



## **UNIVERSIDAD TECNOLÓGICA DE LA MIXTECA**

“Metamodelo para la definición e implantación de los procesos de planificación y control de proyectos en pequeños equipos de desarrollo”

### **TESIS**

**PARA OBTENER EL GRADO DE  
MAESTRA EN TECNOLOGÍAS DE CÓMPUTO APLICADO**

### **PRESENTA**

**ING. NEIRA SÁNCHEZ ROJAS**

**DIRECTOR DE TESIS  
DR. IVÁN ANTONIO GARCÍA PACHECO**

**HUAJUAPAN DE LEÓN, OAXACA. ENERO DE 2016.**



## **Dedicatoria**

A mis padres:

Con amor y cariño les dedico todo mi esfuerzo, gracias por enseñarme a seguir mis sueños y enseñarme que siempre que se quiere se puede.

A mi hijo

Por ser el motor que me impulsa a mejorar día a día, con mucho amor te dedico este trabajo.

A mi esposo:

Por apoyarme en cada momento y por todo el amor y tiempo dedicado para que yo concluyera esta tesis.



## **Agradecimientos**

A mi asesor, Dr. Iván Antonio García Pacheco, por el tiempo, esfuerzo dedicado para leer este trabajo, también por los consejos y apoyo brindado en todo momento para que esto fuera posible, he aprendido muchas cosas. Gracias por enseñarme a ser disciplinada.

A mis sinodales M. C. Everth Haydeé Rocha Trejo, M. C. Ricardo Ruíz Ruíz, Dra. Carla Leninca Pacheco Agüero, y Dr. Carlos Alberto Fernández y Fernández, por todo el tiempo dedicado a la lectura de esta tesis y por las observaciones hechas que ayudaron a mejorar este trabajo.

A mis padres Josefa y Juan por su apoyo incondicional, por sus consejos siempre tan sabios, por haberme apoyado siempre hasta el final y ayudarme a que todo esto fuera posible.

A mi hijo, Joseph Samuel, y esposo, Marcel, por apoyarme en cada momento durante esta etapa de mi vida, sin ustedes no lo hubiera logrado.

A mis suegros Celestina y Jerónimo por todo el apoyo brindado, especialmente a mi suegra por cuidar a su adorable nieto durante en esta etapa de mi vida y por apoyarnos siempre.

A mis hermanas Magdalena, Hassel, y Dalai y hermanos Miguel, Jesús y Johan, por alentarme a seguir hasta el final y por confiar en que terminaría exitosamente esta etapa de mi vida.

Finalmente, a todos aquellos familiares y amigos que me apoyaron para escribir y concluir esta tesis, gracias.



## Índice

Índice .....	vii
Lista de tablas .....	ix
Lista de figuras .....	xi
Resumen .....	xiii
1. Introducción y motivaciones .....	1
1.1. Importancia del problema y necesidad de la solución .....	4
1.2. Hipótesis de la tesis .....	10
1.3. Objetivos de la tesis .....	10
1.3.1. Objetivo general .....	10
1.3.2. Objetivos específicos .....	10
1.4. Delimitaciones y limitaciones de la tesis .....	11
1.5. Metodología de trabajo .....	11
1.6. Aproximación a la solución .....	13
1.7. Estructura de la tesis .....	16
1.8. Publicaciones generadas .....	17
2. Marco teórico .....	19
2.1. Antecedentes .....	19
2.2. Un poco de historia sobre la gestión en la industria de software .....	22
2.3. La gestión de los proyectos de software .....	26
2.3.1. La planificación de los proyectos .....	28
2.3.2. La monitorización y control de los proyectos .....	30
2.4. Estudio del arte .....	32
2.4.1. Un programa de mejora de la gestión de proyectos de acuerdo al estándar ISO/IEC 29110 y PMBoK® .....	33
2.4.1.1. <i>Objetivo</i> .....	33
2.4.1.2. <i>Descripción</i> .....	33
2.4.1.3. <i>Resultados</i> .....	37
2.4.2. Gestión cuantitativa de proyectos en pequeñas y medianas empresas de software .....	39
2.4.2.1. <i>Objetivo</i> .....	39
2.4.2.2. <i>Descripción</i> .....	39
2.4.2.3. <i>Resultados</i> .....	42
2.4.3. Definición de prácticas para la planificación de proyectos en pequeñas y medianas empresas en base a sistemas de planificación de recursos .....	43
2.4.3.1. <i>Objetivo</i> .....	43
2.4.3.2. <i>Descripción</i> .....	44
2.4.3.3. <i>Resultados</i> .....	47

2.4.4. Gestión de proyectos de software en pequeñas y micro empresas: un enfoque de integración de ISO/IEC 12207, ISO/IEC 29110 y PMBoK .....	49
2.4.4.1. <i>Objetivo</i> .....	49
2.4.4.2. <i>Descripción</i> .....	49
2.4.4.3. <i>Resultados</i> .....	55
2.4.5. Un marco de trabajo para la gestión ágil de proyectos en PyMES.....	57
2.4.5.1. <i>Objetivo</i> .....	57
2.4.5.2. <i>Descripción</i> .....	57
2.4.5.3. <i>Resultados</i> .....	59
2.4.6. Gestión de proyectos de software en microempresas usando ISO/IEC 29110.....	62
2.4.6.1. <i>Objetivo</i> .....	62
2.4.6.2. <i>Descripción</i> .....	62
2.4.6.3. <i>Resultados</i> .....	66
2.5. Análisis comparativo de propuestas analizadas.....	68
2.6. Conclusiones finales del capítulo .....	72
3. Metamodelo de gestión para los pequeños equipos.....	75
3.1. Los metamodelos en la Ingeniería de Software .....	75
3.2. Determinar el ámbito del metamodelo .....	78
3.3. Definir y/o seleccionar el lenguaje de representación y creación de objetos .....	94
3.4. Creación del metamodelo y el escenario inicial .....	96
3.5. Probar el metamodelo .....	102
3.5.1. Incorporación de la evaluación.....	105
3.5.2. Implementación del concepto de patrón a través de la PAL .....	108
4. Resultados obtenidos .....	113
4.1. Contexto de la validación .....	113
4.2. Definición del proyecto línea base del caso de estudio .....	114
4.2.1. Identificación de la cobertura de los procesos.....	116
4.2.2. Implantación del metamodelo en el proyecto piloto.....	118
4.2.3. Comparación final con la línea base inicial.....	122
4.2.4. Análisis detallado sobre los resultados del proyecto piloto.....	123
4.2.5. Conclusiones del caso de estudio .....	124
5. Conclusiones.....	127
6. Anexo A.- Acrónimos.....	131
7. Anexo B.- Cuestionarios de evaluación para la experimentación .....	133
8. Anexo C.- Activos para los procesos de DMS y APE.....	157
9. Anexo D.- Acta de publicaciones .....	255
10. Referencias bibliográficas .....	259
10.1. Sitios de internet .....	266

## Lista de tablas

Tabla 1. La importancia de la gestión del proyecto en las causas de los problemas en los proyectos (Brooks, 2012).....	8
Tabla 2. Activos de la planificación del proyecto (Mesquida & Mas, 2014).....	35
Tabla 3. Activos de la ejecución del plan del proyecto (Mesquida & Mas, 2014).....	36
Tabla 4. Activos de la evaluación y control del proyecto (Mesquida & Mas, 2014).....	36
Tabla 5. Activos del cierre del proyecto (Mesquida & Mas, 2014). ....	37
Tabla 6. Perfil de las PyMES participantes (García, Pacheco, & Calvo, 2014).....	42
Tabla 7. Comparación de los resultados entre las evaluaciones (García, Pacheco, & Calvo-Manzano, 2014).....	43
Tabla 8. Información de mejora por empresa (García, Pacheco, & Calvo-Manzano, 2014). ....	43
Tabla 9. Dimensiones de medición de la planificación de proyectos (Salomo, Weise, & Gemunden, 2007). ....	44
Tabla 10. Dimensiones de medición de los proyectos exitosos (Mahaney & Lederer, 2006). ....	45
Tabla 11. Relación entre ISO/IEC 12207 e ISO/IEC 29110 (Mas & Mesquida, 2013).....	51
Tabla 12. Relación las tareas de GP.1 Planificación del Proyecto y PMBoK (Mas & Mesquida, 2013).....	53
Tabla 13. Relación las tareas de GP.2 Ejecución del Plan del Proyecto y PMBoK (Mas & Mesquida, 2013). ....	54
Tabla 14. Relación las tareas de GP.3 Evaluación y Control del Proyecto y PMBoK (Mas & Mesquida, 2013). ....	55
Tabla 15. Relación las tareas de GP.4 Cierre del Proyecto y PMBoK (Mas & Mesquida, 2013). .	55
Tabla 16. Herramientas y procedimientos usados por el marco de gestión ágil (O'Sheedy & Sankaran, 2013).....	59
Tabla 17. Asignación de características de la microempresa a un grupo de perfiles (O'Connor & Laporte, 2012). ....	63
Tabla 18. Perfil graduado del grupo de perfiles genérico (O'Connor & Laporte, 2012). ....	63
Tabla 19. Categorías para los proyectos de ingeniería (O'Connor & Laporte, 2012).....	67
Tabla 20. Asignación de roles a dos miembros de la microempresa (O'Connor & Laporte, 2012). ....	67
Tabla 21. Asignación de documentos a dos personas de la microempresa (O'Connor & Laporte, 2012).....	67
Tabla 22. Definición de criterios para realizar el análisis comparativo. ....	69
Tabla 23. Análisis comparativo de propuestas enfocadas a la gestión de proyectos.....	70
Tabla 24. Tipo de documentación de un proceso (Olson, 20006).....	80
Tabla 25. Mapeo de elementos de proceso entre los modelos/estándares que promueven la gestión. ....	83

