcional a salida y añadir "Anteproyecto de grado completo", como entrada en la tarea "Liberar la monografía".

En total se han encontrado 8 errores de tipo de subproyectos independientes, 5 sobre tareas y 3 sobre artefactos.

Tabla 5.9: Acciones correctivas sobre resultados arrojados por AVISPA del proceso UP-VSE

### 5.3.6. Evaluación cualitativa de la ejecución del estudio de caso

En el desarrollo del estudio de caso se han podido realizar varias observaciones que determinaron algunas malas prácticas en su ejecución:

- **Poco acceso a la documentación del proceso:** En base a la respuesta de la encuesta para tesisistas obtenida, la documentación del proceso PRODIGIA ha sido poco consultado desde su libre acceso a los servidores del Alma Mater por lo que el desarrollo de ciertos artefactos frecuentemente les causaba dificultad y en consecuencia optaban por asesoría del proceso. Sin embargo, en dicha documentación se encuentran ejemplos que ilustran el desarrollo de los diversos artefactos que exige el proceso.
- **Confusión en el objetivo del proceso:** Como consecuencia del ítem anterior, en reuniones de supervisión del trabajo realizado por los estudiantes del grupo AI que decidieron optar por una asesoría para mejoramiento de su trabajo se centraron en el desarrollo de su solución y no en el mejoramiento de la forma de trabajo que estaban realizando. La finalidad del proceso PRODIGIA consiste en realizar una mejora de las prácticas que los estudiantes estaban llevando a cabo y no de definir una solución pronta para realización del código de sus respectivos proyectos software.
- **Poca permanencia y disponibilidad de parte de los tesisistas:** Dado a la diferencia de las fechas límite entre los trabajos de los tesisistas del grupo AI y la presente tesis no se ha visto reflejado una mejora significativa en el crecimiento de artefactos a través del tiempo. Esto afectó el desarrollo de los proyectos de dichos estudiantes debido a que los tesisistas mantenían en otras actividades diferentes al desarrollo de sus tesis.
- **Confusión en el concepto de la herramienta de gestión documental (Anexo C):** El tutor de tesis del grupo AI centro su atención principalmente en el desarrollo de la herramienta de gestión documental asociando el término PRODIGIA a esta. Principalmente cuando se requería cumplimiento en el tiempo de desarrollo del proceso se refería al desarrollo de artefactos más que a la utilización de la herramienta de gestión documental.

En base a estas observaciones, en este estudio de caso no se ha logrado obtener ventaja de la amplia referencia del UP en el ámbito académico para el desarrollo de software. Sin embargo los estudiantes si tuvieron conocimientos previos en el uso de herramientas CASE<sup>1</sup> que soportaban UML en conjunto con los artefactos que requiere el Proceso Unificado.

A pesar del desconocimiento del UP y de la forma adecuada de trabajo para desarrollo de software los integrantes del Grupo AI han mencionado un alto índice de satisfacción con el proceso PRODIGIA y en especial con el desarrollo de la herramienta de soporte documental.

---

<sup>1</sup>**Herramientas CASE (Ingeniería de Software Asistida por Computador):** Son aplicaciones informáticas que soportan el desarrollo de artefactos de diseño del ciclo de vida de desarrollo de software

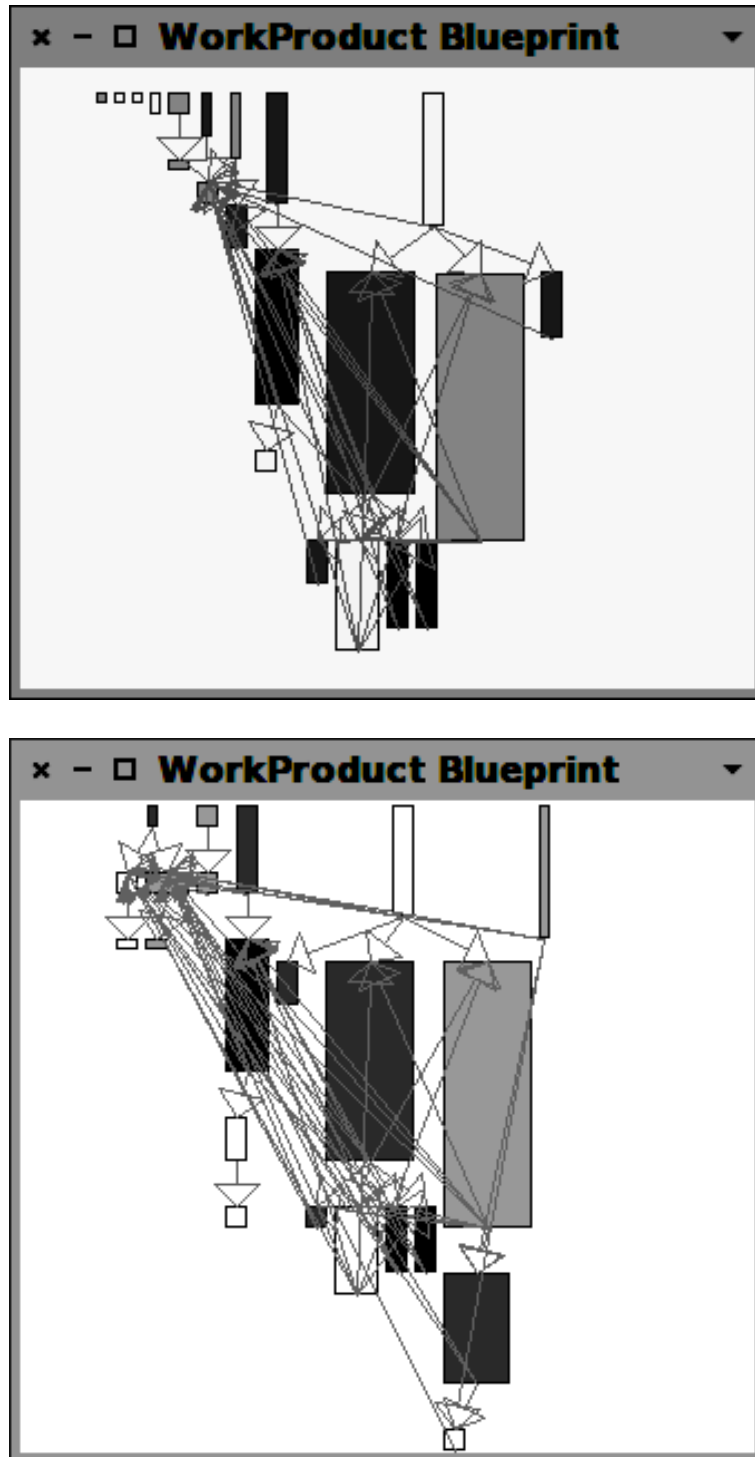


Figura 5.10: Configuración de artefactos de PRODIGIA v. 0.3 arrojada por AVISPA antes (arriba) y después (abajo) de las correcciones

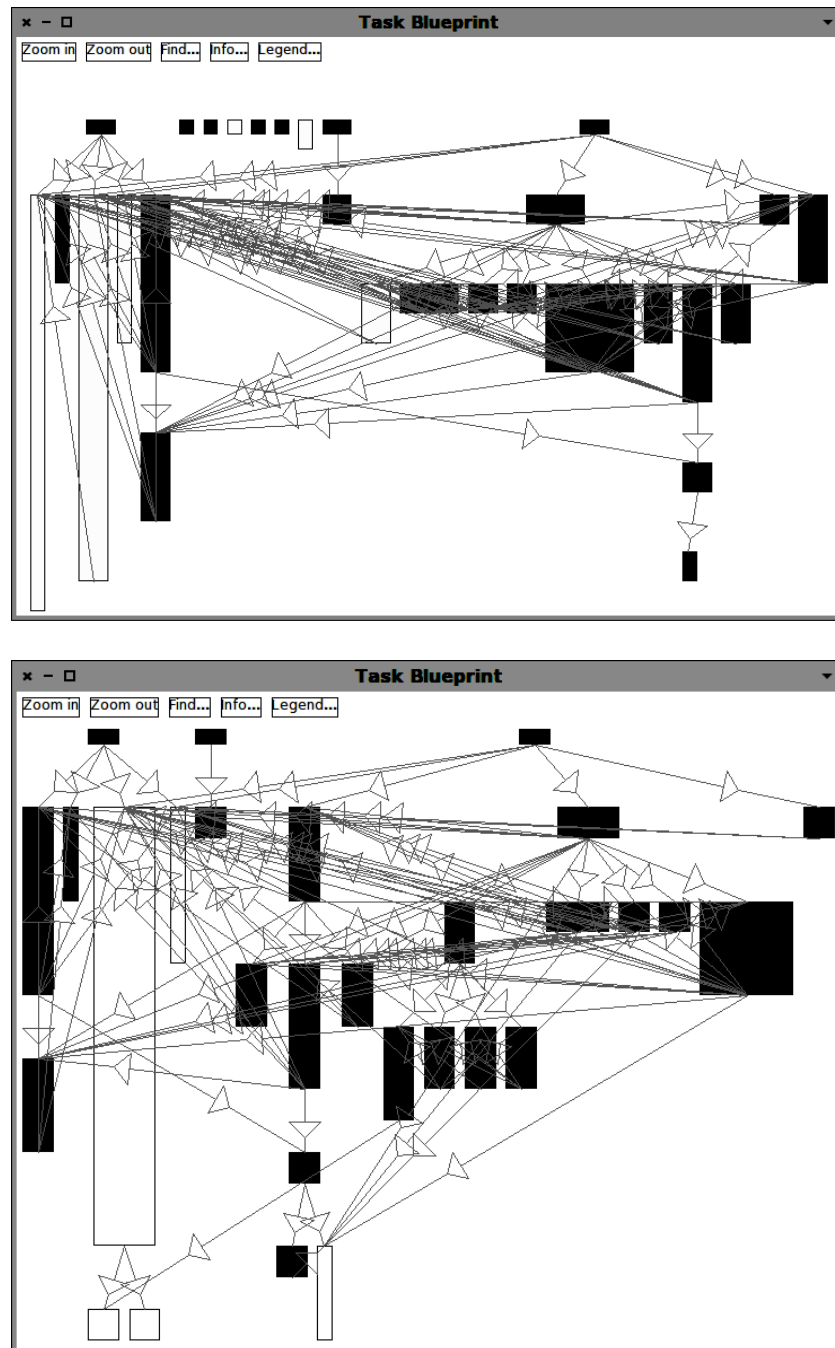


Figura 5.11: Configuración de tareas de PRODIGIA v. 0.3 arrojada por AVISPAantes (arriba) y después (abajo) de las correcciones

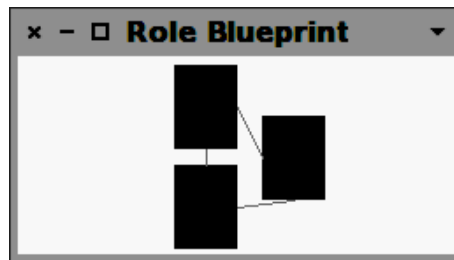


Figura 5.12: Configuración de roles de PRODIGIA v. Alpha arrojada por AVISPA.

## 5.4. Correcciones sobre UP-VSE

Como se ha ilustrado en los anteriores capítulos se han llevado a cabo 3 evaluaciones las cuales han arrojado una serie de valores para el análisis y eliminación de errores del proceso UP-VSE. A continuación se expondrán cada una de las acciones realizadas para los datos obtenidos en cada evaluación:

- **Asesoramiento continuo con énfasis en la construcción de software:** Se ha añadido el rol de asesor del proceso para que acompañe y guíe en la ejecución del proceso, específicamente en aspectos de desarrollo de software según lo evidenciado en las encuestas del estudio de caso redistribuyendo esfuerzos con el objetivo de lograr una ejecución del proceso mas rápida.
- **Disminución de tareas:** A comparación con el proceso PRODIGIA, de las encuestas obtenidas se pudo obtener una evidente complejidad en la comprensión del proceso por lo que se ha disminuido una gran cantidad de tareas que estaban presentes tanto en la ISO/IEC 29110-5-1-1:2012 como en el UP.
- **Corrección de la configuración de los elementos de proceso:** Gracias al uso de AVISPA, el proceso cuenta con una mejor organización de sus elementos.
- **Mejoramiento de la documentación:** Se han realizado descripciones mas detalladas de la documentación que el proceso UP-VSE requiere en base a las encuestas obtenidas y sus observaciones en cuanto a la comprensión del proceso PRODIGIA.
- **Mayor cantidad de ejemplos:** Se han incluido ejemplos sobre como documentar y desarrollar los principales artefacto que contiene la propuesta UP-VSE.
- **Inclusión de artefactos:** Los artefactos de documentación de los manuales del sistema y el script de pruebas han sido agregados como salidas producidas por UP-VSE que estaban también presentes en el proceso PRODIGIA.



## Capítulo 6

# Conclusiones, limitaciones, trabajo futuro y aportes adicionales

Esta tesis presenta una propuesta de un marco metodológico basado en el Proceso Unificado [2] para el desarrollo de artefactos de software para pequeñas entidades donde los métodos son bien conocidos por la industria y ampliamente aceptados por la comunidad académica.

Para el desarrollo de esta propuesta, se ha adaptado el trabajo de I. Jacobson et. al. [2] en conjunto con el perfil básico ISO/IEC 29110-5-1-1:2012 a través de 3 procesos de evaluación ejecutados en paralelo. Estos son el uso del modelo de evaluación propuesto en ISO/IEC 29110-3, la herramienta de análisis visual para procesos en SPEM 2.0 AVISPA, y el estudio de caso PRODIGIA que a su vez también está evaluado con los mismos parámetros elaborados a partir del modelo de evaluación ISO/IEC 29110-3, los cuales han ido refinando los elementos de proceso de UP-VSE y han representado un cambio significativo del modelo a partir de sus primeras versiones.

El trabajo de construcción de la herramienta de soporte sobre la gestión documental de PRODIGIA en conjunto con la utilización de la herramienta COLLAB han permitido una mejor realización y ejecución del proceso, a pesar de las negativas de su uso y de sus observaciones y oportunidades de mejora halladas de las herramientas de soporte han permitido definir estrategias de trabajo que vayan mas acorde al estilo y recursos involucrados para la definición y ejecución del proceso PRODIGIA.

## 6.1. Conclusiones

La evaluación de UP-VSE con PRODIGIA permitió ver una serie de falencias en cuanto a artefactos de entrega y cumplimiento de la norma ISO/IEC 29110-5-1-1 lo que repercutó en actualizar los elementos de proceso y en la disminución en la complejidad cognoscitiva que requería su interiorización y adopción al interior de cada uno de los proyectos. Específicamente, la ejecución de PRODIGIA en una VSE a través de un estudio de caso permitió observar una serie de mejoras a la propuesta UP-VSE:

- Mejora en la documentación disminuyendo el número de tareas tratando de no sacrificar la calidad del producto.
- Disminución de tareas dispersas que, en colaboración con el análisis obtenido por AVISPA, esta falencia ha logrado ser detectada y mitigada.
- Se detectó la necesidad del manejo de un control de versiones que mantuviera cambios actualizados y funcionales del producto software que está incluido en la propuesta UP-VSE y con miras a abarcar parte de la norma ISO/IEC 29110-5-1-1.
- En el estudio de caso se mencionaron diversas observaciones sobre la instalación y configuración de aplicaciones para el desarrollo de software debido a que el diseño de simuladores requirió de varias tecnologías que tenían que ser instaladas y configuradas, por lo que fueron añadidas tareas livianas enfocadas en proveer un entorno de desarrollo de software.