Tabla 26. Clasificación de niveles de rendimiento durante la evaluación del proceso (García, Calvo-Manzano, Cuevas, & San Feliu, 2007). .....	105
Tabla 27. Número de empleados y actividad principal de la empresa incluida en el caso de estudio. ....	114
Tabla 28. Infraestructura de la empresa incluida en el caso de estudio.....	114
Tabla 29. Resumen de funcionalidades cubiertas por el proyecto escogido. ....	115
Tabla 30. Variable y valores objetivos del proyecto línea base. ....	116
Tabla 31. Escala normativa de calificación definida por el Organismo Certificador.....	118
Tabla 32. Objetivos a valorar con el proyecto piloto.....	122
Tabla 33. Incremento en la cobertura de APE y DMS. ....	122
Tabla 34. Análisis de resultados sobre los objetivos establecidos por PES1 para el proyecto piloto. ....	123

## Lista de figuras

Figura 1.1. Tendencia de publicaciones sobre SPI por año.....	2
Figura 1.2. Factor de éxito en proyectos con pequeños equipos (Standish Group International, 2013).....	6
Figura 1.3. Factor de éxito con equipos de cuatro a seis integrantes (Standish Group International, 2013).....	7
Figura 1.4. Resolución de proyectos en el 2012 (Standish Group International, 2013).....	8
Figura 1.5. Metodología propuesta para crear la solución de esta tesis. ....	12
Figura 1.6. Solución propuesta. ....	16
Figura 2.1. Evolución de la gestión de los proyectos de software. ....	22
Figura 2.2. Áreas básicas para la gestión de proyectos (CMMI, 2010). ....	28
Figura 2.3. Síntesis del proceso de planificación de proyectos (CMMI, 2010). ....	30
Figura 2.4. Síntesis del proceso de monitorización y control de proyectos (CMMI, 2010). ....	31
Figura 2.5. Capas de abstracción del marco conceptual de FQPMSE (García, Pacheco, & Calvo-Manzano, 2014).....	40
Figura 2.6. Representación de la PAL requerida para la implantación de FQPMSE (García, Pacheco, & Calvo-Manzano, 2014).....	41
Figura 2.7. Entorno de Ingeniería de Software de FQPMSE (García, Pacheco, & Calvo-Manzano, 2014).....	42
Figura 2.8. Principales resultados sobre el estudio de planificación en base a ERPs (Mahaney & Lederer, 2006). ....	48
Figura 2.9. Procesos de la norma ISO/IEC 12207 (Mas & Mesquida, 2013). ....	50
Figura 2.10. Procesos de gestión de proyectos de la norma ISO/IEC 29110-5-1-2 (Mas & Mesquida, 2013). ....	51
Figura 2.11. El marco de trabajo para la gestión ágil de los proyectos (O'Sheedy & Sankaran, 2013).....	58
Figura 2.12. Diagramas de procesos del perfil básico de ISO/IEC 29110 (O'Connor & Laporte, 2012).....	64
Figura 2.13. Contenido de un paquete de implementación de la gestión del proyecto (O'Connor & Laporte, 2012). ....	65
Figura 3.1. Actividades para desarrollar el metamodelo propuesto. ....	77
Figura 3.2. Dimensiones críticas del proceso dentro de una organización.....	78
Figura 3.3. Tipos de elementos para modelar/representar un proceso (Pino, Ruiz, Garcia, & Piattini, 2012). ....	79
Figura 3.4. Elementos básicos de un proceso para la gestión de proyectos en pequeñas empresas (Varkoi, 2010). ....	80
Figura 3.5. Tipos de documentación de procesos y sus relaciones (Olson, 2006). ....	81
Figura 3.6. Arquitectura de MDA propuesta por OMG (Henderson-Sellers, 2011). ....	96

---

Figura 3.7. Modelo conceptual propuesto. ....	99
Figura 3.8. Metamodelo propuesto para la definición de los procesos de planificación y control de proyectos. ....	100
Figura 3.9. El uso de patrones en las áreas de la ingeniería del software (García, Martín, Urbano, & de Amescua, 2013). ....	103
Figura 3.10. Definición del concepto patrón a implementar en el metamodelo. ....	104
Figura 3.11. Definición conceptual de una PAL (García, de Amescua, Sánchez, & Bermón, 2011). ....	104
Figura 3.12. Ejemplo de análisis de respuestas realizado por los jefes de proyectos. ....	106
Figura 3.13. Modelo de datos obtenido a partir del metamodelo. ....	109
Figura 3.14. Modelo de datos para la implementación del concepto de patrón. ....	110
Figura 4.1. Resultados de la evaluación sobre los procesos de PES1. ....	117
Figura 4.2. Pantalla de la creación del patrón Patrón PV/2013 para realizar el proyecto piloto. ..	119
Figura 4.3. Pantalla de la asociación del patrón creado con el proyecto piloto. ....	119
Figura 4.4. Diagrama de objetos del metamodelo. ....	120
Figura 4.5. Pantalla de la PAL para controlar el progreso del proyecto piloto. ....	121
Figura 4.6. Niveles de cobertura obtenidos al cierre del caso de estudio. ....	123

## **Resumen**

La certificación es importante para toda empresa desarrolladora de software porque le permite acceder a mercados internacionales que demandan el uso de normas de calidad. Sin embargo, las certificaciones son costosas en todos los sentidos y la mayoría de éstas están enfocadas a las grandes empresas. Una aproximación interesante ha planteado que las Pequeñas y Medianas Empresas desarrolladoras de Software (que predominan en la industria mundial) establezcan programas de Mejora del Proceso de Software, pero de igual manera la obtención de resultados se prevé a mediano y largo plazo, demasiado tiempo para este tipo de organizaciones, y además se asume el conocimiento técnico avanzado de la plantilla. Así pues, el presente proyecto de tesis plantea el desarrollo de un metamodelo orientado a los equipos pequeños que son formados dentro de las pequeñas empresas, de tal forma que les permita establecer de forma correcta los procesos de planificación y control de proyectos con el soporte de una herramienta computarizada que ayudará a implantar correctamente los componentes creados para dicho metamodelo. Por último, como estrategia de experimentación se plantea la definición de un caso de estudio en, al menos, una pequeña empresa, con el objetivo de mostrar evidencia cualitativa y cuantitativa de la utilidad del metamodelo propuesto.

### **Palabras clave**

Metamodelo, pequeños equipos, pequeños entornos, pequeñas empresas, planificación y control de proyectos, mejora del proceso de software.



## 1. Introducción y motivaciones

La certificación es importante para las empresas desarrolladoras de software porque permite acceder a mercados que demandan el uso de normas internacionales de calidad, con el objetivo de aumentar las oportunidades de negocio a través de la confianza que se genera en los clientes sobre la calidad de sus productos.

En este sentido, muchas organizaciones pequeñas enfocadas al desarrollo de software están intentando conseguir la certificación en un modelo de calidad, como la Norma Mexicana denominada NMX-I-006/02-NYCE-2004 (o MoProSoft) (Oktaba, 2006) o el CMMI-DEV v1.3 (*Capability Maturity Model Integration for Development v1.3*) con sus ajustes para pequeños entornos (CMMI, 2010), con el fin de acceder al mercado de las exportaciones. Sin embargo, la preparación previa a la certificación es larga y costosa cuando se trata de empresas pequeñas.

No obstante, en el contexto de las certificaciones que abren el abanico de oportunidades en el mercado internacional, los modelos propuestos por organizaciones como el Instituto de Ingeniería de Software (SEI, por sus siglas en inglés) de la Universidad de Carnegie Mellon o la Organización Internacional para la Estandarización (ISO, por sus siglas en inglés), están estructurados para ser utilizados por las grandes empresas que cuentan con una amplia experiencia técnica y el suficiente recurso económico y humano para lograrlo, y difícilmente pueden ser seguidos por empresas pequeñas dado que suponen una importante inversión en dinero y tiempo. Sin embargo, a partir de comienzos de los años noventa, la industria y los investigadores interesados en la Ingeniería de Software (IS) han expresado un interés especial en la Mejora del Proceso de Software (SPI, por sus siglas en inglés) (Pino, García, & Piattini, 2008). Una muestra de esto es la aparición de iniciativas internacionales relacionadas con SPI, tales como MoProSoft (NYCE, 2005), ISO/IEC 12207:2008 (ISO/IEC, 2008), CMMI-DEV v1.3 (CMMI 2010), e ISO/IEC 29110 (ISO/IEC, 2011). En este sentido, la Figura 1.1 muestra el impacto de SPI en el contexto internacional considerando el número de publicaciones generadas desde 1996 hasta la fecha (Calvo-Manzano, García, & Arcilla, 2008; García, Pacheco, & Calvo-Manzano, 2015).

En este sentido, es importante mencionar que a partir del 2005 y a través del *First International Research Workshop for Process Improvement in Small Settings*, organizado por el SEI, el interés sobre la definición de iniciativas de mejora en pequeñas empresas comenzó a incrementarse a nivel mundial (García, Graettinger, & Kost, 2006). Así, una solución alternativa que se ha propuesto para remediar los problemas anteriormente mencionados, es la implementación de modelos o programas de SPI. De acuerdo con Krasner (1997), SPI es un esfuerzo planeado, gestionado y controlado que tiene como objetivo mejorar la capacidad de una organización inmadura para desarrollar software. Es decir, se asume que una organización madura posee un alto nivel de capacidad en sus procesos. Además, SPI establece una eficiente gestión de los proyectos y aplica los fundamentos de ingeniería para el control cuantitativo de su(s) proceso(s) de software, el cual se convierte en la base para la mejora continua del proceso. Por último, la organización no solo asume la responsabilidad de ejecutar completamente sus compromisos enfocados al aumento de la

calidad del proceso y del producto, sino también reducir los costos y el tiempo, aumentar la posibilidad de reproducir los éxitos en los proyectos, controlar los riesgos en los procesos, y mejorar la confianza y satisfacción del cliente (Niazi, Babar, & Verner, 2010).

## Publicaciones de SPI por año

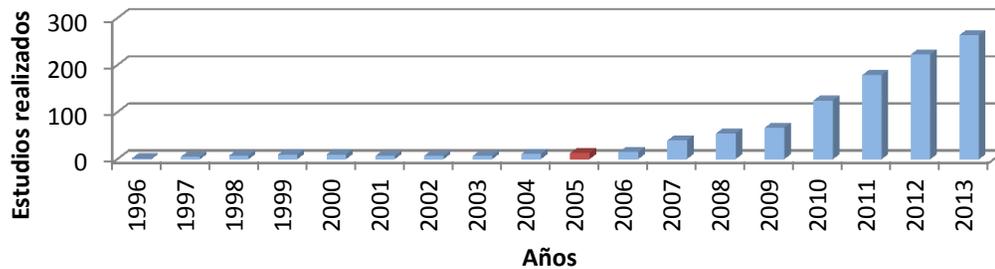


Figura 1.1. Tendencia de publicaciones sobre SPI por año.

Sin embargo, en (Ruíz, Piore, & Schrank, 2005) se argumenta que el mayor error en la implantación de programas de SPI en Pequeñas y Medianas Empresas de Software (PyMES) es la carencia de seguimiento de los planes de acción e implantación, debido fundamentalmente a que estas actividades son muy costosas de realizar en todos los sentidos, ya que consumen mucho tiempo y recursos. En este sentido, Niazi y Zahran (2013) han argumentado que un ciclo completo de SPI puede realizarse en un período de entre 18 y 24 meses; es decir, demasiado tiempo para las PyMES. Otro problema más, de acuerdo a la investigación de (Sosa, Blanc, Pralong, Álvarez, & Galáz, 2013), es que, aunque el Retorno de la Inversión (ROI, por sus siglas en inglés) de un programa de SPI está previsto que alcance entre 4 y 9 veces la inversión original, éste se produce a mediano-largo plazo, nuevamente demasiado tiempo para las PyMES.

Así pues, uno de los problemas cruciales en la industria mexicana de software (y en general) es que las empresas que desean mejorar las habilidades de su personal, el rendimiento de sus procesos para crear sus productos, y por ende la calidad de los productos que desarrollan, normalmente deben invertir una importante cantidad de recursos económicos para la capacitación de los trabajadores debido a las graves deficiencias de conocimiento. En un entorno real industrial, los ingenieros de software deben comunicarse efectivamente con los usuarios, clientes y expertos del dominio con el fin de capturar los requisitos correctos; estimar de manera eficiente los costos del proyecto y calendarizar para maximizar las ganancias; formular planes apropiados; supervisar y controlar los planes establecidos; y tomar medidas correctivas cuando sea necesario (García, Pacheco, & Calvo-Manzano, 2015). Sin embargo, esto es particularmente difícil de hacer en el contexto de los pequeños equipos, pequeños entornos, y pequeñas empresas. En este sentido, estos tres términos son importantes para entender nuestra industria, por lo cual es necesario hacer la distinción entre la terminología usada con el objetivo de situar adecuadamente el contenido de esta tesis:

- De acuerdo con Canós, Letelier, y Penadés (2003) y el SEI (CMMI, 2006), los *pequeños equipos* son aquellos que están compuestos por un máximo de 10 integrantes, y que además cuentan con una infraestructura y recursos limitados. Sin embargo, estos equipos pueden ser mucho más productivos que los grandes, dado que son integrados de manera rápida y presentan pocos problemas de comunicación.

- En el contexto de una iniciativa de mejora, el SEI argumenta que el término *pequeño entorno* ha sido definido para una compañía con menos de 100 personas, una organización con menos de 50 personas, y proyectos de menos de 20 personas (CMMI, 2006).
- Por último, en (Hogan, Smith, & Thomas, 2002) se afirmó que una *pequeña empresa* es una de las categorías otorgadas a las empresas de acuerdo a su región y al número de trabajadores que tiene; en México y principalmente en Latinoamérica este tipo de empresa tiene de 10 a 50 empleados (MIET, 2014).

En este sentido, el problema es que, dentro de las actividades de la gestión de software en pequeños equipos, pequeños entornos de desarrollo, y pequeñas empresas, regularmente no se contempla la planificación de los proyectos a realizar, es decir, no se considera la gestión del tiempo, esfuerzo, o calendario. Esto provoca una incorrecta o nula estimación de los recursos necesarios para el proyecto y como consecuencia se propicia un deficiente control sobre el desarrollo de los proyectos, ocasionando entregas tardías, costos elevados de desarrollo y mantenimiento, y en el peor de los casos la cancelación del proyecto. En este contexto, es importante resaltar que este problema afecta tanto a empresas grandes como a PyMES, puesto que un *pequeño equipo* puede ser formado en cualquier entorno. Sin embargo, en la industria mexicana de software este aspecto impacta negativamente a las más pequeñas.

Así, es importante analizar cómo se encuentra conformada esta industria en nuestro país. Aunque ésta se encuentra en constante crecimiento y evolución, actualmente se le está prestando especial atención puesto que más del 98% de las empresas que la conforman se encuentran en la categoría de micro, pequeñas y medianas empresas; la mayoría de las cuales carecen de la infraestructura adecuada y un enfoque formal para sus actividades de desarrollo, venta y elaboración de contratos de software. Además, estas empresas por lo regular proporcionan servicios con actividades muy específicas a los diversos sectores de la economía mexicana. Por ejemplo, de acuerdo con la *International Data Corporation* (IDC)<sup>1</sup> durante el 2012 el mercado mexicano de software creció \$2,100 MDD y presentó un crecimiento de 8.5% frente al 2011. IDC destaca que este resultado se debió a una gran concentración de inversión del sector corporativo en general, enfocado principalmente en las industrias de finanzas, gobierno, petróleo, gas y minería. Particularmente, con base en el último conteo del Instituto Nacional de Estadística y Geografía (INEGI), se calcula que existen 3,237 unidades económicas relacionadas con el sector de las Tecnologías de la Información (TI) en el país, de éstas el 53% están concentradas en cuatro entidades federativas: Distrito Federal (32%), Nuevo León (9%), Jalisco (7%) y el Estado de México (5%) (Rivera & Quiroz, 2013).

Analizando más a detalle el comportamiento de la industria, en México se desarrolla software empaquetado (aplicaciones, herramientas de software, infraestructura y seguridad) y se proporcionan servicios relacionados con el desarrollo de software que es distribuido por empresas locales (tanto a nivel nacional como internacional), incluidas las importaciones de software (Mochi & Hualde, 2009). Sin embargo, la industria mexicana de software representa un caso atípico de producción donde las mayores ganancias las genera solamente el 1.5% de empresas que cuentan con 250 o más empleados (denominadas empresas grandes), mientras que el 98.5% son micro, medianas y pequeñas empresas que cuentan con menos de 50 empleados y que difícilmente obtienen un contrato importante que, como consecuencia, les permita alcanzar las ganancias de las grandes

---

<sup>1</sup> IDC (*International Data Corporation*) es la empresa líder en inteligencia de mercado, consultoría y eventos en las industrias de Tecnologías de la Información, Telecomunicaciones y mercados de consumo masivo de tecnología. IDC analiza y predice las tendencias tecnológicas para que profesionales, ejecutivos e inversionistas puedan tomar decisiones de compras y negocios en estos sectores.

(Ruíz, Ledeneva, & Cuesta, 2014). Bajo estas condiciones, la participación de la industria de software es baja; sin embargo, es importante considerar dos problemas: en primer lugar, la industria de software solamente considera software empaquetado y no software a la medida, éste se registra en el segmento de servicios de TI. La razón principal de esto es, en parte, porque el software a la medida se considera una actividad específica a las necesidades del usuario y no se produce en grandes cantidades como el software empaquetado (Sampedro & Vera-cruz, 2008). En segundo lugar, los valores de los segmentos de servicios de TI son más grandes que los del software, dado que los primeros registran, además del software a la medida, otros servicios como la consultoría y la capacitación. Siguiendo este razonamiento, el total de la producción de software en México está representado por el 29.4% del total para el software empaquetado, el 8.0% para el software a la medida y el 62.6% para la producción en masa y el consumo personal. En este contexto, si la producción para el consumo personal del Gobierno y las industrias manufactureras fuera contratada con las empresas del país que representan alrededor del 98.5% del total de la industria, la demanda del software a la medida representaría aproximadamente el 70.5% del total de la industria. Esta demanda potencial podría ser útil para las empresas de software siempre y cuando posean la capacidad tecnológica y el conocimiento necesario para ofrecer soluciones a problemas específicos en el sector gubernamental e industrial y además, si el Gobierno y las empresas establecieran los incentivos apropiados para dejar de producirlo por sí mismos y contratar a las empresas de software independientes. Pero menos podría ser más, es decir, al capacitar y apoyar a las PyMES se podría buscar reducir la cantidad de proyectos que las empresas grandes realizan (de las cuales un porcentaje importante está respaldado por inversiones extranjeras) y empezar a buscar el aumento de las ganancias para las empresas que representan la casi totalidad de la industria.