En el estudio de caso, el Proceso Unificado de I. Jacobson et. al [2] permitió iniciar el desarrollo del estudio de caso tomando y adaptando el subproceso de captura de requisitos a conveniencia del Grupo AI, convirtiéndose en la piedra angular de la construcción del proceso PRODIGIA.

UP-VSE es una herramienta que aporta en la construcción de artefactos de ingeniería de software reutilizable y adaptable a las necesidades de las VSE basado en el estándar ISO/IEC 29110-5-1-1:2012 y en el ciclo de vida del Proceso Unificado de I. Jacobson et. al. [2].

La evaluación de UP-VSE respecto a AVISPA ha permitido encontrar una serie de falencias (que se han tratado de mitigar) como tareas y productos de trabajo sin vinculación entre sí y fortalezas halladas accidentalmente como roles altamente colaborativos del proceso. El análisis con AVISPA del proceso UP-VSE permitió mejorar la distribución de carga de trabajo y unir las tareas o subprocesos y artefactos que no pertenecían a la red principal. También permitió observar una configuración de roles en donde se ilustra una alta colaboración entre sí.

La evaluación de UP-VSE respecto al modelo de evaluación ISO/IEC 29110-3 arroja resultados positivos respecto al cumplimiento de elementos de proceso especialmente tareas y artefactos respecto a las ISO/IEC 29110-5-1-1 a excepción de algunos aspectos de la gestión de la configuración que exige la norma. Específica-

mente la evaluación respecto a la ISO/IEC 29110-5-1-1 permitió realizar una serie de mejoras a la propuesta UP-VSE:

- Refinamiento del proceso UP-VSE orientando el Proceso Unificado a las características de las VSE [13].
- Adaptación de las tareas de UP-VSE hacia el cumplimiento de los objetivos propuestos en la ISO/IEC 29110-5-1-1, lo que facilitó su evaluación respecto a esta norma.

La cantidad de esfuerzo involucrado en la primera evaluación (UP, ver sección 3.2) ha conllevado un mayor esfuerzo que las otras 2 (UP-VSE, ver sección 4.5 y PRODIGIA 5.3.3.9) debido al esfuerzo implementado en la formalización inicial del modelo de evaluación instanciado en base a la norma ISO/IEC 29110-5-1-1.

La herramienta de gestión documental de PRODIGIA fue ampliamente utilizada por los estudiantes del grupo AI, hallando una posible solución como forma de ilustrar la documentación del proceso. La interacción con el software que facilite comunicación entre tutores de tesis y tesisistas ha demostrado ser más útil que las demás herramientas usadas para el desarrollo y documentación de procesos de software.

En el transcurso del tiempo de la tesis se alcanzaron otros logros aparte de los objetivos propuestos en el anteproyecto, principalmente fueron:

- Construir un proceso colaborativo para la creación de procesos en una VSE basado en Thinklets (Anexo F).
- Desarrollar un sistema de gestión documental para el grupo AI pensado para la comunicación tutor de tesis - tesisista que, a futuro, puede estar pensado no solamente para ser usado al interior de este grupo sino que podría beneficiar a toda la Universidad (Anexo C).
- Utilizar la herramienta COLLAB [57] para soportar la comunicación y documentación en la definición del proceso PRODIGIA.

Estas metas alcanzadas lograron mejorar ciertos aspectos logísticos de comunicación y de organización al trabajo.

En el transcurso del desarrollo del estudio de caso se vuelven claros los grandes cambios entre metodologías de desarrollo de software, esto es, que es muy poco probable utilizar puramente una metodología de desarrollo de software para proyectos de software que poseen una naturaleza demasiado cambiante [45] y se evidenció en gran medida en proyectos ejecutados por VSE como lo es PRODIGIA, debido a que el desarrollo en el estudio de caso se hace máximo con 3 personas incluyendo el tutor de la tesis y la especialización del proceso UP-VSE hacia el cumplimiento de objetivos particulares como académicos y propios del desarrollo de una simulación software. Como se muestra en el estudio de caso, la estructura de los elementos de proceso de PRODIGIA y UP-VSE son distintas, tanto en relación con UP, la ISO/IEC 29110-5-1-1 y el entorno académico ya que estas variables influyeron para construir un proceso bastante específico al tipo de proyecto que se requería realizar en el estudio de caso.

## 6.2. Limitaciones

Una empresa de desarrollo de software puede brindar un mayor espectro del trabajo que se debe hacer en el proceso por lo tanto sería interesante probar la propuesta UP-VSE en una VSE de desarrollo.

La ejecución del proceso UP-VSE-COLLAB no tuvo el éxito esperado, la herramienta de soporte COLLAB no generó la dinámica de colaboración esperada. En efecto, la comunicación cara a cara sin el uso del proceso colaborativo fue la opción mas escogida.

Debido a que la propuesta UP-VSE está basado en un perfil básico, no ha sido posible expandir la misma hacia el cumplimiento de perfiles intermedios o avanzados debido a la escases de recursos y a la reciente aparición de la norma ISO/IEC 29110. La ISO se encuentra en proceso de elaboración de dichos perfiles.

El desarrollo del estudio de caso también muestra que los integrantes del Grupo AI pertenecen mas a los procesos hardware e industriales que a los software, por lo que no refuerzan sus conocimientos en las buenas prácticas de desarrollo sino en codificación y en el desarrollo de aplicaciones de forma artesanal, debido a lo expresado en las encuestas realizadas al equipo de desarrollo y a su especialización hacia otras ramas de la ingeniería como en este caso es la ingeniería automática industrial.

## 6.3. Trabajos futuros

Debido a que UP-VSE es una aproximación de los procesos técnicos para el desarrollo de software, es relevante ampliar esta propuesta a los subprocesos de gestión de proyectos contemplados en el perfil ISO/IEC 29110-5-1-1:2012.

La propuesta elaborada en esta tesis UP-VSE debería ser aplicada a nuevos casos de desarrollo como los cursos de Proyecto I y II del programa de ingeniería de sistemas o en un caso ideal como una pequeña empresa para visualizar un mayor espectro su aplicabilidad en donde se espera una mejor serie de resultados.

Se debe realizar una nueva evaluación del proceso realizando un énfasis en su estructura y realizar mejoras con el fin de simplificar el proceso y mejorarlo. Como por ejemplo transformar tareas a pasos, simplificar la ruta de ejecución de una rama del proceso que puede presentar el mismo resultado o mejor con menos tareas y pasos o mejorar las plantillas que menciona el proceso UP-VSE, así se pretende mostrar una configuración con un contenido mas concreto de las tareas y como consecuencia una gestión mas adecuada del esfuerzo. Lo anterior en el marco de futuros estudios de caso

Para futuros trabajos de definición de procesos a través de un estudio de caso en el que se den las condiciones para un entorno colaborativo se podría utilizar, modificar o reformular el proceso colaborativo ilustrado en el anexo F donde su construcción de mantenga lo suficientemente estable como para brindar un modelo mas adecuado para la elaboración de procesos de software utilizando y/o acoplando la herramienta COLLAB preferiblemente o cualquier otra que se adapte a las necesidades de la VSE.

En base al trabajo de V. Ribaud et al. [44] se debe tomar en cuenta que el trabajo realizado por estos autores contemplan desde el nivel de capacidad 1 hasta el 3 de la ISO/IEC 15504, y que a futuro, la ISO desarrollará perfiles intermedios y avanzados de la sección ISO/IEC 29110-5, y por ende, de la ISO/IEC 29110-3 que ilustrará cuales atributos de proceso deberán ser cumplidos para cada uno de estos perfiles. Se propone entonces construir diversas versiones de UP-VSE que traten de cumplir con los perfiles intermedios y avanzados dado a que las pequeñas organizaciones necesitarán una certificación de calidad a la medida de su experiencia y competitividad, ya que la ISO/IEC 29110-5-1-1 corresponde a un perfil básico. Una posible estrategia podría incluir la realización de un análisis de la version actual de UP-VSE a través del modelo de evaluación propuesto por la ISO/IEC 29110-3 para encontrar el nivel de cumplimiento de un perfil intermedio o avanzado y desde ese punto de partida utilizar elementos de proceso UP o RUP hasta alcanzar un cumplimiento aceptable del perfil.

La propuesta UP-VSE puede ser modificada para incluir un proceso de capacitación de personal que no se encuentra familiarizado con este proceso para tratar de mejorar sus prácticas de construcción de software aplicada en un estudio de caso.

Es necesario realizar una evaluación mas minuciosa de cada uno de los artefactos de la propuesta con el objetivo de mejorar traducción y campos de cada una de las plantillas de UP-VSE.

## 6.4. Aportes adicionales

Este trabajo ha presentado aportes que no estaban contemplados en los objetivos de la presente tesis:

El primer aporte consistió en la elaboración de una herramienta documental cuyas funcionalidades no solamente favorecieron al grupo AI participante del estudio de caso de la presente tesis, sino que también pueden funcionar para lo programas de la Universidad del Cauca que necesiten una labor documental entre tesis y tutores de tesis (ver anexo C).

El segundo aporte fue la elaboración de un artículo que ha sido aceptado para el Noveno Congreso Colombiano de Computación que ilustra como fue el trabajo inicial de la tesis respecto a la instanciación del proceso de captura de requisitos de la metodología de desarrollo UP de I. Jacobson et. al. [2] y que ilustra los inicios del proceso PRODIGIA y el trabajo con el equipo de procesos (ver Anexo E).

El tercer aporte fue la elaboración de un proceso colaborativo denominado UP-VSE-COLLAB el cual contribuyo al trabajo colaborativo en el estudio de caso del grupo AI instanciando la herramienta COLLAB elaborada en el framework CakePHP, arreglando errores de inicio de sesión, específicamente arreglando el código de encriptación md5 en la comparación de la contraseña de la base de datos con la que el usuario digitó (ver Anexo F).

El cuarto aporte fue diseñar un modelo de evaluación reutilizable basado en la ISO/IEC 29110-3 para cualquier proceso orientado a VSE (ver capítulo 3).

El quinto aporte se consideró debido a que el alcance de UP-VSE ha sido mayor al planeado pues la propuesta consideraba solo aspectos de ingeniería. Sin embargo, esta propuesta recurre a otros subprocesos de gestión que se encargan de tareas como planeación, gestión y evaluación de la iteración, la planeación del entorno por iteración y definición de la visión del proyecto.

## Bibliografía

- [1] *ISO/IEC 29110-3: Software Engineering – Lifecycle Profiles for Very Small Enterprises (VSE) – Part 3: Assessment Guide*, ISO (International Organization for Standardization) SC7-WG24 Group, 2011. [vii](#), [5](#), [13](#), [25](#)
- [2] I. Jacobson, G. Booch, and J. Rumbaugh, *The Unified Software Development Process*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999. [x](#), [1](#), [5](#), [6](#), [10](#), [11](#), [25](#), [29](#), [32](#), [33](#), [34](#), [35](#), [36](#), [37](#), [38](#), [39](#), [40](#), [46](#), [49](#), [52](#), [66](#), [73](#), [74](#), [86](#), [93](#), [94](#), [97](#), [105](#), [133](#)
- [3] A. Cockburn, “Selecting a project’s methodology,” *IEEE Softw.*, vol. 17, no. 4, pp. 64–71, Jul. 2000. [Online]. Available: <http://dx.doi.org/10.1109/52.854070> [1](#)
- [4] P. B. Petersen, “Total quality management and the deming approach to quality management,” *Journal of Management History*, vol. 5, pp. 468–488, Dec 1, 1999. [1](#)
- [5] I. Sommerville, *Software Engineering (7th Edition)*. Pearson Addison Wesley, 2004. [1](#), [10](#)
- [6] —, *Software Engineering*, ser. International computer science series. Addison-Wesley Publishing Company, 2007. [Online]. Available: <http://books.google.com.co/books?id=B7idKfL0H64C> [1](#), [10](#)
- [7] F. J. P. Correa, F. García, and M. Piatinni, “Revisión sistemática de mejora de procesos software en micro, pequeñas y medianas empresas,” *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, vol. 2, pp. 6–23, 2006. [Online]. Available: <http://www.redalyc.org/articulo.oa?id=92220103> [1](#), [3](#), [4](#), [5](#), [22](#), [66](#)
- [8] M. Halling, W. Zuser, M. Kohle, and S. Biffel, “Teaching the unified process to undergraduate students,” in *Software Engineering Education and Training, 2002. (CSEE T 2002). Proceedings. 15th Conference on, 2002*, pp. 148–159. [1](#), [25](#), [38](#)
- [9] C. Laporte, S. Alexandre, and R. O’Connor, “A software engineering lifecycle standard for very small enterprises,” in *Software Process Improvement*, R. O’Connor, N. Baddoo, K. Smolander, and R. Messnarz, Eds. Springer

- Berlin Heidelberg, 2008, vol. 16, ch. 12, pp. 129–141. [Online]. Available: [http://dx.doi.org/10.1007/978-3-540-85936-9\\_12](http://dx.doi.org/10.1007/978-3-540-85936-9_12)
- [10] C. Y. Laporte, N. Berrhouma, M. Doucet, and E. Palza-Vargas, “Measuring the cost of software quality of a large software project at bombardier transportation: A case study,” *Software Quality Professional*, vol. 14, pp. 14–31, 2012. [Online]. Available: [http://www.etsmtl.ca/Professeurs/claporte/documents/publications/Project-at-bombardier-transportation\\_SQP\\_June-2012.pdf](http://www.etsmtl.ca/Professeurs/claporte/documents/publications/Project-at-bombardier-transportation_SQP_June-2012.pdf)
- [11] R. O’Connor and C. Y. Laporte, “Deploying lifecycle profiles for very small entities: An early stage industry view,” in *11th International SPICE Conference on Process Improvement and Capability dEtermination*. Springer Verlag, 2011. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-21233-8\\_27](http://dx.doi.org/10.1007/978-3-642-21233-8_27)
- [12] R. O’Connor, S. Basri, and G. Coleman, “Exploring managerial commitment towards spi in small and very small enterprises,” in *Systems, Software and Services Process Improvement*, ser. Communications in Computer and Information Science, A. Riel, R. O’Connor, S. Tichkiewitch, and R. Messnarz, Eds. Springer Berlin Heidelberg, 2010, vol. 99, pp. 268–279. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-15666-3\\_24](http://dx.doi.org/10.1007/978-3-642-15666-3_24)
- [13] R. V. O’Connor and C. Laporte, “Using iso/iec 29110 to harness process improvement in very small entities,” in *Systems, Software and Service Process Improvement*, ser. Communications in Computer and Information Science, R. O’Connor, J. Pries-Heje, and R. Messnarz, Eds. Springer Berlin Heidelberg, 2011, vol. 172, pp. 225–235. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-22206-1\\_20](http://dx.doi.org/10.1007/978-3-642-22206-1_20)
- [14] C. Y. Laporte, A. Renault, and S. Alexandre, “The application of international software engineering standards in very small enterprises,” in *Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies*. IGI Global, 2008, pp. 42–70. [Online]. Available: <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-59904-906-9.ch002>
- [15] R. V. O’Connor and C. Y. Laporte, “Towards the provision of assistance for very small entities in deploying software lifecycle standards,” in *Proceedings of the 11th International Conference on Product Focused Software*, ser. PROFES ’10. New York, NY, USA: ACM, 2010, pp. 4–7. [Online]. Available: <http://doi.acm.org/10.1145/1961258.1961259>
- [16] V. Ribaud, P. Saliou, R. O’Connor, and C. Laporte, “Software engineering support activities for very small entities,” in *Systems, Software and Services Process Improvement*, ser. Communications in Computer and Information Science, A. Riel, R. O’Connor, S. Tichkiewitch, and R. Messnarz, Eds. Springer Berlin Heidelberg, 2010, vol. 99, pp. 165–176. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-15666-3\\_15](http://dx.doi.org/10.1007/978-3-642-15666-3_15)
- [17] R. O’Connor and C. Laporte, “Software project management in very small entities with iso/iec 29110,” in *Systems, Software and Services*