Dentro de este panorama, las PyMES desean mejorar la calidad de sus procesos, implicando así al incremento de la calidad de sus procesos, y en consecuencia, la mejora de sus productos a través de programas de SPI (López, Cabrera, & Valencia, 2008; Pino, Pardo, García, & Piattini, 2010). Tal como se mencionó anteriormente, Sosa et al. (2013) argumentan que una de las razones fundamentales que motiva a las PyMES a incursionar en este ámbito es la competencia con las empresas grandes dentro del sector. Sin embargo, esto no será posible si los problemas de estas empresas no son resueltos antes por los especialistas. En este sentido, William E. Deming<sup>2</sup> argumentó que “*No puedes controlar lo que no puedes medir, y no puedes medir lo que no puedes planear*”, lo que representa la problemática central de esta tesis como se demostrará más adelante.

En base a lo anterior, es importante trabajar en el desarrollo de propuestas que proporcionen apoyo a las PyMES mexicanas para desarrollar sistemas de calidad, aunque dispongan de escasos recursos financieros y humanos. Esto con el objetivo de obtener mejoras en sus procesos y productos a bajo costo, y brindar la seguridad de ofrecer productos de calidad que mejoren su ventaja competitiva en el sector.

## 1.1. Importancia del problema y necesidad de la solución

De acuerdo con (Calvo-Manzano, García, & Arcilla, 2008), la “crisis del software” del año 1969 perdura hasta nuestros días, hasta el punto de que aún están presentes los viejos problemas relativos al fracaso de los proyectos. En este sentido, de acuerdo con el Manifiesto del *Chaos*

---

<sup>2</sup> William E. Deming fue un ingeniero estadounidense especializado en la estadística y la gestión. Las aportaciones de Deming a la Calidad Total han sido de mucha utilidad para que la calidad de una empresa se desarrolle. La filosofía de Deming se basa en descubrir mejoras en la calidad de los productos y servicios, puesto que afirmaba que a mayor calidad, menores costos y mayor productividad. Una de las aportaciones a la Ingeniería de Software más reconocidas de Deming es el ciclo Planear-Hacer-Verificar-Actuar (PDCA, por sus siglas en inglés).

*Report*<sup>3</sup>, publicado por el Standish Group International (2013), existen 10 factores de éxito que permiten que un proyecto sea desarrollado de acuerdo a las necesidades y funciones requeridas por un cliente, y dentro del tiempo y presupuesto inicialmente establecidos. Estos factores son listados a continuación:

- Factor 1. Apoyo a la gestión ejecutiva.
- Factor 2. Participación del usuario.
- Factor 3. Optimización.
- **Factor 4. Recursos especializados.**
- **Factor 5. Experiencia en la gestión de proyectos.**
- Factor 6. Proceso ágil.
- Factor 7. Claridad de los objetivos de negocio.
- Factor 8. Madurez emocional.
- **Factor 9. Ejecución.**
- Factor 10. Herramientas e infraestructura.

En este sentido, es verdad que estos factores afectan a empresas de todos los tamaños dado que, como se indicó anteriormente, un *pequeño equipo* puede ser integrado en cualquier entorno; sin embargo, analizando particularmente el caso de los equipos que son formados dentro de las PyMES es posible identificar tres factores (resaltados de la lista) que serán abordados en este proyecto de tesis:

- El factor de *recursos especializados*, se refiere a que una empresa relacionada con la TI debe contar con recursos calificados para desarrollar los proyectos, o bien como el Manifiesto del *Chaos Report* lo describe: “tener a la gente adecuada haciendo el trabajo correcto”. Sin embargo, a menudo en una PyMES el personal puede desempeñar muchos roles a la vez, lo que origina que la mayoría de las veces éste deba auto-capacitarse. Por lo tanto, en este tipo de empresas la formación de pequeños equipos de desarrollo es una tarea difícil, lo que genera importantes repercusiones para cada proyecto que impactan directamente en el éxito del mismo. Así pues, la Figura 1.2 muestra que las empresas encuestadas afirman que el contar con personal competente incrementa las probabilidades de éxito de los proyectos.
- Por otro lado, la *experiencia en la gestión de proyectos* se refiere a la importancia del conocimiento técnico y gerencial del jefe de proyectos para tomar un proceso complejo y hacerlo sencillo y repetible. Es decir, la actividad principal que un jefe de proyectos debe realizar es administrar el avance natural de un proyecto para conducirlo a una resolución exitosa. De acuerdo con el *Standish Group*, aquellos proyectos que cuentan con el liderazgo apropiado y el buen juicio de un jefe de proyectos talentoso y una organización que lo

---

<sup>3</sup> El *Chaos Report* o Reporte del Caos en español, es un informe que realiza el *Standish Group*, con la finalidad de medir el éxito y fracaso de los proyectos de TI, siendo el informe más famoso de la industria y tomado como referencia para intentar analizar el comportamiento de la Ingeniería de Software moderna. Los datos proporcionados por el *Chaos Report* son concentrados cada cuatro años y resumen las respuestas de empresas encuestadas en los Estados Unidos de Norteamérica (60%), Europa (25%), y el resto del mundo (15%), pero que bien pueden ser utilizados para entender el fracaso de los proyectos a nivel mundial en las PyMES por su edición 2012 que se enfoca a pequeños proyectos y pequeñas empresas.

respalde, tendrán mayor control sobre su gestión y éxito. Sin embargo, el no tener la capacidad de emitir un juicio adecuado sobre las cuestiones relacionadas con la gestión del proyecto y tomar malas decisiones, puede conducir a un aumento de tiempo y costo o, en el peor de los casos, el fracaso rotundo del proyecto. Considerando este panorama, un *pequeño equipo* formado en una *pequeña empresa* se encuentra frecuentemente con esta problemática al no contar con herramientas de gestión o las habilidades requeridas para la toma de decisiones.



**Figura 1.2.** Factor de éxito en proyectos con pequeños equipos (Standish Group International, 2013).

- Por último, el factor relacionado con la *ejecución*, que es definida por el Manifiesto del *Chaos Report* como el acto de conducir un proyecto hasta su finalización en base a un plan, debe realizarse combinando pequeños proyectos que sigan una metodología formal, la cual con el paso del tiempo y al cabo de la experiencia acumulada por el personal puede cambiarse a una ágil. Pero para poder lograr esto, el equipo de desarrollo debe asumir y estar de acuerdo con lo que se espera de él, y conocer qué y cómo se medirá el progreso del proyecto. En este sentido, esta es una situación que una PyMES con *pequeños equipos* conoce muy bien; es decir, estas empresas están acostumbradas a desperdiciar recursos cuando el objetivo de un proyecto es pequeño (lo terminan pero con muchos problemas), pero cuando se trata de un proyecto fuera de lo habitual (mayor complejidad, menos tiempo), el equipo tiende a fracasar lo que provoca conflictos entre los miembros.

Así pues, de acuerdo con estos tres factores las expectativas de un proyecto deben ser alcanzables y razonables dado un conjunto de habilidades y recursos. Por lo que, en el caso particular de las PyMES, se recomienda cumplir, o superar si es posible, las expectativas de las partes interesadas a través de pequeños equipos que conduzcan al éxito del proyecto (véase Figura 1.3).

Sin embargo, lograr que un *pequeño equipo* se comporte como la gráfica lo indica es una tarea demasiado compleja. Por ejemplo, en el entorno de las PyMES con frecuencia la metodología se interpone en el camino del progreso y es adecuada por actividades *ad hoc* que mantienen a los equipos de desarrollo en una zona de confort. Por otra parte, el desconocer el desempeño del proyecto y el rendimiento del equipo de trabajo es riesgoso para la PyMES puesto que se pierde el factor de predictibilidad y no le permite saber qué y cuándo se entregará, o incluso determinar cuándo existe una alta probabilidad de que se produzca una desviación en el avance del proyecto.



**Figura 1.3.** Factor de éxito con equipos de cuatro a seis integrantes (Standish Group International, 2013).

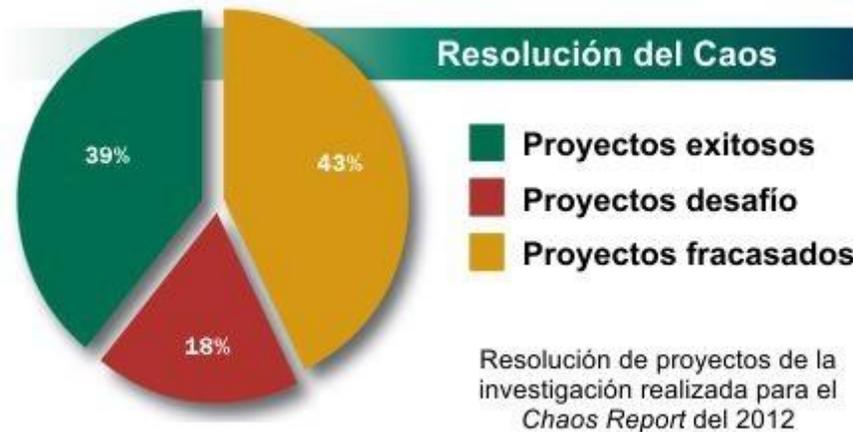
En este sentido, para que un *pequeño equipo* funcione correctamente es imprescindible que sus integrantes planeen y controlen sus actividades de una forma práctica y sencilla, para cubrir los tres factores de éxito mencionados anteriormente. Pero, ¿por qué es necesario enfocarse en estas actividades? A continuación se analizarán datos crudos que evidencian la necesidad urgente de enfocarse a su estudio.