- Process Improvement*, ser. Communications in Computer and Information Science, D. Winkler, R. O'Connor, and R. Messnarz, Eds. Springer Berlin Heidelberg, 2012, vol. 301, pp. 330–341. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-31199-4\\_29](http://dx.doi.org/10.1007/978-3-642-31199-4_29) 2, 5, 17, 19
- [18] *ISO/IEC 29110-5-1-1: Ingeniería de Software - Perfiles de ciclo de vida para VSE – Part 5-1-1: Guía de Administración e Ingeniería: Grupo de perfiles genéricos: Perfil básico*, ISO (International Organization for Standardization) SC7-WG24 Group, 2012. 2, 24, 32
- [19] R. V. O'Connor, "Evaluating management sentiment towards iso/iec 29110 in very small software development companies," in *Software Process Improvement and Capability Determination*, ser. Communications in Computer and Information Science, A. Mas, A. Mesquida, T. Rout, R. O'Connor, and A. Dorling, Eds. Springer Berlin Heidelberg, 2012, vol. 290, pp. 277–281. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-30439-2\\_31](http://dx.doi.org/10.1007/978-3-642-30439-2_31) 2
- [20] A. Scott, "Unified process sets it projects on right course," *Computing Canada*, vol. 24, no. 46, pp. 23–23,26, Dec. 1998. [Online]. Available: <http://search.proquest.com/docview/224998829?accountid=41027> 3
- [21] J. A. H. Alegría and R. P.H., "A software process line based on the unified process," in *Computing Congress (CCC), 2012 7th Colombian*, 2012, pp. 1–6. 3
- [22] J. A. H. Alegría and M. C. Bastarrica, "Implementing cmmi using a combination of agile methods," *CLEI Electron. J.*, vol. 9, no. 1, 2006. [Online]. Available: <http://dblp.uni-trier.de/db/journals/cleiej/cleiej9.html#AlegríaB06> 3, 22
- [23] *ISO/IEC 29110-1: Software Engineering – Lifecycle Profiles for Very Small Enterprises (VSE) – Part 1: Overview*, ISO (International Organization for Standardization) SC7-WG24 Group, 2011. 4
- [24] S. Basri and R. V. O'Connor, "Understanding the perception of very small software companies towards the adoption of process standards," in *Systems, Software and Services Process Improvement*, ser. Communications in Computer and Information Science, A. Riel, R. O'Connor, S. Tichkiewitch, and R. Messnarz, Eds. Springer Berlin Heidelberg, 2010, vol. 99, pp. 153–164. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-15666-3\\_14](http://dx.doi.org/10.1007/978-3-642-15666-3_14) 5, 17, 19, 23
- [25] *Assessing the Rational Unified Process against ISO/IEC 15504-5: Information Technology – Software Process Assessment Part 5: An Assessment Model And Indicator Guidance*, Rational Corporation Whitepaper, 2000. 5, 21, 25
- [26] M. Camacho and J. Alegría, "Analizing the viability for adopting the software process line approach in small entities," in *Computing Congress (CCC), 2012 7th Colombian*, Oct 2012, pp. 1–6. 5
- [27] J. A. H. Alegría, "Toward a scientific method in software engineering (position paper)," p. 2. 5

- [28] M. Bunge, *La ciencia: su método y su filosofía*. Siglo Veinte, 1976. [Online]. Available: <http://books.google.com.co/books?id=QehcAAAAMAAJ> 5
- [29] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Software Engineering*, vol. 14, no. 2, pp. 131–164, 2009. [Online]. Available: <http://dx.doi.org/10.1007/s10664-008-9102-8> 6, 67
- [30] M. Shaw, "Writing good software engineering research papers: Minitutorial," in *Proceedings of the 25th International Conference on Software Engineering*, ser. ICSE '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 726–736. [Online]. Available: <http://dl.acm.org/citation.cfm?id=776816.776925> 6
- [31] P. H. Feiler and W. S. Humphrey, "Software process development and enactment: concepts and definitions," in *Software Process, 1993. Continuous Software Process Improvement, Second International Conference on the*. IEEE, Feb. 1993, pp. 28–40. [Online]. Available: <http://dx.doi.org/10.1109/spcon.1993.236824> 10
- [32] U. Becker-Kornstaedt, "Descriptive software process modeling - how to deal with sensitive process information," *Empirical Software Engineering*, vol. 6, no. 4, pp. 353–367, 2001. [Online]. Available: <http://dx.doi.org/10.1023/A%3A1011986902298> 10
- [33] J. Münch, O. Armbrust, M. Kowalczyk, and M. Soto, *Software Process Definition and Management*, ser. The Fraunhofer IESE Series on Software and Systems Engineering. Springer, 2012. [Online]. Available: [http://books.google.com.co/books?id=S\\_8Yf1wTKNoC](http://books.google.com.co/books?id=S_8Yf1wTKNoC) 11
- [34] F. J. P. Correa, F. García, and M. Piatini, *Calidad de Sistemas de Información*, Rama, Ed., 2011, vol. 0. 11, 67
- [35] *ISO/IEC 12207: Systems and software engineering - Software Lifecycle Process*, ISO (International Organization for Standardization), 2008. 13
- [36] *ISO/IEC 15504-1: Software Process Improvement and Capability Determination - Concepts and Introductory Guide*, ISO (International Organization for Standardization), 1998. 13
- [37] *ISO/IEC 15504-5: Software Process Improvement and Capability Determination - An Assessment Model and Indicator Guidance*, ISO (International Organization for Standardization), 1999. 14, 26, 30
- [38] J. A. Hurtado, M. C. Bastarrica, and A. Bergel, "Analyzing software process models with avispa." in *ICSSP*, D. Raffo, D. Pfahl, and L. Zhang, Eds. ACM, 2011, pp. 23–32. [Online]. Available: <http://dblp.uni-trier.de/db/conf/ispw/icssp2011.html#AlegriaBB11> 16, 57

- [39] L. Rodríguez-Martínez, M. Mora, and F. Álvarez, “A descriptive/comparative study of the evolution of process models of software development life cycles (pm-sdlcs),” in *Computer Science (ENC), 2009 Mexican International Conference on*, 2009, pp. 298–303. [20](#)
- [40] B. Day, S. C. Ke-Zun, L. Lovelock, and C. Lutteroth, “Climbing the ladder: Cmmi level 3,” in *Proceedings of the 13th IEEE International Conference on Enterprise Distributed Object Computing*, ser. EDOC’09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 79–88. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1719357.1719366> [20](#)
- [41] K. Sägesser, B. Joseph, and R. Grau, “Introducing an iterative life-cycle model at credit suisse it switzerland,” *Software, IEEE*, vol. 30, no. 2, pp. 68–73, 2013. [20](#), [22](#)
- [42] G. Jeyaraman, K. Krishnamurthy, and V. Raveendra, “Reengineering legacy application to e-business with modified rational unified process,” in *Software Maintenance and Reengineering, 2003. Proceedings. Seventh European Conference on*, 2003, pp. 143–150. [20](#), [22](#)
- [43] M. E. Morales-Trujillo, H. Oktaba, T. Ventura, and R. Torres, “From moprosoft level 2 to iso/iec 29110 basic profile: Bridging the gap,” *CLEI Electronic Journal*, vol. 16, pp. 3 – 3, 04 2013. [Online]. Available: [http://www.scielo.edu.uy/scielo.php?script=sci\\_arttext&pid=S0717-50002013000100003&nrm=iso](http://www.scielo.edu.uy/scielo.php?script=sci_arttext&pid=S0717-50002013000100003&nrm=iso) [21](#), [25](#), [29](#), [30](#)
- [44] V. Ribaud and P. Saliou, “Process assessment issues of the iso/iec 29110 emerging standard,” in *Proceedings of the 11th International Conference on Product Focused Software*, ser. PROFES ’10. New York, NY, USA: ACM, 2010, pp. 24–27. [Online]. Available: <http://doi.acm.org/10.1145/1961258.1961264> [21](#), [25](#), [97](#)
- [45] R. Motschnig-Pitrik, “Employing the unified process for developing a web-based application — a case-study,” in *Practical Aspects of Knowledge Management*, ser. Lecture Notes in Computer Science, D. Karagiannis and U. Reimer, Eds. Springer Berlin Heidelberg, 2002, vol. 2569, pp. 97–113. [Online]. Available: [http://dx.doi.org/10.1007/3-540-36277-0\\_10](http://dx.doi.org/10.1007/3-540-36277-0_10) [22](#), [95](#)
- [46] G. Schneider and J. P. Winters, *Applying Use Cases: A Practical Guide*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1998. [23](#)
- [47] R. Falbo, L. Borges, and F. Valente, “Using knowledge management to improve software process performance in a cmm level 3 organization,” in *Quality Software, 2004. QSIC 2004. Proceedings. Fourth International Conference on*, Sept 2004, pp. 162–169. [23](#)
- [48] P. Borges, P. Monteiro, and R. Machado, “Mapping rup roles to small software development teams,” in *Software Quality. Process Automation in Software Development*, ser. Lecture Notes in Business Information Processing, S. Biffel,

- D. Winkler, and J. Bergsmann, Eds. Springer Berlin Heidelberg, 2012, vol. 94, pp. 59–70. [Online]. Available: [http://dx.doi.org/10.1007/978-3-642-27213-4\\_5](http://dx.doi.org/10.1007/978-3-642-27213-4_5) 23
- [49] G. Hanssen, H. Westerheim, and F. Bjørnson, “Tailoring rup to a defined project type: A case study,” in *Product Focused Software Process Improvement*, ser. Lecture Notes in Computer Science, F. Bomarius and S. Komi-Sirviö, Eds. Springer Berlin Heidelberg, 2005, vol. 3547, pp. 314–327. [Online]. Available: [http://dx.doi.org/10.1007/11497455\\_26](http://dx.doi.org/10.1007/11497455_26) 23
- [50] *ISO/IEC 15504-2: Software Process Improvement and Capability Determination - Performing an Assessment*, ISO (International Organization for Standardization), 2003. 26
- [51] “Free rup templates,” 2014. [Online]. Available: <http://www.findthatzip-file.com/search-43193485-hZIP/winrar-winzip-download-free-rup-templates.zip.htm> 35, 37, 39, 56
- [52] G. Hanssen, F. Bjørnson, and H. Westerheim, “Tailoring and introduction of the rational unified process,” in *Software Process Improvement*, ser. Lecture Notes in Computer Science, P. Abrahamsson, N. Baddoo, T. Margaria, and R. Messnarz, Eds. Springer Berlin Heidelberg, 2007, vol. 4764, pp. 7–18. [Online]. Available: [http://dx.doi.org/10.1007/978-3-540-75381-0\\_2](http://dx.doi.org/10.1007/978-3-540-75381-0_2) 38
- [53] “Proceso unificado para pequeñas entidades de desarrollo de software (up-vse),” Grupo IDIS, Universidad del Cauca. [Online]. Available: <http://artemisa.unicauca.edu.co/~jjalvarezl/UP-VSE/> 39
- [54] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, ser. The Addison-Wesley object technology series. Pearson Education, 1999. [Online]. Available: <http://books.google.co.uk/books?id=a5J49FoFKq8C> 53
- [55] R. Pressman, R. Martín, and L. Aguilar, *Ingeniería del software: un enfoque práctico*. McGraw-Hill, 1997. [Online]. Available: <http://books.google.com.co/books?id=8UV5jxkuBZIC> 57
- [56] “Proceso de desarrollo iterativo del grupo de ingeniería automática (prodigia).” Grupos IDIS - IA, Universidad del Cauca. [Online]. Available: <http://artemisa.unicauca.edu.co/~jjalvarezl/PRODIGIA/> 66, 80
- [57] W. L. P. Yopez, C. Collazos, and V. M. R. Penichet, “Entorno colaborativo de apoyo a la mejora de procesos de software en pequeñas organizaciones de software,” *Dyna*, vol. 80, pp. 40–48, 2013. 72, 95

## Anexo A

# PRODIGIA: Ciclo de vida

### A.1. Introducción

Este capítulo presenta el modelo de proceso PRODIGIA, que ha nacido y ha sido ejecutado en ambientes reales VSE tras la necesidad de satisfacer los requisitos sobre el desarrollo de proyectos de ingeniería de software.

PRODIGIA (PROceso de Desarrollo Industrial del Grupo Automática Industrial) se caracteriza por ser un proceso de pocos roles, flexible y adaptable, simple y fácil de aprender dado a que utiliza un lenguaje sencillo y corto para transmisión del conocimiento acerca del flujo de las tareas y manejo de plantillas, y hace uso de buenas prácticas ágiles bien conocidas y referenciadas frecuentemente tanto en entornos académicos como en la industria. Dado a su creación en un entorno universitario en proyectos de tesis de maestría y pregrado éste proceso es apto para presentación de tesis utilizando plantillas propias de la Universidad del Cauca para presentación de artefactos tanto de investigación para tesis como de desarrollo de software.

Al igual que en el capítulo 4, el lenguaje SPEM 2.0 ha sido utilizado para la especificación del proceso bajo el entorno de desarrollo Eclipse Process Framework Composer (EPF), una herramienta CASE usada para su diseño y formalización.

### A.2. Características del proceso

- **Guiado por casos de uso, centrado en la arquitectura, iterativo e incremental:** Al igual que el trabajo I. Jacobson et. al. [2], PRODIGIA cuenta con éstas 3 características que ilustran partes del ciclo de vida del proceso desde una vista mas general mostrando tanto su flujo de ejecución como de la interrelación de

los artefactos mas importantes.