De acuerdo con la Figura 1.4, que resume los últimos datos proporcionados en el 2012 por el *Standish Group* sobre 50,000 proyectos<sup>4</sup>, el éxito o fracaso de los proyectos software se desarrolló en base a los siguientes porcentajes:

- El 18% de los proyectos de software son cancelados o entregados y nunca utilizados.
- El 43% de los proyectos fueron modificados continuamente dados los desafíos que se presentaron durante su desarrollo, lo que originó que se excedieran el tiempo de entrega y los costos previstos, y/o que los proyectos fueran terminados con menos características y funciones necesarias.
- El 39% de los proyectos fueron completados dentro del tiempo y presupuesto establecidos y con las características y funciones requeridas.

Otra importante referencia sobre las causas de fracaso de los proyectos de software, es la investigación de Christopher Brooks presentada en el Colegio de Ingeniería y TI de la Universidad de Berkeley. De acuerdo con el estudio presentado por Brooks (2012) sobre diversas empresas gubernamentales de los Estados Unidos de Norteamérica (incluido el FBI), es posible identificar las causas raíz de ocho de los problemas más comunes que pudieron propiciar el fracaso de los proyectos realizados por estas empresas durante 2010-2012 (véase Tabla 1).

<sup>4</sup> Los informes del *Standish Group* clasifican a los proyectos en tres categorías: Resolución de Tipo 1, o proyecto exitoso, cuando el proyecto se termina a tiempo y dentro del presupuesto establecido, y ofrece todas las características y funciones que se especifican al principio; Resolución de Tipo 2, o proyecto desafío, cuando el proyecto está terminado y en uso, pero excedió el presupuesto y el tiempo establecidos, y ofrece menos características y funciones de las especificadas originalmente; Resolución de Tipo 3, o un proyecto fracasado, cuando el proyecto fue cancelado en algún momento del ciclo de desarrollo.



**Figura 1.4.** Resolución de proyectos en el 2012 (Standish Group International, 2013).

**Tabla 1.** La importancia de la gestión del proyecto en las causas de los problemas en los proyectos (Brooks, 2012).

Causa del problema	Gestión del proyecto	TI	Metodología de software
1. Requisitos pobremente definidos	X		
2. Debilidades contractuales	X		
3. Debilidad en la gestión de inversiones de TI		X	
4. Carencia de una arquitectura empresarial		X	
5. Falta de continuidad en la gestión y control	X		
6. Planificación poco realista de las tareas	X		
7. Carencia de una integración adecuada del proyecto			X
8. Resolución inadecuada de los problemas reportados en los informes del proyecto	X		

Es importante mencionar que estos datos fueron recogidos por Brooks de proyectos que fueron iniciados con contratos firmados sin requisitos definidos, sin hitos especificados, sin puntos de revisión para decisiones críticas, ni penalidades por mal desempeño del contrato. Para variar, condiciones hasta el día de hoy presentes en la mayoría de las PyMES. A través del estudio de Brooks se puede identificar que el 62.5% de las causas que originan los problemas se relacionan con la gestión del proyecto, pero concretamente las causas 5, 6 y 8 están vinculadas directamente con el problema abordado en esta tesis. Por último, los hallazgos de Brooks resumen los principales factores que propician el fracaso en los proyectos de software y se definen como:

- Metas poco realistas o no articuladas para el proyecto.
- **Estimaciones inexactas de los recursos necesarios para desarrollar el proyecto.**
- Requisitos del sistema mal definidos.
- **Información deficiente sobre el estado del proyecto.**
- Riesgos no gestionados.
- Falta de comunicación entre los clientes, desarrolladores y usuarios.
- Uso de una tecnología inmadura.
- **Incapacidad para manejar la complejidad del proyecto.**

- **Prácticas descuidadas de desarrollo.**
- **Mala gestión del proyecto.**
- Políticas de las partes interesadas.
- Presiones comerciales.

En resumen, una tercera parte de los factores de fracaso mencionados anteriormente (y que han sido resaltados de la lista) se relacionan con las actividades principales realizadas en la planificación y control de los proyectos.

En este sentido, los resultados mostrados por Brooks soportan los hallazgos proporcionados por el estudio conducido por Nizam y Sahibuddin (2011). Dicho estudio considera, de manera similar al realizado por Brooks, que es importante asumir tres factores críticos para el éxito de un proyecto: requisitos y especificaciones claras, objetivos y metas bien definidas, y un calendario realista. Estos tres factores identificados por Nizam y Sahibuddin a menudo son considerados tareas importantes de un pre-proyecto que deben realizarse de forma sólida antes de iniciar y ejecutar un proyecto de software. Sin embargo, es difícil encontrar que en el entorno de la PyMES éstas se realicen de manera correcta.

En este contexto, es verdad que los factores de fracaso mencionados a lo largo de estas páginas afectan a las empresas de todos los tamaños; sin embargo, las PyMES están más expuestas a las consecuencias del fracaso. Entonces, ¿cómo es que la comunidad de investigadores relacionados con la Ingeniería de Software ha intentado resolver este problema? En la actualidad es común escuchar que las PyMES intentan planificar y controlar sus proyectos mediante la adopción de metodologías ágiles como Scrum o XP, puesto que éstas plantean el realizar lanzamientos pequeños y frecuentes, a modo de evitar que el software se convierta en obsoleto o aumente en complejidad en el lapso entre un lanzamiento y otro. Pero, cómo utilizar una metodología ágil que depende de la experiencia del personal para tener éxito, cuando éste no tiene la experiencia o habilidades necesarias. Simplemente, es intrigante el pensar en que una PyMES pueda implementar la técnica de *card planning*, recomendada por Scrum para estimar la duración de las actividades del proyecto, si el equipo de trabajo no tiene experiencia previa con proyectos de diferente tamaño y complejidad. Es innegable que esta técnica se puede implantar en un pequeño entorno de trabajo, pero para esto es necesario primero aprender y practicar.

Así, una opción muy utilizada en la actualidad para solventar la falta de conocimiento requerido por los enfoques ágiles es el uso de herramientas computarizadas para realizar la planificación y control del ciclo de desarrollo. Por ejemplo, Marante, Letelier, y Suárez (2009) proponen la herramienta TUNE-UP como resultado de una iniciativa de SPI en una PyMES. La herramienta TUNE-UP combina aspectos importantes de las metodologías ágiles y tradicionales, pero su característica más innovadora fue el énfasis en la gestión de tiempos como factor clave para la planificación y control de un proyecto.

Por otro lado, García, Pacheco, y Calvo-Manzano (2014) plantean el uso del Marco de Trabajo FQPMSE (*Framework for Quantitative Project Management in Small Enterprises*) junto con el uso de dos herramientas computarizadas, PALSS (*Process Asset Library for Small Settings*) y PROMEP (*Project Management based on Effective Practices*), que permiten definir un proceso eficiente para la gestión de proyectos como resultado de una iniciativa de SPI.

Por último, García (2014) introduce la Herramienta para la Integración de MoProSoft con Scrum (orientada exclusivamente a los procesos de Desarrollo y Mantenimiento de Software (DMS) y Administración de Proyectos Específicos (APE) en su Nivel 2) como medio para agilizar la

adopción de la Norma NMX-I-006/02-NYCE-2004 (que representa el qué), a través de la introducción de actividades concretas de Scrum (que representan el cómo).

Estos trabajos evidencian la importancia del problema que se pretende abordar con esta tesis, pero a diferencia de éstos éste trabajo plantea la idea de definir un marco conceptual que almacene todo el conocimiento necesario para que un pequeño equipo dentro de una pequeña empresa sea capaz de definir e implantar procesos de manera correcta. Una vez que el marco esté definido, una herramienta computacional serviría solamente de soporte para llevar a la práctica el conocimiento almacenado.

En conclusión, en este contexto es donde se sitúa la presente tesis redactada en este documento. Se asume pues que para resolver la problemática de que una PyMES sea capaz de establecer un proceso efectivo para planificar y gestionar sus proyectos, y por ende reducir sus factores de fracaso, es necesario proporcionarle herramientas útiles que describan qué tiene que hacer, cómo tiene que hacerlo, qué herramienta debe utilizar para hacerlo, y cómo saber que lo hizo bien, todo esto a través de una forma práctica y sencilla. En este sentido, un metamodelo con soporte computacional es una decisión inteligente para alcanzar este objetivo.

## 1.2. Hipótesis de la tesis

La hipótesis del presente trabajo se plantea como una relación causal y se enuncia de la siguiente manera:

***“Mediante el uso de un metamodelo un pequeño equipo puede reducir el tiempo para definir e implantar los procesos de planificación y control de proyectos dentro de una PyMES.”***

## 1.3. Objetivos de la tesis

Considerando la hipótesis anterior, a continuación se especifican el objetivo general y los objetivos específicos de la tesis desarrollada en este documento:

### 1.3.1. Objetivo general

*Crear un metamodelo que reduzca el tiempo de la definición e implantación de los procesos de planificación y control de proyectos en pequeños equipos dentro de las PyMES.*

### 1.3.2. Objetivos específicos

Para alcanzar el objetivo general será necesario conseguir ciertos objetivos específicos. Éstos establecen las aportaciones esperadas al final de la tesis y se resumen de la siguiente manera:

1. Realizar una revisión de literatura sobre iniciativas de SPI enfocadas a los procesos de planificación y control de proyectos en el contexto de *pequeños equipos dentro de pequeñas empresas*.
2. Como consecuencia de esta revisión, analizar los modelos y estándares propuestos en la literatura con el objeto de identificar las principales lecciones aprendidas.
3. Diseñar y documentar un metamodelo para definir e implantar los procesos de planificación y control de proyectos.
4. Crear el soporte computacional que permita introducir el metamodelo creado en pequeños equipos.

5. Definir un caso de estudio para probar el metamodelo creado en, al menos, un entorno real de trabajo. En el mejor de los casos, este caso de estudio puede incluir a un mayor número de empresas siempre y cuando se logre captar su atención a través de mejoras permanentes.
6. Obtener datos sobre la implantación del metamodelo, a través del caso de estudio, para confirmar, o no, la hipótesis establecida anteriormente. De manera concreta, para comprobar la reducción en el tiempo de establecimiento de los procesos mencionados será necesario definir un grupo experimental (que utilizará la propuesta definida en esta tesis) que trabajarán, dentro del caso de estudio, en un proyecto almacenado en los datos históricos de la empresa (línea base) intentando mejorar los resultados obtenidos. Los datos serán recogidos a través de un proceso de observación.

#### 1.4. Delimitaciones y limitaciones de la tesis

Esta tesis está enmarcada por los siguientes factores:

- El metamodelo está orientado específicamente a los procesos de planificación y control de proyectos, puesto que son actividades esenciales para el éxito de los proyectos de software de cualquier PyMES sin experiencia.
- El metamodelo está orientado a pequeños equipos formados dentro de pequeñas empresas, puesto que la industria mexicana de software está constituida principalmente por PyMES y es necesario realizar propuestas de acuerdo al contexto nacional.
- El metamodelo será probado en, al menos, una PyMES del Estado de Oaxaca.
- La resistencia al cambio puede retrasar la implantación del metamodelo, puesto que para cualquier organización el cambio significa, de alguna manera, miedo a lo desconocido y se puede dudar sobre la obtención de beneficios sobre el proceso.
- La cultura organizacional de la PyMES puede también representar un obstáculo en la implantación del metamodelo, puesto que el personal de software (y en general la organización) está acostumbrado a las prácticas *ad hoc* que ya están demasiado arraigadas y tienden a crear costumbres que con el tiempo se vuelven cuestiones difíciles de cambiar.

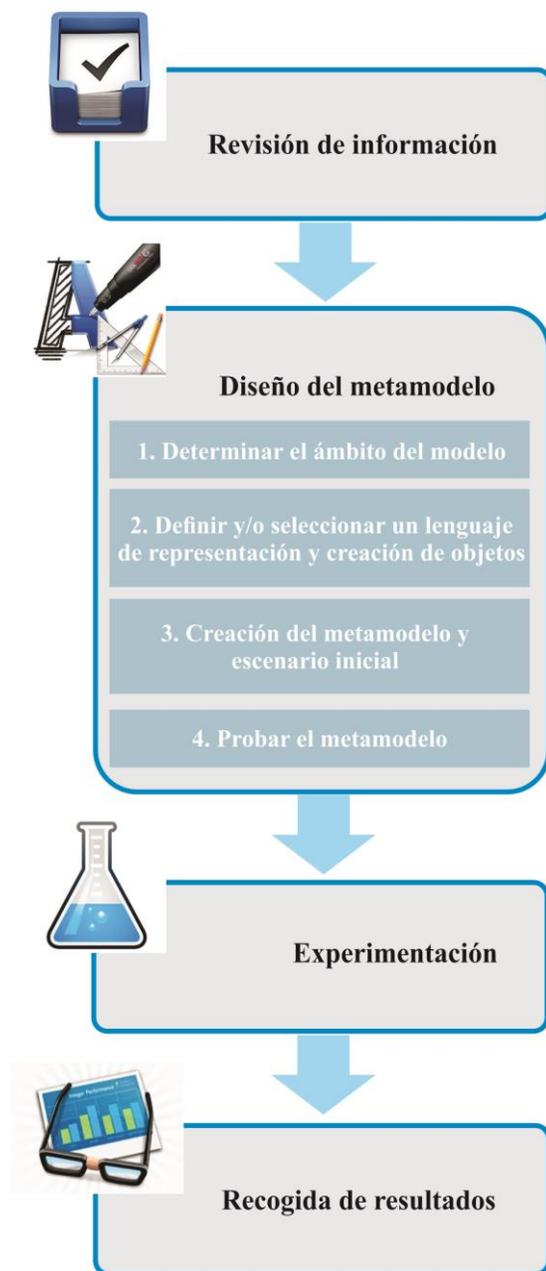
#### 1.5. Metodología de trabajo

Una vez que se han definido las partes elementales que delimitan el proceso de investigación de la tesis, a continuación se describe de forma general la metodología que será utilizada para su desarrollo. En este sentido, la Figura 1.5 muestra las fases que serán realizadas y que son descritas a continuación:

1. Revisión de información: Revisar toda la información correspondiente a los procesos de planificación y control de proyectos en pequeños equipos dentro de las PyMES, como pueden ser modelos de procesos y estándares, con el fin de obtener un *benchmarking*<sup>5</sup> para continuar con la implementación del modelo.

---

<sup>5</sup> El *benchmarking* es un anglicismo que, en las Ciencias de la Administración de Empresas, puede definirse como un proceso sistemático y continuo para evaluar comparativamente los productos, servicios y procesos de trabajo en organizaciones que se reconocen como representantes de las mejores prácticas (Spendolini, 1992). El análisis *benchmarking* en esta tesis se enfoca en analizar comparativamente las investigaciones realizadas previamente, con el objetivo de situar en un contexto actual la solución propuesta en la misma.



**Figura 1.5.** Metodología propuesta para crear la solución de esta tesis.

2. **Diseño del metamodelo:** Esta fase establece cuatro actividades para el desarrollo del metamodelo:
  - **Determinar el ámbito del modelo:** Con esta actividad se pretende crear un conjunto de fichas bibliográficas sobre el tema, identificando los elementos y procesos esenciales de la realidad actual sobre metamodelos orientados a resolver el problema planteado. Estos elementos se pueden modelar con un lenguaje formal o informal, lo importante es su independencia con el software, hardware o personas. El resultado final de esta etapa será una serie de diagramas que formalizan la abstracción realizada de la realidad.

- Definir y/o seleccionar un lenguaje de representación y creación de objetos del metamodelo: Para realizar esta actividad será necesario seleccionar un lenguaje ya definido y estandarizado que se adecue a las necesidades de representación de la información del modelo, como por ejemplo UML, o en su defecto realizar la definición de un nuevo lenguaje el cual debe empezar por definir la filosofía, elementos estáticos y dinámicos del metamodelo y los diagramas a construir.
  - Crear el metamodelo y escenario inicial: En esta actividad será importante definir una serie de reglas de creación de modelos (a partir del metamodelo), por medio de las cuales se determinen los pasos que cualquier persona debe seguir para implantar un modelo personalizado. Y a su vez proponer un modelo específico desarrollado con el metamodelo, esto con el fin de evaluar la utilidad y poder llevar a cabo la implantación del metamodelo en un entorno real. Para esto será importante definir el ciclo del modelo, a través del cual se especifique una manera de evaluación, se definan las métricas a monitorizar y por consiguiente lograr una realimentación de mejora continua.
  - Probar el metamodelo: Esta actividad establece que la primera prueba del metamodelo se realiza con el modelo específico propuesto, para esto se creará una herramienta computacional que servirá de soporte en las pruebas del metamodelo dentro de un entorno real.
3. Experimentación: Esto significa que primero se debe de realizar la elección de la empresa considerando la base de datos históricos que posee de todos sus proyectos, obviamente tomando en cuenta su disponibilidad de colaboración en la implantación de un proyecto piloto, así como la realización y cumplimiento del convenio que contenga las bases para la implementación exitosa de éste. Una vez sentadas las bases de la experimentación se deberá proceder al diagnóstico de la empresa específicamente en los procesos de planificación y control de proyectos para situarse en el contexto real de la misma, posteriormente se debe de llevar a cabo la implantación del metamodelo en un proyecto piloto a través de la herramienta computacional diseñada. Por último, se realizará una evaluación final en la organización para analizar los resultados con respecto al diagnóstico inicial y verificar si los objetivos establecidos (indicadores de mejora) fueron alcanzados o no. En concreto, se diseñará un caso de estudio que permita demostrar, o no, la utilidad del metamodelo creado.
  4. Recogida de resultados: En esta etapa se analizarán los resultados obtenidos durante la aplicación del metamodelo en un entorno real, y se contrastarán con la hipótesis inicial para concluir si ha sido aceptada, o no, en el contexto de la tesis. Por último, también se documentará todo el proceso realizado a lo largo del proyecto piloto.