- **Utiliza disciplinas y fases como vistas del proceso:** Las disciplinas en PRODIGIA son conocidas como actividades principales para mejor comprensión del lector e ilustran lo que el usuario realizará para cumplir con el proceso, las fases brindan una vista de gestión, ilustran en que parte del proceso se encuentra el proyecto y los principales entregables al final de cada una.
- **Adaptable:** Para tomar acciones sobre la naturaleza cambiante y única de cada proyecto, PRODIGIA es adaptable gracias la filosofía iterativa e incremental que permite la planeación de la iteración antes de su comienzo y las vistas de proceso por fases y disciplinas del Proceso Unificado que conservan su cualidad adaptable dado a la posibilidad de modificación de sus componentes como las actividades principales y de soporte lo que influye en el cambio de los componentes de proceso compatibles con los del UP.
- **Simple:** Al contener pocos roles y tareas acordadas con los participantes, PRODIGIA es un proceso fácil de entender y contiene artefactos mínimos para la realización del proceso y pocos artefactos opcionales (en su mayoría diagramas de software). También se incluye un lenguaje simple sobre la explicación de cada tarea y ejemplos sencillos de los artefactos mas importantes.
- **Orientado a la realización de proyectos académicos:** PRODIGIA, a parte de orientar a los proyectos de software de las VSE brinda una guía para el cumplimiento de artefactos en su mayoría partes de la monografía y del anteproyecto de grado.
- **Mismas actividades, diferentes fases:** Existen tareas que se repiten en diversas fases, éste proceso aprovecha esta ventaja a su favor, evitando el re-trabajo de memorizar e interiorizar nuevas actividades y configuraciones de los elementos del proceso.
- **Orientado a la realización de simuladores de software:** Este proceso también contempla la construcción de proyectos de simulación de software al requerir artefactos como la construcción de un modelo matemático que soporte el movimiento de diversos objetos (generalmente tridimensionales) en el software.

### A.3. Ciclo de vida

PRODIGIA, al igual que el Proceso Unificado consta de 2 vistas de proceso (disciplinas, que en este proceso son llamadas actividades principales y fases). La figura A.1, ilustra gráficamente el ciclo de vida de éste proceso.

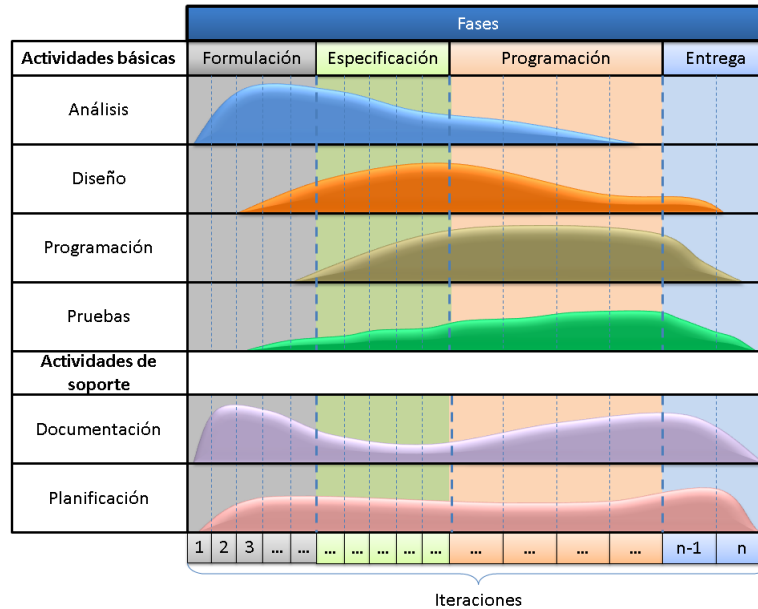


Figura A.1: Ciclo de vida de PRODIGIA

PRODIGIA se divide en 2 tipos de **actividades principales**, en este caso las **actividades básicas** que se encargan de abordar los artefactos software mínimos y las **actividades de soporte** que aportan con el seguimiento y control del proyecto y con el desarrollo de otros artefactos y el soporte de algunos que producen las actividades básicas, ver la sección A.3.2, para mas información de las actividades principales.

### A.3.1. Fases:

También, el ciclo de vida de PRODIGIA consta de 4 fases que son **Formulación**, **Validación**, **Programación** y **Entrega**, como lo ilustra la figura A.2.

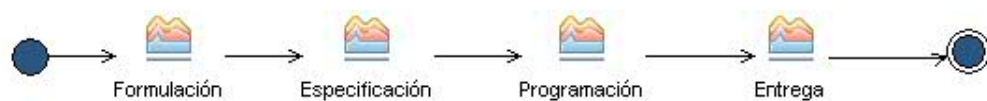


Figura A.2: Fases de PRODIGIA

#### A.3.1.1. Fase de Formulación:

La fase de formulación inicia el proyecto, construye artefactos como el anteproyecto de grado como hito de ésta fase, documenta adecuadamente los requisitos en casos de uso y los organiza por orden de prioridad, documenta los resultados constantemente e inicia el modelado de los planos arquitectónicos de software con el diagrama de dominio. La figura A.3 ilustra gráficamente ésta fase.



Figura A.3: Fase de formulación

**Actividad Construcción del anteproyecto:** Principalmente se enfoca en la construcción del anteproyecto enfatizando en las reuniones periódicas con el tutor de tesis, la realización de objetivos y planteamiento del problema y por ultimo del cronograma, del presupuesto y del acta de propiedad. La figura A.4 ilustra ésta actividad.

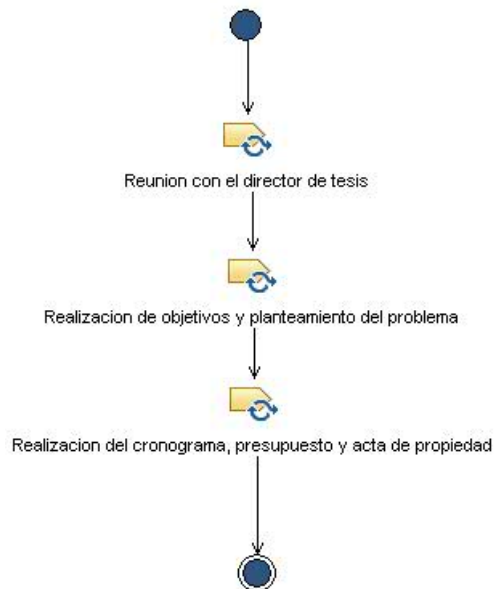


Figura A.4: Actividad Construcción del Anteproyecto



### A.3.1.2. Fase de Especificación:

La fase de especificación es la encargada de definir los requisitos funcionales y no funcionales documentandolos en la monografía, y define al final cuales son las especificaciones técnicas requeridas del sistema, la figura A.5 ilustra gráficamente ésta fase.



Figura A.5: Fase de Especificación

**Iteraciones de la fase de especificación:** las actividades incluidas en una iteración se repiten tantas veces como haya sido definido al inicio de la fase cumpliendo con el cronograma de actividades del proyecto. La figura A.6 ilustra gráficamente el contenido de una iteración.

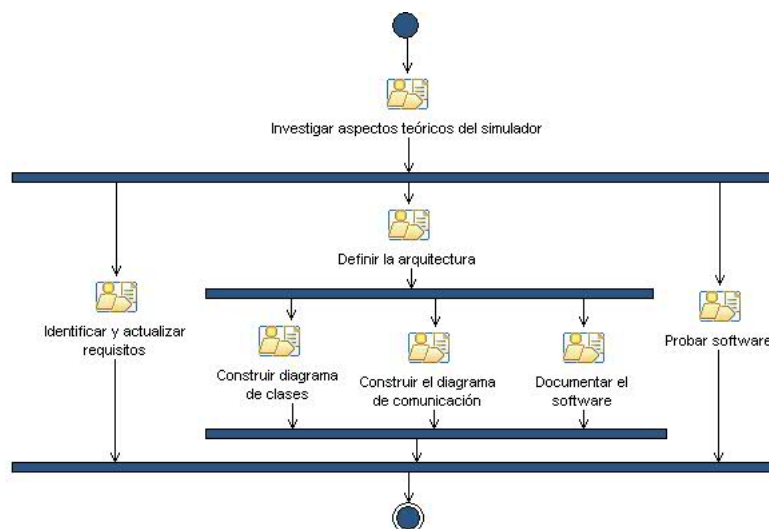


Figura A.6: Iteraciones de la fase de especificación

**Actividad Investigar aspectos teóricos del simulador:** Esta actividad se encarga de la construcción del modelado matemático, que para el desarrollo de simuladores software consiste en la base teórica del movimiento de los componentes tridimensionales móviles. La figura A.7 ilustra el contenido de esta actividad.



Figura A.7: Actividad investigar aspectos teóricos del simulador

**Actividad Definir la arquitectura:** PRODIGIA, al estar centrado en la arquitectura, considera necesaria la construcción de modelos de software con el fin de lograr un mapa de la estructura y comunicación de los distintos componentes del software. La figura A.8 ilustra las tareas a realizar en ésta fase.



Figura A.8: Actividad Definir la arquitectura

**Actividad Identificar y actualizar requisitos:** la identificación y actualización constante de requisitos hace parte fundamental de los objetivos del desarrollo de software y así lograr la satisfacción de las expectativas del cliente. La figura A.9 ilustra detalladamente el contenido de ésta actividad.

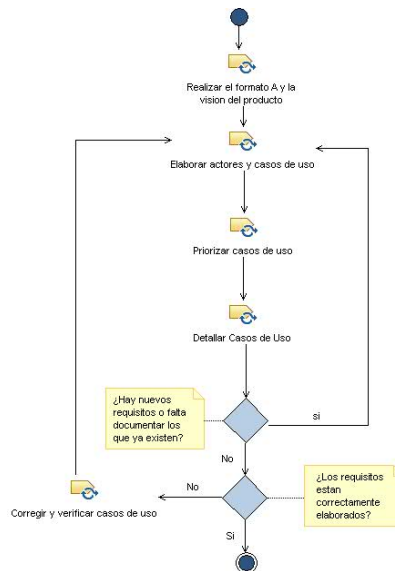


Figura A.9: Actividad Identificar y actualizar requisitos

**Actividad Probar software:** Probar los artefactos es parte básica para asegurar la coherencia y trazabilidad con los requisitos capturados. La figura A.10 ilustra ésta actividad.

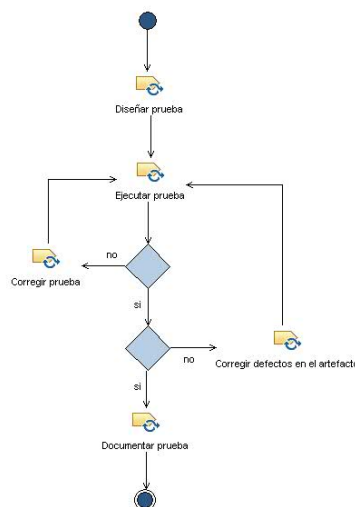


Figura A.10: Actividad Probar software

**Actividad Construir el diagrama de clases:** Parte de la construcción de la arquitectura de software radica en la elaboración del diagrama de clases, la figura A.11 ilustra gráficamente el contenido de ésta actividad

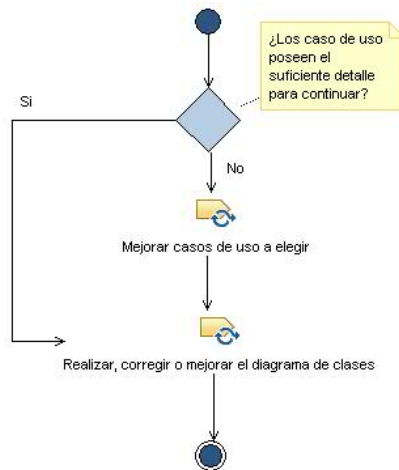


Figura A.11: Actividad Construir el diagrama de clases

**Actividad Construir el diagrama de comunicación:** El diagrama de comunicación ilustra detalles específicos de la construcción y los llamados a objetos con su respectivo orden que pueden ser de utilidad al momento de elaborar patrones de construcción y de ilustrar cómo han sido construidas ciertas secciones que ejecutan funcionalidades deseadas en otras secciones del software. La figura A.12 ilustra gráficamente esta actividad.

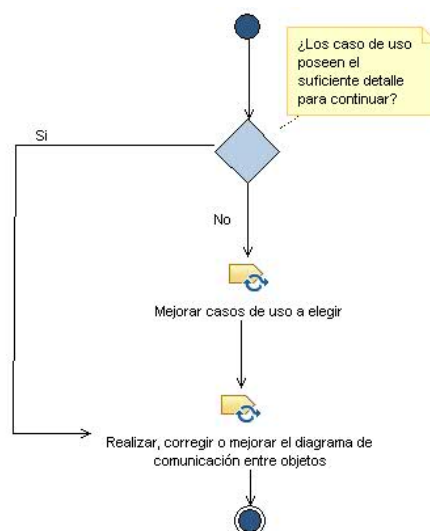


Figura A.12: Actividad Construir el diagrama de comunicación

**Actividad Documentar el software:** La figura A.13 ilustra gráficamente como documentar el software. Esta actividad se asegura de incluir artefactos tanto de ámbito académico como de software, cumpliendo con la entrega de los principales artefactos como el **anteproyecto**, el **artículo** y la **monografía**.

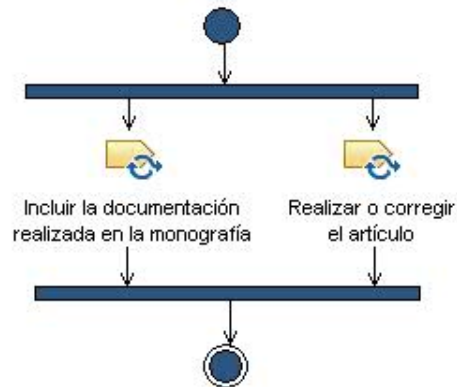


Figura A.13: Actividad Documentar el software

### A.3.1.3. Fase de Programación:

En esta fase se enfocan gran parte de los esfuerzos a la codificación del software basándose en los diagramas arquitectónicos elaborados en la fase anterior entregando al final un prototipo de software validado, la figura A.14 ilustra gráficamente ésta fase.



Figura A.14: Fase de programación

**Iteraciones de la fase de programación:** al igual que en las iteraciones de la fase de especificación incluidas en una iteración, estas actividades se repiten tantas veces como haya sido definido al inicio de la fase manteniendo tiempos de entrega. La figura A.15 ilustra gráficamente el contenido de una iteración.

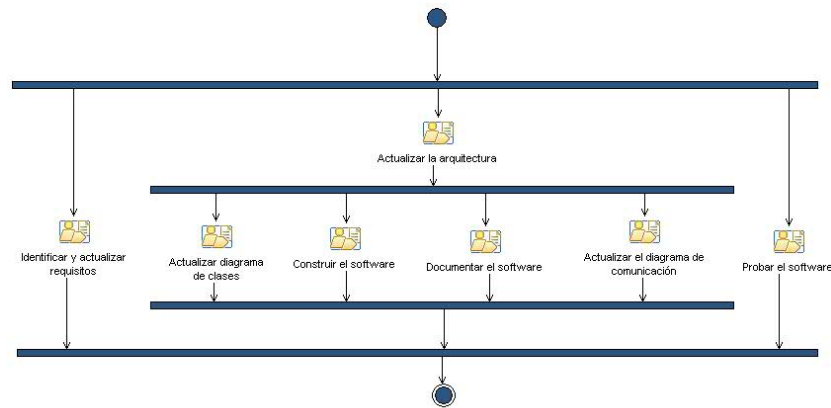


Figura A.15: Iteraciones de la fase de programación

**Actividad Actualizar la arquitectura:** Ver la sección “Actividad Definir la arquitectura” en A.3.1.2.

**Actividad Identificar y actualizar requisitos:** Esta actividad a diferencia de la actividad “Identificar y actualizar requisitos” en A.3.1.2 deja a un lado las actividades de realización del formato A y visión del producto y se enfoca en corregir y verificar los casos de uso en paralelo con actividades para encontrar y documentar nuevos requisitos. La figura A.16 ilustra gráficamente ésta actividad.