## 1.6. Aproximación a la solución

La presente tesis asume que es más fácil que los pequeños equipos de trabajo implanten los procesos de planificación de proyectos y control de proyectos cuando existe un metamodelo que soporte las actividades de la PyMES y que promueva la repetitividad del éxito a través del aprendizaje de los proyectos anteriores. Mejor aún, si una vez que los procesos mencionados son implementados de manera eficiente, el metamodelo estaría agregando una alta dosis de

institucionalización<sup>6</sup> en las actividades diarias de la PyMES. En este sentido, es importante definir primero el concepto de metamodelo y la forma en que éste se adecua al contexto de esta tesis.

De acuerdo con (Pérez-Castillo, García-Rodríguez, & Piattini, 2011), un metamodelo describe de forma detallada las actividades que deben realizarse en las organizaciones, puesto que cada una de éstas puede construir a través de organizaciones jerárquicas, y haciendo uso de los activos que componen al metamodelo, diferentes modelos específicos de acuerdo a las características y contexto de cada proyecto. Tal y como ocurre con otros artefactos del software, los metamodelos evolucionan con el tiempo debido a varias razones: durante su diseño, se crean versiones alternativas del metamodelo y se genera un conjunto de soluciones que será personalizado para diferentes aplicaciones (usos). Durante su implementación, los metamodelos son adaptados a un modelo concreto que es soportado por una herramienta de software. Durante su mantenimiento, se corrigen los errores en el metamodelo para mejorar su rendimiento. Por otra parte, las partes del metamodelo son rediseñadas para mejorar su entendimiento o para facilitar su reutilización (Wachsmuth, 2007). Es decir, en el contexto de la Ingeniería de Software, un metamodelo especifica de manera consistente diferentes artefactos que pueden ser utilizados de diferentes maneras en diferentes proyectos a través de diferentes capas y vistas. En el entorno de una PyMES, la ventaja más importante que se obtiene de esta idea se relaciona con un ahorro considerable en costo y esfuerzo puesto que los equipos disponen del conocimiento obtenido con los proyectos realizados y que les permite, en primera instancia, reutilizar los artefactos y predecir el comportamiento de los proyectos futuros con planes más realistas y mecanismos de control más ajustados.

En este sentido, con la intención de aclarar un poco más la utilidad de un metamodelo en el contexto de la Ingeniería de Software es común referirse al metamodelo SPEM (*Software & Systems Process Engineering Metamodel Specification*), que es el estándar definido por el OMG (*Object Management Group*) para la representación de procesos de desarrollo y establece un lenguaje de referencia mediante el cual los procesos de software son definidos. Así, SPEM ofrece un marco de trabajo para el modelado, documentación, presentación, gestión e intercambio de los procesos de desarrollo de software y sus componentes, a través de una sintaxis y una estructura común para cada aspecto del proceso (OMG, 2002). Es decir, SPEM representa una guía teórica que proporciona la información necesaria para definir un proceso de software, y a partir de ésta pudieron desarrollarse decenas de herramientas computacionales que implementan dicha guía en modelos que pueden ser compartidos entre diversos grupos de trabajo. Esta es pues, la definición más general del concepto de “metamodelo” en el contexto de la Ingeniería de Software.

Sin embargo, a diferencia de SPEM que, como se ha mencionado, es un metamodelo para la definición general de los procesos de software y por ende su alcance se limita a los elementos mínimos necesarios para definir un proceso sin añadir características específicas de un dominio o disciplina particular y que es útil para modelar procesos de diferentes estilos, culturas, niveles de formalismo, o simplemente modelos de ciclo de vida; el metamodelo que se propone en esta tesis se enfocaría específicamente a facilitar la definición de los procesos de *planificación de proyectos* y *control de proyectos* a través de una arquitectura que relacione todos los elementos definidos en él y la adaptación de dichos procesos mediante una herramienta computacional que permitirá poner en práctica aquellas actividades que sean definidas para un proyecto en particular. En este mismo sentido, la propuesta de solución de esta tesis está más relacionada con KUALI-BEH (Oktaba, Morales Trujillo, & Dávila, 2012), que describe los conceptos comunes y sus relaciones para cualquier proyecto de software a través del conocimiento y experiencia obtenidos en múltiples proyectos, y recopilando las mejores prácticas de conjunto de estándares para el desarrollo de software. Así, los conceptos definidos en KUALI-BEH se pueden aplicar para definir métodos y

---

<sup>6</sup> De acuerdo con (Chrissis, Konrad, & Shrum, 2011) la institucionalización es “La forma arraigada de funcionamiento que una organización sigue rutinariamente como parte de su cultura corporativa.” (p. 626).

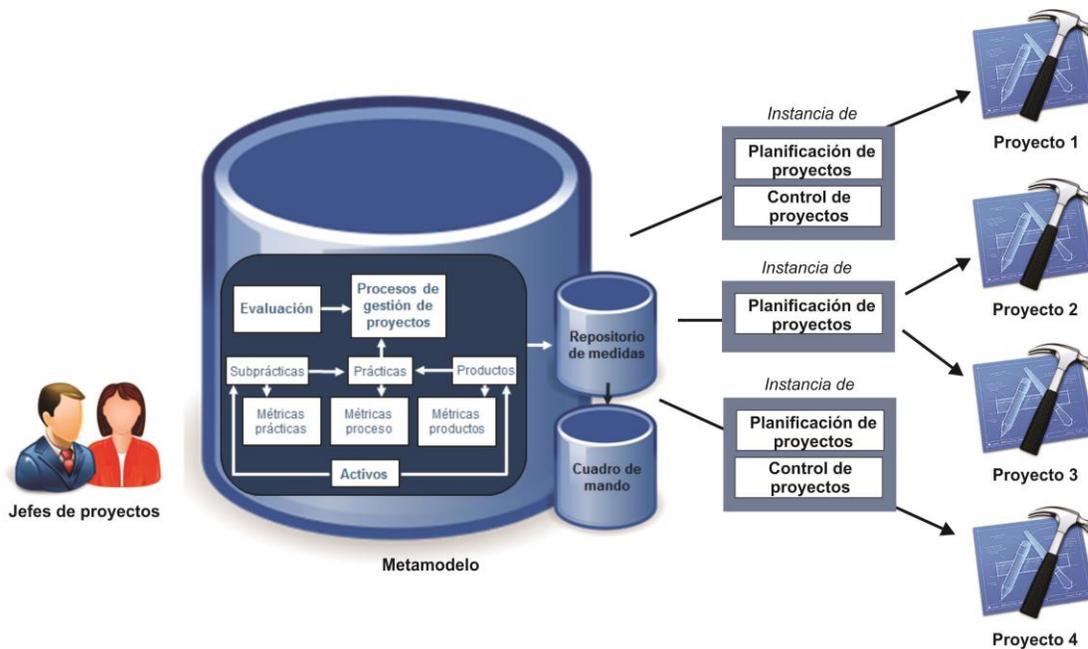
prácticas para cualquier proyecto de software, independientemente de su tamaño, complejidad, modelo de ciclo de vida utilizado o tecnología. El metamodelo propuesto en esta tesis, al igual que KUALI-BEH, estaría compuesto por prácticas, actividades, plantillas, y demás artefactos orientados a la definición y adaptación de procesos involucrados en el desarrollo de software: la planificación y el control. Es decir, la mayor diferencia entre el metamodelo propuesto en KUALI-BEH es que éste permitirá definir un patrón de proceso que puede ser adaptado a cualquier proyecto definido por la PyMES y establecer eficientemente su gestión.

Ahora bien, considerando los argumentos de Braun & Winter (2005) sobre el carácter pasivo y activo de un metamodelo se plantea que como instrumento pasivo, el metamodelo debe ayudar a detectar inconsistencias entre las especificaciones de un producto, los indicadores de rendimiento, las especificaciones de los procesos, las estructuras y flujos de información, y el diseño de aplicaciones. Si se utiliza de forma activa, el metamodelo debe guiar el rediseño de los procesos en todas las capas de la empresa, incluyendo cambios estratégicos, el rediseño organizacional, y el desarrollo de la Ingeniería de Software (partiendo de la propia idea de la empresa).

En este contexto es posible que alguien considere que existen diversos modelos (e.g., PMBoK, PSP, TSP) que pueden ayudar a las empresas a establecer un proceso de planificación y otro de control sin problemas. Sin embargo, ¿cuál será el modelo más adecuado para una PyMES? Es importante recordar que estos modelos constituyen el punto de partida para gestionar, documentar y mejorar los procesos, sin embargo ante la vasta diversidad de modelos existentes, un metamodelo representa una herramienta útil para ayudar a la PyMES a definir el proceso que más se ajuste a las particularidades de un proyecto.

En ese sentido, Bailey y Basili (1981) afirmaban hace más de 30 años que cualquier iniciativa relacionada con la incorporación de metamodelos como medio para mejorar el desarrollo de software debía considerar la propia productividad pasada de la empresa con el fin de poder predecir y estimar sus productividades futuras. Es decir, tradicionalmente un buen jefe de proyectos puede estimar la cantidad de recursos que necesita para un proyecto específico en base a su experiencia en ese entorno particular en el que se desarrolla dicho proyecto. Bailey y Basili argumentan que un metamodelo debe ser capaz de hacer lo mismo definiendo un marco de trabajo para utilizar la experiencia en forma de activos, y servir como una ayuda útil para cualquier jefe de proyectos que desee realizar esta tarea de estimación.