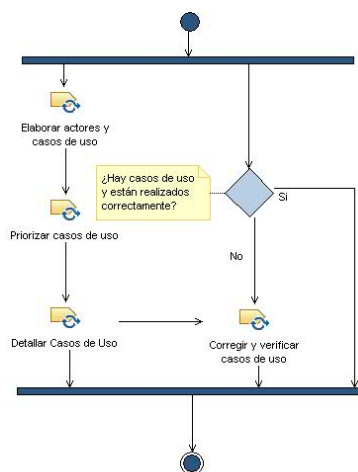


Figura A.16: Actividad Identificar y actualizar requisitos

**Actividad Probar software:** Ver la sección “Actividad Probar software” en A.3.1.2.

**Actividad Actualizar diagrama de clases:** Ver la sección “Construir diagrama de clases” en A.3.1.2.

**Actividad Construir el software:** La construcción del diagrama de software es en pocas palabras el mapeo a código de los planos construidos anteriormente y la corrección los defectos en caso de ser encontrados en el código. La figura A.17 ilustra en detalle la ejecución de ésta actividad.

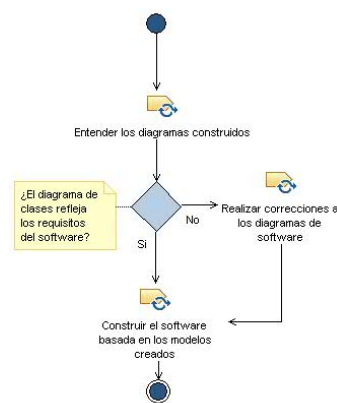


Figura A.17: Actividad Construir el software

**Actividad Documentar el software:** A diferencia de la actividad “Documentar el software” en A.3.1.2, ésta contempla tareas como la realización de los manuales del software (de instalación y de usuario) que se consideran como fundamentales dentro de la monografía. La figura A.18 ilustra la ejecución de ésta actividad.

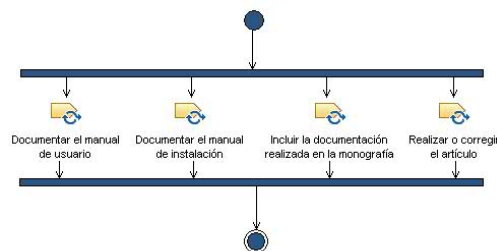


Figura A.18: Actividad Documentar el software

**Actividad Actualizar el diagrama de comunicación:** Ver la sección “Construir el diagrama de comunicación” en A.3.1.2.

#### A.3.1.4. Fase de Entrega:

La fase de entrega es la fase final de PRODIGIA, y se encarga de terminar y liberar (como hito) todos los entregables indicados a lo largo del proceso, la figura A.19 ilustra el contenido de ésta fase.



Figura A.19: Fase de entrega

**Iteraciones de la fase de entrega:** Con el fin de continuar el trabajo sin mayor esfuerzo en la interiorización y ejecución del proceso se ha conservado el mismo conjunto de actividades con la misma configuración que en A.3.1.3. La figura A.20 ilustra gráficamente esta configuración de actividades.

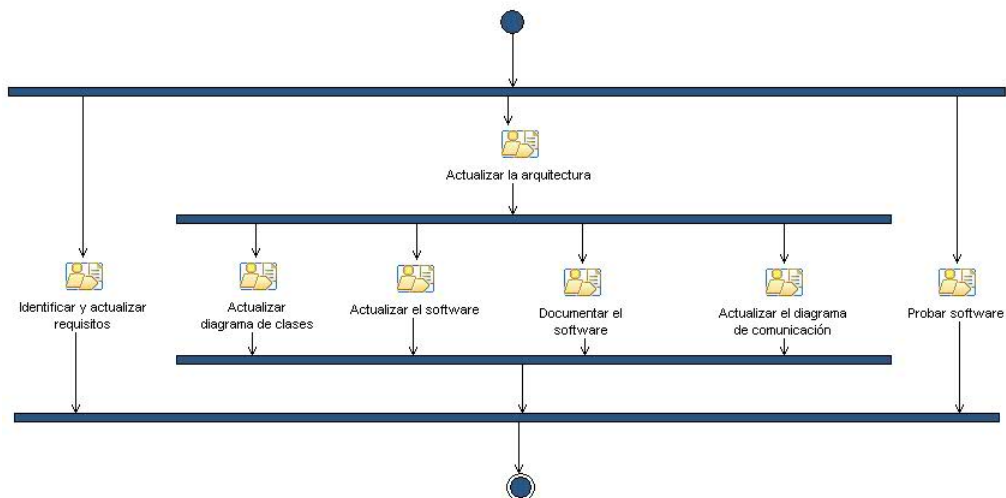


Figura A.20: Actividad Documentar el software



**Actividad Actualizar la arquitectura:** Ver la sección “Actividad Definir la arquitectura” en [A.3.1.2](#).

**Actividad Probar software:** Ver la sección “Actividad Probar software” en [A.3.1.2](#).

**Actividad Actualizar diagrama de clases:** Ver la sección “Construir diagrama de clases” en [A.3.1.2](#).

**Actividad Actualizar el software:** Ver la sección “Construir el software” en [A.3.1.3](#).

**Actividad Documentar el software:** Ver la sección “Documentar el software” en [A.3.1.3](#)

**Actividad Actualizar el diagrama de comunicación:** Ver la sección “Construir el diagrama de comunicación” en [A.3.1.2](#).

**Liberar entregables:** Esta actividad consiste en terminar toda la documentación que se necesita entregar dentro del proyecto, liberar la monografía integrando todas sus partes, el artículo y el código del programa software. la figura [A.21](#) ilustra la configuración de esta actividad.

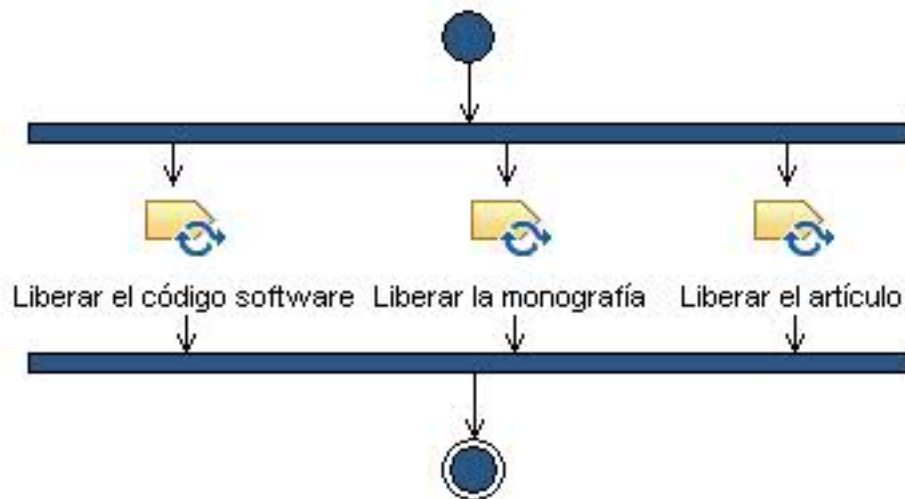


Figura A.21: Actividad Liberar entregables

### A.3.2. Actividades principales

PRODIGIA entiende las disciplinas con el nombre de **actividades principales** de las cuales se derivan las:

- **Actividades básicas:** son parte del conjunto mínimo de elementos de proceso para el funcionamiento adecuado para cumplir con el objetivo de la finalización exitosa de un proyecto con los artefactos básicos para el desarrollo de software.
  - **Análisis:** Se define el inicio del proyecto, la exploración inicial, aportando en gran parte al desarrollo del anteproyecto de grado y en el estado del arte incluyendo las bases teóricas (formulas) para simuladores, y los casos de uso en la monografía.
  - **Diseño:** La actividad de diseño se encarga de convertir los requisitos (casos de uso y los componentes teóricos) en los planos arquitectónicos matemáticos y del software.
  - **Programación:** Esta actividad convierte los planos arquitectónicos en líneas de código siguiendo buenos principios de programación, preferiblemente basándose en prácticas ágiles como la propiedad colectiva del código y programación en pares.
  - **Pruebas:** Esta actividad se ha documentado de forma general de modo que cualquier artefacto pudiera ser probado, de esta manera se cubren errores de código y ejecución y trazabilidad con los requisitos al mismo tiempo.
- **Actividades de soporte:** Complementan el proceso para la entrega de artefactos extra que puedan surgir en el transcurso del proyecto o para el seguimiento y control tanto de los artefactos como de su ejecución.
  - **Documentación:** La documentación en PRODIGIA se considera una actividad que demanda gran esfuerzo dado a que en proyectos de la academia el esfuerzo se centra en 3 artefactos principales: la monografía, el anteproyecto y el artículo.
  - **Planificación:** La planificación se encarga de la organización del proyecto que se pueda culminar todo el trabajo requerido en el proyecto dentro del alcance, el tiempo, la planificación en este caso se encarga de organizar el cronograma de actividades.

### A.3.3. Roles:

- **Tesista:** Es el encargado de realizar todos los entregables que corresponden a su proyecto de software.
- **Tutor de tesis:** Es el consultor, experto en el tema particular a tratar a lo largo del proyecto, encargado de brindar soporte al desarrollo del tesista.

## Anexo B

# Diseño de los instrumentos de recolección, caso PRODIGIA

### B.1. Introducción

El diseño de los instrumentos de recolección ha sido pensado en tratar de capturar la mayor cantidad de datos posibles para cada uno de los elementos de proceso del estudio de caso. Cada uno de los instrumentos captura elementos para las métricas definidas para el estudio de caso de la sección 5.3.3.6. A continuación se ilustra el diseño de cada instrumento:

### B.2. Encuestas

La obtención de datos se ha basado tomando en cuenta 3 tipos de entrevistas realizadas que han sido catalogadas por cada rol desempeñado del estudio de caso (calidad, quienes han definido el proceso, gestión, quienes validan y verifican el trabajo realizado o los tutores de tesis y desarrollo, quienes ejecutaron el proceso PRODIGIA o los tesistas)

Cada pregunta está enfocada en calificar el proceso desde los 3 roles que participaron en la definición y ejecución del proceso, esto es, el tutor de tesis que participó en la parte de gestión de los proyectos que ejecutaron los 2 tesistas, quienes participaron en el proceso, uno en ejecución y otro en definición del proceso respectivamente calificando en un rango de 0 a 5 en cada pregunta mas una pregunta adicional (campo abierto) que calificó de manera cualitativa tanto el proceso como la misma

encuesta. Cada respuesta obtenida y calificada numéricamente se promedió, lo que dió el valor de PRODIGIA para cada participante. Cada pregunta fue realizada de forma general al proceso a excepción de la encuesta de tesisistas que ejecutaron el proceso la cual, además de las preguntas realizadas de forma general, indagó por la completitud que tenía cada estudiante de cada tarea del proceso en contraste con su respectivo proyecto. A continuación se ilustra cada encuesta realizada:

### B.2.1. Encuestas para tesisistas:

Las 2 encuestas que se mostrarán a continuación se basa en la asistencia de los estudiantes a la primera reunión de socialización del proceso, donde la primera se utiliza para obtener una aproximación del porcentaje de los estudiantes del grupo que si han desarrollado software siguiendo buenas prácticas. La segunda encuesta asegura la recolección de la opinión de los estudiantes que asistieron a la primera reunión de PRODIGIA sobre el proceso.

#### B.2.1.1. Encuesta general:

El tesisista indica de 0 (no aplicado) a 5 (completamente aplicado) el nivel de completitud de cada actividad mencionada a continuación (la numeración adjunta a la tarea es el respectivo indicador), la tabla B.1 ilustra la encuesta:

Pregunta	Calificación					
	0	1	2	3	4	5
1.1.1 Reunión con el director de tesis						
1.1.2 Realización de objetivos y planteamiento del problema						
1.1.3 Realización del cronograma, presupuesto y acta de propiedad.						
2.1.1 Realizar el formato A y la visión del producto						
2.2.1.1 Construcción del modelo matemático						
2.2.2.1 Elaborar actores y casos de uso						
2.2.2.2 Priorizar casos de uso						
2.2.2.3 Detallar Casos de Uso						
2.2.2.4 Corregir y verificar casos de uso						
2.2.3.1 Escoger los casos de uso a diseñar						
2.2.3.2 Mejorar casos de uso a elegir						
2.2.3.3 Realizar, corregir o mejorar el modelo de diseño del sistema						
2.2.3.4 Construir o actualizar los diagramas arquitectónicos						
2.2.4.1 Entender los diagramas construidos						
2.2.4.2 Realizar correcciones a los diagramas de software						
2.2.4.3 Construir el software basado en los modelos creados						
2.2.5.1 Incluir la documentación realizada en la monografía						
2.2.5.2 Realizar o corregir el artículo						
2.2.5.3 Documentar el manual de instalación						
2.2.5.4 Documentar el manual de usuario						
2.2.6.1 Diseñar prueba						
2.2.6.2 Ejecutar prueba						
2.2.6.3 Corregir prueba						
2.2.6.4 Corregir defectos en el artefacto						
2.2.6.5 Documentar prueba						
4.2.1 Liberar el código software						
4.2.2 Liberar el artículo						
4.2.3 Liberar la monografía						
E1) ¿Cual ha sido la utilidad de PRODIGIA en el desarrollo de su proyecto?						
E2) ¿Los integrantes del equipo que ejecutaron alguna parte de este proceso han logrado sus objetivos?						
E3) ¿El proceso PRODIGIA es coherente con la realidad de la tesis que están desarrollando?						
E4) ¿Cuanto esfuerzo requirió la comprensión del proceso PRODIGIA?. 5 (fácil de comprender), 0 (muy desgastante)						
E5) ¿Existe alguna observación sobre el proceso PRODIGIA o sobre la creación de esta encuesta?						

\* Link de la encuesta: <https://docs.google.com/forms/d/1zPTcpwPH6AUPoqISMoFU78GwspzCQerPmeDzVZJSUQ/viewform>

Tabla B.1: Encuesta para obtener la completitud del proceso PRODIGIA

### B.2.2. Encuesta para tutores de tesis:

Las preguntas diseñadas a continuación sirven para la captura del nivel de satisfacción del proceso que ha utilizado para la realización de sus proyectos de grado utilizando el proceso PRODIGIA, por favor califique de 0 a 5 su nivel de satisfacción. La tabla B.2 ilustra la encuesta:

Pregunta	Calificación					
	0	1	2	3	4	5
1) ¿Cual ha sido la utilidad de PRODIGIA en el desarrollo de sus proyectos?						
2) ¿Cual es el nivel de calidad del proceso? (¿Cuales son los criterios de comparación en este caso la calidad del proceso con los otros proyectos de tesis que ha tenido?)						
3) ¿Cuanto esfuerzo requirió la comprensión del proceso PRODIGIA? (de 0 - muy desgastante, a 5 - facil)						
4) ¿Existe alguna observación sobre el proceso PRODIGIA o sobre la creación de esta encuesta?	<i>Campo abierto</i>					

\* Link de la encuesta: <https://docs.google.com/forms/d/1CeipjwAWKDaJHEUrMBzuEgBGJf3FHNk-PhFDIoLJOuk/viewform>

Tabla B.2: Encuesta para obtener el nivel de satisfacción de los tutores de tesis sobre el proceso PRODIGIA

### B.2.3. Encuesta para los integrantes del equipo de procesos:

Las preguntas diseñadas a continuación sirven para la captura del nivel de satisfacción del proceso que ha utilizado para la definición del proceso PRODIGIA, por favor califique de 0 a 5 su nivel de satisfacción. La tabla B.3 ilustra la encuesta:

Pregunta	Calificación					
	0	1	2	3	4	5
1) ¿Cual ha sido la utilidad de PRODIGIA en el desarrollo de su proyecto?						
2) ¿Cual es el nivel de calidad del proceso? (¿Cuales son los criterios de comparación en este caso la calidad del proceso con los otros proyectos de tesis que ha tenido?)						
3) ¿Cual ha sido la utilidad de PRODIGIA en el desarrollo de su proyecto?						
4) ¿Cuanto esfuerzo requirió la comprensión del proceso PRODIGIA? (de 0 - muy desgastante, a 5 - facil)						
5) ¿Existe alguna observación sobre el proceso PRODIGIA o sobre la creación de esta encuesta?	<i>Campo abierto</i>					

\* Link de respuesta: <https://docs.google.com/forms/d/1vWbYW8zSDEntuXeeQDMYeLX2bwtX6k8VRxZNTv3KNr4/viewform>

Tabla B.3: Encuesta para obtener el nivel de satisfacción del equipo de procesos sobre el proceso PRODIGIA

## B.3. Artefactos

Los artefactos utilizados como instrumentos de recolección principalmente fueron tesis de estudiantes graduados que sirvieron de insumos para la definición del proceso PRODIGIA, principalmente en la elección de los modelos arquitectónicos para software que fueran más útiles y brindaran mas información de la construcción y definición del código, se han anexado algunos ejemplos de monografías que sirvieron como insumo en anexos digitales, en la carpeta Monografías de estudiantes graduados.

## **B.4. Modelo de evaluación de la ISO/IEC 29110-3**

El modelo de evaluación de la ISO/IEC 29110-3 propone una evaluación orientada a perfiles de desarrollo de la ISO/IEC 29110-5, en este estudio de caso se realizó con respecto al proceso PRODIGIA lo cual ilustró insumos sobre la aplicabilidad del proceso sobre las VSE en la industria tanto de tareas como de artefactos.

El modelo se encuentra explicado en el capítulo 3 de la monografía y la evaluación en la sección 5.3.3.9 de dicho documento.

## **B.5. Herramienta de soporte a la documentación**

Debido a la continua necesidad del grupo AI de mantener un repositorio propio e interno del manejo de la documentación y comunicación entre tesista y tutor de tesis, se desarrollo una herramienta de gestión documental que ayudara a organizar los documentos de todos los tesistas, del cual se pudo obtener el trabajo en curso de los tesistas del grupo AI, en anexos digitales, en la carpeta Monografías de tesistas que ejecutaron PRODIGIA.

La documentación técnica de la herramienta de soporte documental está en el anexo [C](#).

## **B.6. Herramienta de soporte al trabajo colaborativo COLLAB**

Esta herramienta de soporte fue utilizada con el fin de lograr una mejor comunicación con los integrantes del grupo AI que manifestaban la necesidad de poder trabajar sus tesis desde otros lugares distintos al campus universitario. La herramienta COLLAB cuenta con serie de funcionalidades que se intentaron aprovechar con el fin de mitigar o evitar la materialización de riesgos, entre ellos trabajo incompleto o sin iniciar, comunicación poco frecuente o falta de organización de ideas sobre la construcción del proceso PRODIGIA.

Las figuras [B.1](#) y [B.2](#) ilustra capturas de pantalla de esta herramienta.

## UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras de Software Basado en el Proceso Unificado



Figura B.1: Instancia de COLLAB para ser utilizado en PRODIGIA (Menú principal)

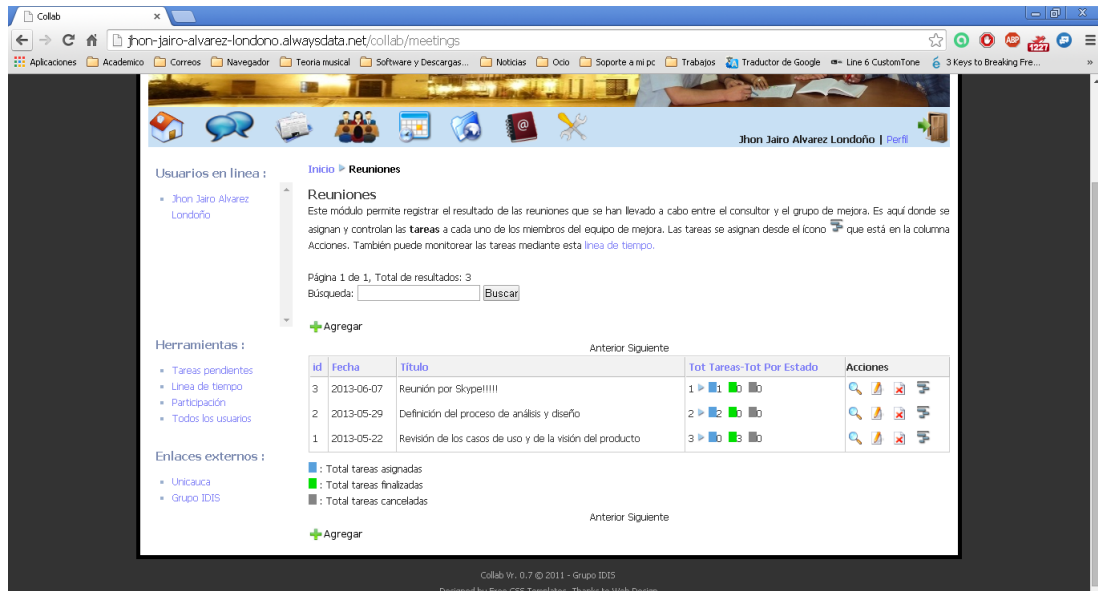


Figura B.2: Instancia de COLLAB para ser utilizado en PRODIGIA (Reuniones y estado de las tareas)

## **Anexo C**

# **Especificación e implementación de la herramienta en Drupal**

### **C.1. Resumen**

El sistema debe brindar soporte a 2 roles, los tutores de tesis y los tesisistas quienes principalmente mantienen interacción con la herramienta. El sistema debe brindar soporte a una serie de funcionalidades cuyo objetivo es comunicar avances de tesis y tesis desarrolladas que les permita una comunicación mas directa sobre avances de tesis de los usuarios para su respectiva revisión de los artefactos y que las tesis sirvan como ejemplo a la comunidad que tenga los permisos para acceder a la herramienta.

### **C.2. Documentación de la herramienta de soporte**

Al igual que los modelos de proceso anteriormente documentados, se ha optado por la captura de requisitos en forma de casos de uso. Tras reuniones periódicas con ingenieros del grupo AI (Automática Industrial) se ha logrado capturar los requisitos de la herramienta y se han procesado en casos de uso que orientan la funcionalidad de software organizadas en casos de uso:



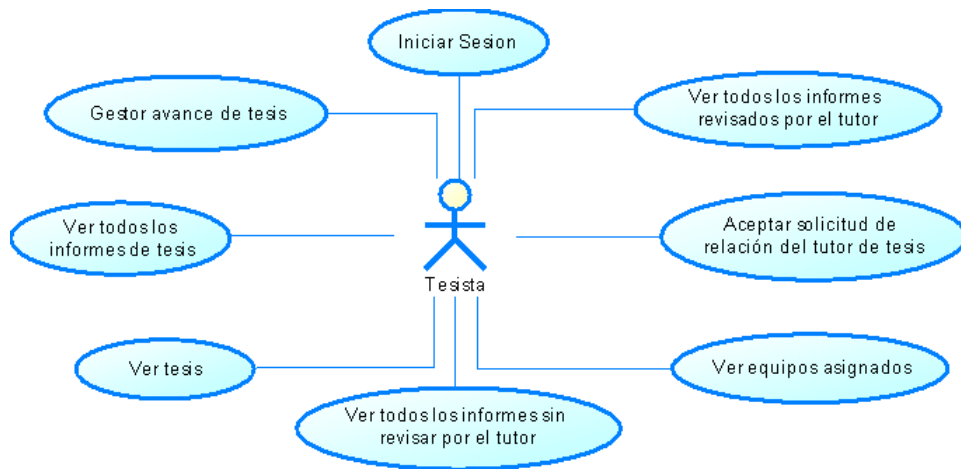


Figura C.1: Modelo de caso de uso, funcionalidades del rol de tesista

Caso de uso	tesista01
Nombre	Gestor avance de tesis
Actor	tesista, tutor de tesis
Resumen	Esta funcionalidad de software permitirá al usuario crear, ver, actualizar y eliminar avances de tesis en donde podrán ser enviadas a su respectivo tutor el cual enviará sugerencias y guías para continuar con su desarrollo
Otros casos de uso involucrados	tesista04, tesista05, tesista06,
Precondiciones	Ninguna
Descripción	<p><b>1) Crear avance de tesis:</b></p> <p>1.1) El usuario accede desde el menú lateral izquierdo "Navegación" haciendo click en agregar contenido, automáticamente es redirigido a agregar un nuevo avance de tesis.</p> <p>1.2) El usuario ingresa los datos del avance de tesis, nombre, descripción, tutor al que va dirigido (obligatorio), check box revisado por el tutor de tesis, check box necesita revisión, el tipo de artefacto a revisar (artefactos del proceso PRODIGIA. Obligatorio), cargar archivo adjunto.</p> <p>1.3) El usuario da click en aceptar para enviar el avance al tutor de tesis o cancelar para descartar esta opción.</p> <p><b>2) Ver avance de tesis:</b> Ver tesista04, tesista05, tesista06</p> <p><b>3) Modificar y eliminar avance de tesis:</b></p> <p>3.1) Realizar el mismo flujo de ejecución de tesista04 / tesista05/ tesista06.</p> <p>3.2) Dando click en el título de algún informe (si hay en la vista) se accede al respectivo avance de tesis.</p> <p>3.3) Dar click en la pestaña editar en la parte superior del informe de tesis.</p> <p>3.4) Modificar los atributos permitidos. Para el tesista están permitidos todos excepto "revisado por el tutor de tesis"</p> <p>3.5) Dar click en guardar para salvar modificaciones y en eliminar para borrar el informe.</p>
Postcondiciones	Ninguna

Tabla C.1: Caso de uso "Gestor avance de tesis".

### C.2.1. Funcionalidades sobre el rol de tesista

La figura C.1 ilustra las funciones que un usuario registrado en el sistema, generalmente el tesista, puede llevar a cabo en la herramienta de soporte. Las siguientes secciones ilustran las funcionalidades de cada caso de uso asociado a este rol:

**Caso de uso "Gestor avance de tesis":** Ver tabla C.1.

**Caso de uso "Ver tesis":** Ver tabla C.2.

**Caso de uso "Ver equipos asignados":** Ver tabla C.3.

**Caso de uso "Ver todos los informes de tesis":** Ver tabla C.4.

**Caso de uso "Ver los informes revisados por el tutor":** Ver tabla C.5.

**Caso de uso "Ver los informes sin revisar por el tutor":** Ver tabla C.6.

**Caso de uso "Aceptar solicitud de relación del tutor de tesis":** Ver tabla C.7.

**Caso de uso "Iniciar sesión":** Ver tabla C.8.

## UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras de Software Basado en el Proceso Unificado

---

<b>Caso de uso</b>	tesista02
<b>Nombre</b>	Ver tesis
<b>Actor</b>	tesista, tutor de tesis
<b>Resumen</b>	Esta funcionalidad de software permitirá al usuario ver las tesis que han sido publicadas por todos los integrantes del grupo de investigación y las tesis en el que él se encuentra asociado.
<b>Otros casos de uso involucrados</b>	tutor01
<b>Precondiciones</b>	Para visualizar las tesis deben existir registros de este tipo en base de datos y tener chequeada la opción "hacer publica"
<b>Descripción</b>	1) <b>Si no existen registros de tesis, entonces crearlas:</b> ejecutar secuencia de creación de tesis del caso de uso tutor01 2) <b>Ver avance de tesis:</b> Esta vista es accesible a través del menú lateral derecho "Consultas del tesista", en el link "Mis tesis", para ver las tesis en las que el tesista está asociado o en el link "Tesis del grupo publicadas" para ver todas las tesis que tienen chequeada la opción "hacer publica"
<b>Postcondiciones</b>	Ninguna

Tabla C.2: Caso de uso "Ver tesis".

<b>Caso de uso</b>	tesista03
<b>Nombre</b>	Ver equipos asignados
<b>Actor</b>	tesista, tutor de tesis
<b>Resumen</b>	Gracias a esta funcionalidad, el tesista podrá consultar los equipos que el tutor de tesis le haya asignado
<b>Otros casos de uso involucrados</b>	tutor02,
<b>Precondiciones</b>	Para que la vista arroje valores, en el sistema deben haber datos de tipo de contenido "Asignar equipos".
<b>Descripción</b>	1) <b>Si no hay datos, crear la asignación de un equipo a un estudiante:</b> ver flujo de creación de asignación de equipos en tutor02 2) <b>Ver equipos asignados:</b> En el menú lateral derecho "Consultas del tesista", el vínculo "Mis equipos asignados", redirige a la vista de los equipos asignados por el tutor de tesis
<b>Postcondiciones</b>	Ninguna

Tabla C.3: Caso de uso "Ver equipos asignados".

<b>Caso de uso</b>	tesista04
<b>Nombre</b>	Ver todos los informes de tesis
<b>Actor</b>	tesista, tutor de tesis
<b>Resumen</b>	Esta funcionalidad de software permite al usuario ver todos los avances de tesis que ha creado
<b>Otros casos de uso involucrados</b>	tesista01
<b>Precondiciones</b>	Para que la vista arroje valores, en el sistema deben haber datos de tipo de contenido "avance de tesis"
<b>Descripción</b>	1) <b>Si no hay avance de tesis crearlos:</b> Ver el flujo de creación de avances de tesis en tesista01. 2) <b>Ver avance de tesis:</b> En el menú lateral derecho "Consultas del tesista", el vínculo "Todos mis informes", redirige a la vista de todos los avances de tesis realizados hasta el momento.
<b>Postcondiciones</b>	Ninguna.

Tabla C.4: Caso de uso "Ver todos los informes de tesis".

<b>Caso de uso</b>	tesista05
<b>Nombre</b>	Ver los informes revisados por el tutor
<b>Actor</b>	tesista, tutor de tesis
<b>Resumen</b>	Esta funcionalidad de software permite al usuario ver todos los avances de tesis que el tutor de tesis ha revisado
<b>Otros casos de uso involucrados</b>	tesista01
<b>Precondiciones</b>	Para que la vista arroje valores, en el sistema deben haber datos de tipo de contenido "avance de tesis", donde su opción "revisado por el tutor", halla sido seleccionada.
<b>Descripción</b>	1) <b>Si no hay avance de tesis crearlos:</b> Ver el flujo de creación de avances de tesis en tesista01. 2) <b>Editar avance de tesis creada:</b> Solamente el usuario con el rol de tutor de tesis puede chequear la opción "revisado por el tutor", que el tesista creó. 3) <b>Ver avance de tesis:</b> En el menú lateral derecho "Consultas del tesista", el vínculo "Informes revisados por el tutor", redirige a la vista de todos los avances de tesis que el tutor haya revisado.
<b>Postcondiciones</b>	Ninguna.

Tabla C.5: Caso de uso "Ver los informes revisados por el tutor".

<b>Caso de uso</b>	tesista06
<b>Nombre</b>	Ver los informes sin revisar por el tutor
<b>Actor</b>	tesista, tutor de tesis
<b>Resumen</b>	Esta funcionalidad de software permite al usuario ver todos los avances de tesis que el tutor de tesis aún no ha revisado
<b>Otros casos de uso involucrados</b>	tesista01
<b>Precondiciones</b>	Para que la vista arroje valores, en el sistema deben haber datos de tipo de contenido "avance de tesis", donde su opción "necesita revisión", halla sido seleccionada.
<b>Descripción</b>	1) <b>Si no hay avance de tesis crearlos:</b> Ver el flujo de creación de avances de tesis en tesista01, añadiendo el chequeo de la opción "necesita revisión". 2) <b>Ver avance de tesis:</b> En el menú lateral derecho "Consultas del tesista", el vínculo "Informes sin revisar por el tutor", redirige a la vista de todos los avances de tesis que el tutor aún no ha revisado.
<b>Postcondiciones</b>	Ninguna.

Tabla C.6: Caso de uso "Ver los informes sin revisar por el tutor".

UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras de Software Basado en el Proceso Unificado

<b>Caso de uso</b>	tesista07
<b>Nombre</b>	Aceptar solicitud de relación del tutor de tesis
<b>Actor</b>	tesista, tutor de tesis
<b>Resumen</b>	Esta funcionalidad de software permitirá al usuario crear, ver, actualizar y eliminar avances de tesis en donde podrán ser enviadas a su respectivo tutor el cual enviará sugerencias y guías para continuar con su desarrollo
<b>Otros casos de uso involucrados</b>	tutor05,
<b>Precondiciones</b>	Ninguna
<b>Descripción</b>	<p>1) <b>Enviar solicitud de relación desde el tutor de tesis al tesista:</b> ver tutor05</p> <p>2) <b>Ver solicitudes de relación:</b> Cuando ingresa a su cuenta de usuario, luego del inicio de sesión aparecerá una notificación donde se informa de la solicitud de relación del tutor de tesis</p> <p>2.1) en el contenido de la página se ilustra quien ha enviado la solicitud, dar click en "Aprobar" para aceptar la solicitud o "Rechazar" para eliminarla.</p> <p><b>Flujo alternativo:</b></p> <p>2.1) En la parte superior izquierda puede acceder a la misma pantalla a través del link "Mis relaciones", haciendo click sobre la pestaña "Solicitudes recibidas", se mostrará el mismo menú con las opciones "Aprobar", y "Rechazar".</p>
<b>Postcondiciones</b>	El vínculo tutor de tesis y tesista quedará formalizado en el sistema.

Tabla C.7: Caso de uso "Aceptar solicitud de relación del tutor de tesis".

<b>Caso de uso</b>	tesista08
<b>Nombre</b>	Iniciar Sesión
<b>Actor</b>	tesista, tutor de tesis
<b>Resumen</b>	Como proceso de autenticación el sistema ofrece una funcionalidad llamada inicio de sesión que valida que el ingreso de un usuario al sistema sea legítimo
<b>Otros casos de uso involucrados</b>	tutor04
<b>Precondiciones</b>	Que el usuario que desee ingresar se encuentre en la base de datos del sistema.
<b>Descripción</b>	<p>1) <b>Si el usuario no existe, entonces crearlo:</b> ver tutor04</p> <p>2) <b>Iniciar sesión:</b> El usuario podrá iniciar sesión utilizando el nombre de usuario y la contraseña. Estas credenciales son otorgadas por aquella persona quien creó dicha cuenta.</p>
<b>Postcondiciones</b>	Ninguna.

Tabla C.8: Caso de uso "Iniciar Sesión".

C.2.2. Funcionalidades sobre el rol de tutor de tesis

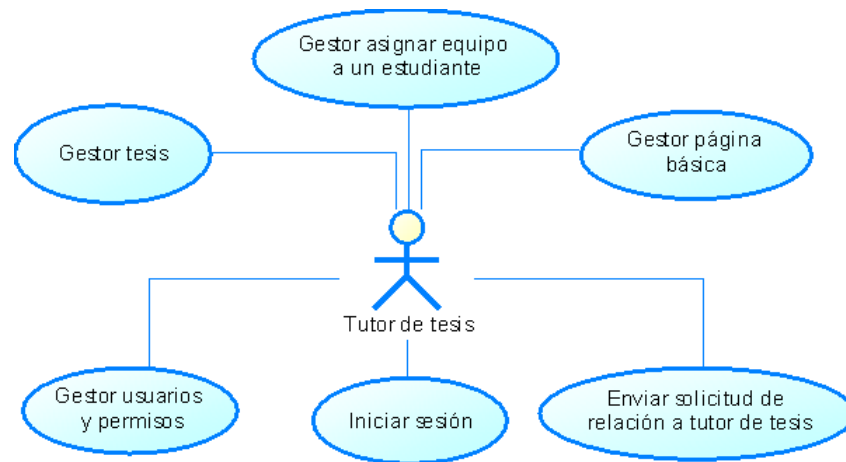


Figura C.2: Modelo de casos de uso, funcionalidades del tutor de tesis

UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras  
de Software Basado en el Proceso Unificado

---

<b>Caso de uso</b>	tutor01
<b>Nombre</b>	Gestor tesis
<b>Actor</b>	tutor de tesis, tesista (solo en edición)
<b>Resumen</b>	Esta funcionalidad de software permitirá al usuario crear, ver, actualizar y eliminar tesis que son ilustradas como punto de referencia para otros usuarios de la herramienta que deben realizar su propia tesis.
<b>Otros casos de uso involucrados</b>	Ninguno
<b>Precondiciones</b>	Ninguna
<b>Descripción</b>	<p><b>1) Crear tesis:</b></p> <p>1.1) El tutor de tesis accede desde el menú lateral izquierdo "Navegación", haciendo click en agregar contenido.</p> <p>1.2) El tutor selecciona de la lista, el tipo de contenido "tesis".</p> <p>1.3) El usuario ingresa los campos de la tesis, como título, tesista 1, tesista 2, descripción, campos para subir archivos como el anteproyecto, la resolución de aprobación, la monografía, el artículo, las diapositivas y 2 campos de chequeo ("hacer pública", que hace que la tesis aparezca en la vista de tesis publicadas por el grupo y "Ha sido sustentada").</p> <p>1.4) El usuario da click en aceptar para guardar la tesis o cancelar para descartarla.</p> <p><b>2) Ver tesis:</b> Desde el menú lateral derecho "Consultas del tutor de tesis", haciendo click en los hipervinculos "Tesis que yo he creado"(ilustra las tesis propias que el tutor de tesis ha subido) , y "Tesis del grupo publicadas"(ilustra todas las tesis que el tutor ha hecho publicas con el botón de chequeo "Hacer publica") se puede acceder a la función ver tesis haciendo click en el nombre de alguna tesis que estén en estas 2 vistas.</p> <p><b>3) Modificar y eliminar una tesis:</b></p> <p>3.1) Realizar el mismo flujo de ejecución de ver tesis de esta descripción.</p> <p>3.2) Dando click en el título de alguna tesis (si hay en la vista) se accede a la respectiva tesis.</p> <p>3.3) Dar click en la pestaña editar en la parte superior de la de tesis.</p> <p>3.4) Modificar los atributos permitidos. Para el tesista están permitidos solamente los campos de subida de archivos (ver sección 1.3 de este flujo de ejecución de caso de uso).</p> <p>3.5) Dar click en guardar para salvar modificaciones y en eliminar para borrar el informe.</p>
<b>Postcondiciones</b>	Ninguna

Tabla C.9: Caso de uso "Gestor tesis".

<b>Caso de uso</b>	tutor02
<b>Nombre</b>	Gestor asignar un equipo a un estudiante
<b>Actor</b>	tutor de tesis
<b>Resumen</b>	Esta funcionalidad de software permitirá al usuario crear, ver, actualizar y eliminar asignaciones de equipos a tesisistas las cuales brindan información de las responsabilidades de los estudiantes sobre diferentes implementos al interior de la institución.
<b>Otros casos de uso involucrados</b>	Ninguno
<b>Precondiciones</b>	Ninguna
<b>Descripción</b>	<p><b>1) Crear asignación de equipo a un estudiante:</b></p> <p>1.1) El tutor de tesis accede desde el menú lateral izquierdo "Navegación", haciendo click en agregar contenido.</p> <p>1.2) El tutor selecciona de la lista, el tipo de contenido "Asignar equipo a un estudiante".</p> <p>1.3) El usuario ingresa los campos de este tipo de contenido, como título y descripción.</p> <p>1.4) El usuario da click en aceptar para guardar la tesis o cancelar para descartarla.</p> <p><b>2) Ver asignación de equipo a un estudiante:</b> Desde el menú lateral derecho "Consultas del tutor de tesis", haciendo click en el hipervinculo "Equipos asignados"se puede acceder a la función ver tesis haciendo click en el título de algún informe que estén en esta vista.</p> <p><b>3) Modificar y eliminar una asignación de equipo a un estudiante:</b></p> <p>3.1) Realizar el mismo flujo de ejecución de ver asignación de equipo a un estudiante de esta descripción.</p> <p>3.2) Dando click en el título de algún resultado (si hay en la vista) se accede a la respectiva asignación.</p> <p>3.3) Dar click en la pestaña editar en la parte superior de la asignación de equipo.</p> <p>3.4) Modificar los atributos permitidos.</p> <p>3.5) Dar click en guardar para salvar modificaciones y en eliminar para borrar el informe.</p>
<b>Postcondiciones</b>	Ninguna

Tabla C.10: Caso de uso "Gestor asignar un equipo a un estudiante".

La figura C.2 ilustra las funciones que un usuario con el rol de tutor de tesis, puede llevar a cabo en la herramienta de soporte. Las siguientes secciones ilustran las funcionalidades de cada caso de uso asociado a este rol:

**Caso de uso "Gestor tesis":** Ver tabla C.9.

**Caso de uso "Gestor asignar un equipo a un estudiante":** Ver tabla C.10.

**Caso de uso "Gestor página básica":** Ver tabla C.11.

**Caso de uso "Gestor usuarios y permisos":** Ver tabla C.12.

**Caso de uso "Enviar solicitud de relación de tutor de tesis":** Ver tabla C.13.

## UP-VSE: Un Proceso de Referencia para las Pequeñas Entidades Desarrolladoras de Software Basado en el Proceso Unificado

<b>Caso de uso</b>	tutor03
<b>Nombre</b>	Gestor página básica
<b>Actor</b>	tutor de tesis
<b>Resumen</b>	La página básica es una funcionalidad propia del núcleo de Drupal y consiste en la creación de contenido a modo de información el cual pueda ser mostrado en la página, bien sea como parte de la portada o de una subsección de la página web.
<b>Otros casos de uso involucrados</b>	Ninguno
<b>Precondiciones</b>	Ninguna
<b>Descripción</b>	<p><b>1) Crear una página básica:</b></p> <p>1.1) El tutor de tesis accede desde el menú lateral izquierdo "Navegación", haciendo click en agregar contenido.  1.2) El tutor selecciona de la lista, el tipo de contenido "página básica".  1.3) El usuario ingresa los campos de este tipo de contenido, como título y descripción.  1.4) El usuario puede promover la página basándose sobre un árbol de navegación tras chequear la opción "Proporcionar un enlace de menú", en donde se puede indicar el elemento (página web) padre al cual va a pertenecer esta página básica proporcionando también un "título del enlace del menú", y una "descripción".  1.5) El usuario da click en aceptar para guardar la página básica o cancelar para descartarla.</p> <p><b>2) Ver página básica:</b> Para ver la página básica creada, primero se accede al elemento (página web) padre que ha sido asignado a ésta y por último se hace click en el link que tiene por nombre el "título del enlace de menú".</p> <p><b>3) Modificar y eliminar una página web:</b></p> <p>3.1) En la barra menú superior de la página dar click en la pestaña "Contenido".  3.2) Se ilustrará una lista con todos los contenidos que posee la página, buscar el "título del enlace de menú" de la página creada y en la última columna "Operaciones", aparecen las opciones de "editar", y "eliminar".  3.3) Dar click en "editar" para modificar los atributos permitidos (ver flujo de ejecución 1.3 del presente caso de uso).  3.4) Dar click en "eliminar" para borrar la página web del sistema.</p>
<b>Postcondiciones</b>	Ninguna

Tabla C.11: Caso de uso "Gestor página básica".

<b>Caso de uso</b>	tutor04
<b>Nombre</b>	Gestor usuarios y permisos
<b>Actor</b>	tutor de tesis
<b>Resumen</b>	La funcionalidad de usuarios y permisos concede al usuario acceso a la creación, modificación, contenido y eliminación de usuarios y sus permisos que tengan sobre la herramienta por rol.
<b>Otros casos de uso involucrados</b>	Ninguno
<b>Precondiciones</b>	Ninguna
<b>Descripción</b>	<p><b>1) Crear un usuario:</b></p> <p>1.1) En la barra menú superior de la página se coloca el cursor en la pestaña "Personas", luego descender el cursor hasta posicionarlo en la subsección "Añadir personas", y dar click.  1.2) El tutor llena los campos del formulario como lo son "Nombre de usuario", "Dirección de correo electrónico", "contraseña", "confirmar contraseña", "Estado (bloqueado o activo)", "Roles (usuario autenticado, administrador, tutor de tesis y/o tesista)", "Notificar al usuario acerca de su nueva cuenta", y el "idioma", para fines de notificación por correo electrónico (inglés y español).  1.3) Click en crear cuenta para crearla.</p> <p><b>2) Ver Usuario:</b> 2.1) En la barra menú superior de la página dar click en la pestaña "Personas" la cual desplegará una lista de usuarios. 2.2) Dar click en el usuario que desea consultar, esto mostrará su información-</p> <p><b>3) Modificar y eliminar un usuario:</b></p> <p>3.1) Repetir los mismos pasos de "2) Ver Usuario" de este caso de uso.  3.2) Dar click en la pestaña "editar" para modificar los campos del usuario.  3.3) Dar click en "cancelar cuenta", para eliminarla o en "guardar", para conservar los cambios.</p> <p><b>4) Modificar permisos por rol:</b></p> <p>1.1) En la barra menú superior de la página se coloca el cursor en la pestaña "Personas", luego descender el cursor hasta posicionarlo en la subsección "Permisos", y dar click. Se ilustrará una lista con los permisos por rol  1.2) Para editar, dar click en las cajas de chequeo hasta lograr la configuración por roles deseada.  1.3) Dar click en "Guardar permisos", para conservar la configuración.</p>
<b>Postcondiciones</b>	Ninguna

Tabla C.12: Caso de uso "Gestor usuarios y permisos".

<b>Caso de uso</b>	tutor05
<b>Nombre</b>	Enviar solicitud de relación de tutor de tesis
<b>Actor</b>	tutor de tesis
<b>Resumen</b>	Esta funcionalidad permite realizar una vinculación entre tutor de tesis y estudiantes para permitir un manejo más adecuado sobre las consultas.
<b>Otros casos de uso involucrados</b>	Ninguno
<b>Precondiciones</b>	Ninguna
<b>Descripción</b>	<p>1) Para establecer el vínculo tutor de tesis - tesista, primero se deben consultar todos los tesistas haciendo click en el vínculo "Tesistas", del menú lateral derecho "Consultas del tutor de tesis".  2) Luego dar click en el tesista al que se le va a enviar la solicitud.  3) Dar click en el vínculo "Convertirse en tutor de tesis de &lt; nombre_de_usuario_seleccionado &gt;".  4) Se puede llenar el campo "Comentarios", opcionalmente.  5) Click en el botón "Enviar".</p>
<b>Postcondiciones</b>	Para concluir el vínculo tutor de tesis - tesista realizar el caso de uso tesista07

Tabla C.13: Caso de uso "Enviar solicitud de relación de tutor de tesis".

### C.3. Capturas de pantalla, prototipos funcionales

Actualmente la herramienta ha culminado la fase de producción y se encuentra en etapas intermedias de despliegue donde se pretende que su funcionamiento esté albergado en los servidores de la Universidad del Cauca como extensión de su url<sup>1</sup>, actualmente la herramienta se encuentra montada en un servidor gratuito debido a protocolos de seguridad y de accesibilidad inmediata de la herramienta<sup>2</sup> y también en la intranet de la Universidad del Cauca<sup>3</sup>.

A continuación se ilustran algunas figuras que ilustran parte del funcionamiento de la herramienta:

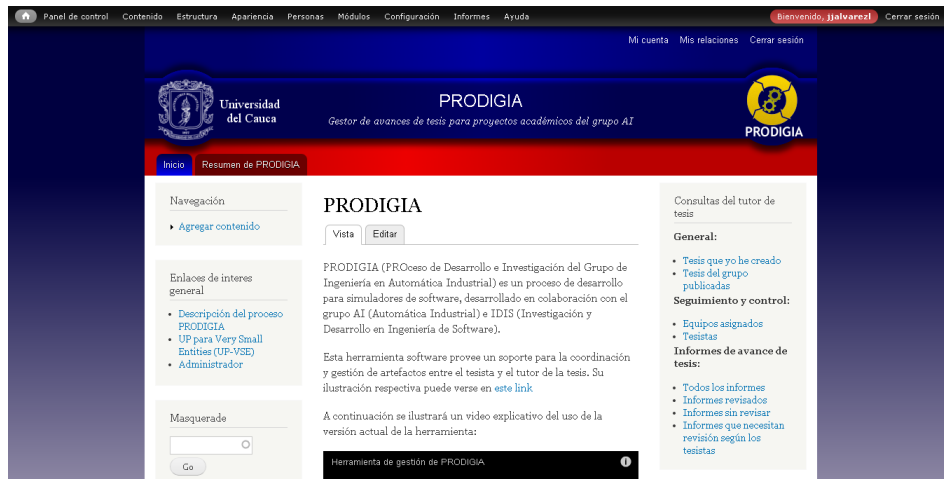


Figura C.3: Página principal de la herramienta de soporte

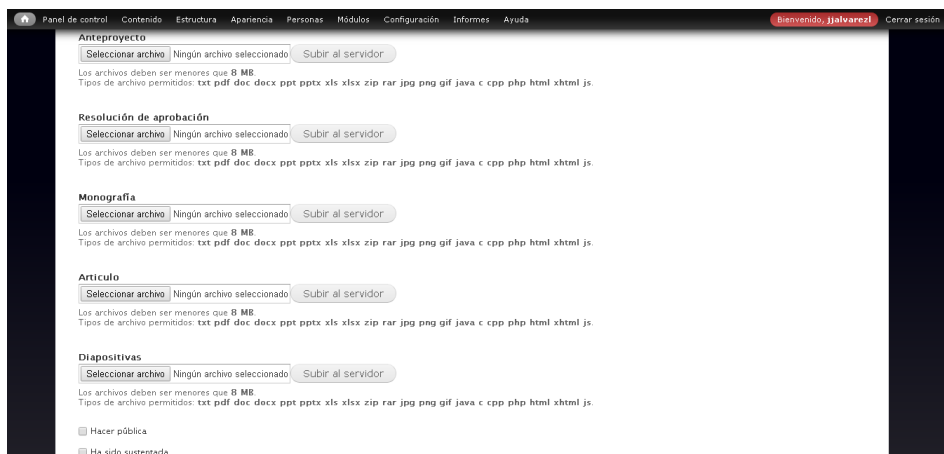


Figura C.4: Funcionalidad de "Agregar tesis" de la herramienta de soporte.

<sup>1</sup><http://www.unicauca.edu.co/prodigia>

<sup>2</sup><http://jhonalvarez.hol.es/PRODIGIA/gestor/>

<sup>3</sup><http://facultades.unicauca.edu.co/prodigia>

## Anexo D

# Minutas de reunión

Las minutas de reunión conservan los aspectos mas importantes de cada reunión realizada en el estudio de caso, a continuación se mostrarán los informes de mayor relevancia:

### D.1. Minuta de reunión (09-04-2013):

1. Datos principales:

- **Descripción:** Planeación del proyecto
- **Lugar:** Salón de reuniones del IPET
- **Hora de inicio:** 14:00
- **Hora de termino:** 16:00

2. Participantes:

- **Andres Vivas:** Director del Grupo AI
- **Jhon Jairo Alvarez:** Tesista del proyecto UP-VSE (Grupo IDIS)

3. Temas:

- 3.1. Exposición del proceso unificado y de los artefactos para las primeras iteraciones del proyecto.
- 3.2. Selección del equipo de calidad del proyecto, y discusión sobre cómo se manejará el proceso.

4. Compromisos:

- Realizar una reunión para comenzar a definir el proceso para lo cual hay que definir el proceso para la actividad de ingeniería de requerimientos. Plazo de realización 09-04-2013. Responsable: Jhon Jairo Alvarez.

5. Conclusiones:

- Se realizó la presentación del UP como una herramienta para eliminar la falta de documentación y de orden de artefactos software.
- En la siguiente reunión se expondrá el flujo que propone el UP para la ingeniería de requerimientos a los integrantes del grupo de calidad

## D.2. Minuta de reunión (28-04-2013):

1. Datos principales:

- **Descripción:** Realización del proceso de ingeniería de requisitos
- **Lugar:** Salón 116 del IPET
- **Hora de inicio:** 11:00
- **Hora de termino:** 12:30

2. Participantes:

- **Diego Bravo:** Miembro del equipo de calidad (Grupo AI)
- **Luis Daladier:** Miembro del equipo de calidad (Grupo AI)
- **Didier Vera:** Miembro del equipo de calidad (Grupo AI)
- **Jhon Jairo Alvarez:** Tesista del proyecto UP-VSE (Grupo IDIS)

3. Temas:

- 3.1. Exposición de la captura de requerimientos como disciplina del proceso unificado.
- 3.2. Diseño del proceso para la obtención de los requerimientos: No hubo opinión de los participantes.
- 3.3. Definición de roles, artefactos y responsables junto con fechas de entrega: Diego Bravo opinó que aún no definamos las fechas de entrega del resto de artefactos ni del resto de roles hasta no tener bien en claro los artefactos sobre casos de uso.
- 3.4. Muestra de herramientas CASE.

4. Compromisos:

- Realizar la entrevista con el cliente y la visión del producto. Plazo de realización 02-05-2013. Responsable: Luis Daladier y Didier Vera.



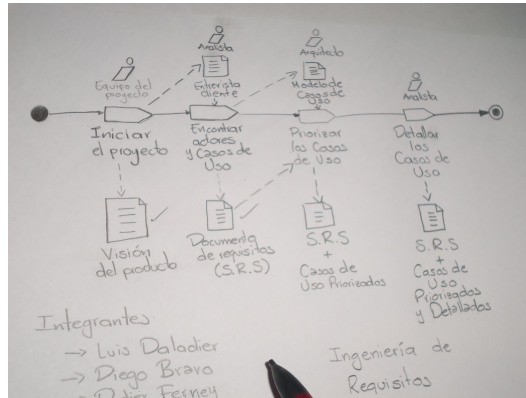


Figura D.1: Proceso acordado sobre la captura de requisitos en el grupo IA.

- Estudiar documentación de casos de uso. Plazo de realización 02-05-2013. Responsable: Luis Daladier.
- Enviar la presentación de ingeniería de requisitos. Plazo de realización: 29-05-2013. Responsable: Jhon Jairo Alvarez.
- Hablar sobre el equipo de proyecto, definiendo integrantes y fecha de capacitación. Plazo de realización: 02-05-2013. Responsable: Jhon Jairo Alvarez.
- Hablar sobre el repositorio de trabajo (SVN a utilizar), Plazo de realización: 02-05-2013. Responsable: Jhon Jairo Alvarez.
- Hablar sobre la gestión del riesgo. Plazo de realización: 02-05-2013. Responsable: Jhon Jairo Alvarez.

#### 5. Conclusiones:

- Ha sido diseñado el proceso que se seguirá para la ingeniería de requisitos en el proyecto basandose en el Proceso Unificado (UP) de I. Jacobson et al [2].

#### 6. Anexos:

- La figura D.1 ilustra el proceso acordado.

### D.3. Minuta de reunión (08-10-2013):

#### 1. Datos principales:

- **Descripción:** Socialización del proceso PRODIGIA con ingenieros del grupo AI.
- **Lugar:** Salón de reuniones del IPET
- **Hora de inicio:** 13:00



Figura D.2: Proceso acordado sobre la captura de requisitos en el grupo IA.

- **Hora de termino:** 16:00

2. Participantes:

- **Ocho ingenieros del grupo AI**
- **Julio Ariel Hurtado:** Tutor de tesis del proyecto UP-VSE (Grupo IDIS).
- **Jhon Jairo Alvarez Londoño:** tesista del proyecto UP-VSE (Grupo IDIS).

3. Temas:

- 3.1. Exposición del proceso PRODIGIA y de los adelantos que se llevaban hasta la fecha.
- 3.2. Exposición de la estrategia de trabajo del proceso PRODIGIA.
- 3.3. Localización web sobre la documentación formal del proceso.

4. Compromisos:

- Comunicarse con los interesados del proyecto. Responsable: Jhon Jairo Alvarez.

5. Conclusiones:

- Los ingenieros integrantes del grupo AI demostraron interés y apoyo al proyecto al ver que podría ser una buena solución para solventar los problemas de documentación.

6. Anexos:

- La figura [D.2](#) ilustra una sección de la reunión.

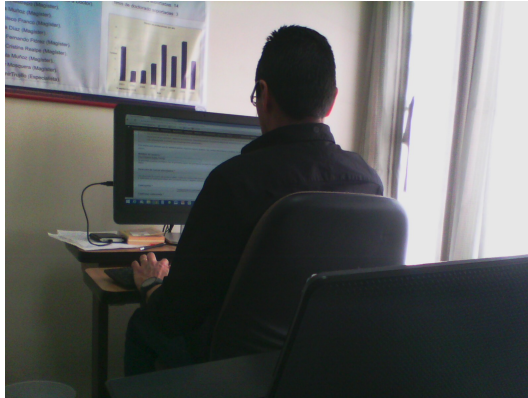


Figura D.3: Reuniones para validación y verificación de la herramienta PRODIGIA.

#### **D.4. Minuta de reunión (02-05-2014):**

1. Datos principales:

- **Descripción:** Reunión para corregir la herramienta de soporte de PRODIGIA.
- **Lugar:** Salón 428
- **Hora de inicio:** 10:00
- **Hora de termino:** 10:30

2. Participantes:

- **Andres Vivas:** Director del grupo AI.
- **Jhon Jairo Alvarez Londoño:** tesista del proyecto UP-VSE (Grupo IDIS).

3. Temas:

- 3.1. Mejora de las consultas de la herramienta de soporte del proceso PRODIGIA.
- 3.2. Eliminación de mensajes enviados automáticamente por el sitio.
- 3.3. Definición de la ejecución de las encuestas.

4. Compromisos:

- Mejorar las encuestas realizadas. Plazo: 02-05-2014. Responsable: Jhon Jairo Alvarez Londoño.
- Hablar en contacto 55 (infraestructura de servidores de la Universidad del Cauca) para que permitan subir la aplicación como parte de la extensión de la página de la institución. Plazo: 02-05-2014. Responsable: Andres Vivas.

5. Conclusiones:

- El software está listo para comenzar las primeras pruebas reales.



Figura D.4: Socialización de la herramienta de soporte de PRODIGIA.

- Las encuestas deben ser realizadas con las personas que ejecuten el proceso.

6. Anexos:

- La figura [D.3](#) ilustra una sección de la reunión.

## D.5. Minuta de reunión (14-05-2014):

1. Datos principales:

- **Descripción:** Reunión de socialización del proceso PRODIGIA en conjunto con su herramienta de soporte.
- **Lugar:** Salón 313
- **Hora de inicio:** 17:00
- **Hora de termino:** 16:30

2. Participantes:

- **Andres Vivas:** Director del Grupo AI.
- **Doce ingenieros del grupo AI**
- **Jhon Jairo Alvarez Londoño:** tesista del proyecto UP-VSE (Grupo IDIS).

3. Temas:

- 3.1. Exponer parte del proceso PRODIGIA.
- 3.2. Exponer la herramienta de soporte del proceso PRODIGIA.
- 3.3. Dar contacto de asesoría y soporte de la herramienta.

4. Compromisos:

- Subir la herramienta al servidor de Unicauca. Responsable: Jhon Jairo Alvarez Londoño.
- Concretar los campos de IP y MAC del computador sobre el cual se subirá la herramienta de soporte de PRODIGIA Responsable: Andres Vivas, Jhon Jairo Alvarez.

5. Conclusiones:

- El software está listo para comenzar las primeras pruebas reales.
- Las encuestas deben ser realizadas con las personas que ejecuten el proceso.

6. Anexos:

- La figura [D.4](#) ilustra una sección de la reunión.