Así pues, la PyMES podría definir de forma paralela un proceso específico para cada proyecto a través de los modelos ya existentes, tomar decisiones sobre la marcha con respecto a los factores que afectan su entorno, y podría moldear el proceso definido de acuerdo a cualquier entorno específico. Para conseguir estas funciones, se define un repositorio o biblioteca de activos en función de un metamodelo. El metamodelo determina la forma en que se almacena la información en el repositorio, la forma en que las herramientas pueden acceder a los datos y estos datos pueden ser visualizados por los jefes de proyectos, el grado hasta el cual se puede mantener la seguridad e integridad de los datos, y la facilidad con que se puede ampliar el modelo ya existente para admitir nuevas necesidades (véase Figura 1.6). Es decir, definiéndolo de manera sencilla, un metamodelo es la representación jerarquizada de los elementos de un proceso de software en términos de políticas, estrategias y objetivos de negocio; mientras que la biblioteca de activos representa la plantilla en la cual se sitúa toda la información relacionada con la Ingeniería de Software, de acuerdo a la jerarquía y reglas definidas por el metamodelo.



**Figura 1.6.** Solución propuesta.

Es verdad que posiblemente algún modelo disponible podría brindar la capacidad de implementar de manera general en la PyMES un proceso para la gestión del proyecto (orientado a cubrir los procesos de planificación y control). Sin embargo, el metamodelo que se propone proporcionará a la PyMES la base cuantitativa para aplicar la nueva tecnología y controlar con datos reales los productos y procesos, y definirá los procesos que se integrarán con las prácticas de gestión y que serán institucionalizados en la empresa. Así, el metamodelo permitirá incrementar la calidad y la productividad, al mismo tiempo que reduce el riesgo de fracaso.

En este sentido, la Figura 1.6 ejemplifica la forma en que la solución propuesta en esta tesis, el metamodelo en sí, pretende resolver el problema planteado al inicio de este documento. El metamodelo, como se ha explicado anteriormente, estará formado por activos (productos, prácticas, subprácticas, métricas, guías, plantillas, ejemplos, etc.) que permitan que los jefes de proyectos definan su propio proceso para desarrollar un proyecto. Dado que se pretende establecer un control real sobre el trabajo del equipo de trabajo, se pretende incorporar un repositorio de medidas que permita monitorizar desvíos en el avance de los proyectos y un cuadro de mando que permita incorporar objetivos de negocio y estrategias de lanzamiento para cada proyecto. De esta manera, y a través de una herramienta computacional, los jefes de proyectos de la PyMES pueden definir diferentes procesos para diferentes proyectos, considerando las características de los mismos. Es decir, en base a los argumentos anteriores se pretende crear un metamodelo para que la PyMES cree el modelo que mejor le funcione dada la complejidad, alcance y características generales de un proyecto.

## 1.7. Estructura de la tesis

La estructura del documento de tesis se detalla a continuación:

El Capítulo 2 presenta el marco teórico sobre la planificación y control de proyectos en pequeños equipos formados en pequeñas empresas, el cual sirve de base para desarrollar un conjunto de características iniciales de trabajo. Se presenta también la exploración de literatura acerca de

metamodelos o herramientas computacionales similares, y mediante un estudio comparativo empírico se presenta por qué la solución propuesta en esta tesis es factible.

El Capítulo 3 expone la arquitectura del metamodelo (solución propuesta), definiendo así las reglas, políticas y guías de uso del mismo.

El Capítulo 4 presenta los resultados obtenidos al experimentar con el metamodelo creado a través de un caso de estudio que incluye a una PyMES desarrolladora de software.

El Capítulo 5 presenta las conclusiones finales obtenidas durante el desarrollo del presente trabajo y expone posibles líneas de trabajo futuro que corresponden a mejorar el uso y funcionalidad del metamodelo y su soporte computacional.

El Anexo A resume una lista de acrónimos.

El Anexo B proporciona información detallada sobre los cuestionarios de evaluación incluidos en el marco conceptual del metamodelo.

El Anexo C detalla todos los activos de proceso creados para la experimentación con el metamodelo.

El Anexo D ofrece una copia del acta de la publicación generada en esta tesis.

Por último se presentan las referencias bibliográficas utilizadas en el desarrollo de la tesis.

## 1.8. Publicaciones generadas

A continuación se muestra la información relacionada con la publicación generada durante el desarrollo del presente trabajo.

---

Autores:	García, I., Pacheco, C., Arcilla, M. & Sánchez, N.
Título:	“Project Management in Small-sized Software Enterprises: A Metamodeling-based approach”
Publicación:	Advances in Intelligent Systems and Computing Series, Springer-Verlag Heidelberg, Berlin.
Año:	2015.

---



## 2. Marco teórico

### 2.1. Antecedentes

La planificación es fundamental dentro del proceso de desarrollo de un producto de software (y para cualquier otro proceso en general). En ésta se establece, entre otras cosas, qué tareas y cuándo se van a realizar y la cantidad de recursos que utilizarán las mismas. En base a un plan de trabajo la gerencia es capaz de determinar si el proceso está dentro del tiempo establecido inicialmente y si éste está utilizando correctamente los recursos asignados (e.g., tiempo, personal, dinero, infraestructura) y de la forma esperada. Sin embargo, la planificación y el control de los proyectos no son parte de una ciencia nueva. De acuerdo con (Kwak, 2005), estos dos procesos, o bien la Gestión de Proyectos (*Project Management*), se convirtieron en una ciencia moderna a comienzos de 1900.

Así, la gestión de proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades de un proyecto, con el objetivo de cumplir con los requisitos que hayan sido planteados por el mismo. De acuerdo con el Cuerpo de Conocimiento sobre la Gestión de Proyectos (PMBok<sup>7</sup>, por sus siglas en inglés), esto se puede lograr fácilmente mediante la aplicación, e integración de cinco grupos de procesos: Inicio, Planificación, Ejecución, Monitoreo y Control, y Cierre (PMI, 2013). Sin embargo, la definición de estos procesos requirió muchos años de práctica y experiencia.

En este sentido, los primeros indicios sobre la gestión de proyectos se remontan a un proyecto que fue realizado de forma disciplinada por Taylor (1914), en el cual se aplicó el razonamiento científico para desarrollar el trabajo. Dicho proyecto ha sido replicado y mejorado por cientos de investigadores que se enfocan en sus partes elementales, a tal grado que es considerado en la actualidad como una referencia obligada para la creación de herramientas de gestión. Años más tarde, Henry L. Gantt desarrolló durante la Primera Guerra Mundial la famosa técnica de ayuda visual para el control del trabajo, una variante del diagrama de barras que lleva su nombre: el diagrama de Gantt (Gantt, 1919). Fue tanta la importancia del aporte de Gantt, que el Gobierno Federal de los Estados Unidos de Norteamérica fue uno de sus usuarios más destacados, y durante la Segunda Guerra Mundial el diagrama se convirtió en una herramienta de planificación aceptada totalmente en el mundo de los negocios. De acuerdo con Kerzner (2013), durante y después de la Segunda Guerra Mundial existía la necesidad de mucha coordinación para el despliegue de tropas (lo cual puede compararse con un proyecto por la cantidad de personas involucradas, el presupuesto

---

<sup>7</sup> La guía del PMBoK es el conjunto de conocimientos propuesto por el Instituto para la Gestión de los Proyectos (PMI, por sus siglas en inglés) para la gestión de proyectos, generalmente estos conocimientos son reconocidos como «buenas prácticas» que constituyen un estándar de gestión. El PMBoK comprende dos grandes secciones, la primera sobre los procesos y contextos de un proyecto, y la segunda sobre las áreas de conocimientos específicos para la gestión de un proyecto (PMI, 2013).

asignado, y los resultados esperados). Así, en este periodo es cuando las herramientas de gestión de proyectos se pusieron al servicio de los ejércitos. A pesar de estos casos exitosos de uso, durante este mismo periodo las prácticas de gestión “tradicionales” de proyectos serían consideradas, por el sector privado, una técnica de gestión sobrada que consumía mucho tiempo.

Posteriormente, durante los últimos años de 1950 el Dr. John W. Mauchly desarrolló el Método de la Ruta Crítica (CPM, por sus siglas en inglés), que analiza a un proyecto basándose en las tareas que se deben de realizar dentro del tiempo establecido, y que no pueden ser demoradas si se desea que el proyecto termine de acuerdo al plan previsto. La técnica de gestión del Dr. Mauchly fue ampliada más tarde por Willard Frazer, un consultor del proyecto Polaris, para el diseño de un submarino nuclear que fue desarrollado a finales de ese año por la Marina de los Estados Unidos de Norteamérica. La contribución de Frazer a la planificación de proyectos se denominó PERT (*Program Evaluation and Review Technique*). Originalmente, PERT proporcionaba tres estimaciones para un proyecto: la mejor (optimista), la peor (pesimista) y la más probable (realista); sin embargo, con el paso del tiempo estas tres estimaciones desaparecieron y en la actualidad el nombre PERT se asocia a un diagrama de flujo utilizado para representar la relación predecesora-sucesora entre las distintas tareas de un proyecto. PERT sería extendido años más tarde como una Estructura Desglosada de Trabajo (WBS<sup>8</sup>, por sus siglas en inglés). Hoy en día, estas dos técnicas gráficas, Gantt y PERT, son ampliamente utilizadas para obtener una representación descriptiva de cualquier proyecto. De esta manera, ambas técnicas fueron incorporadas en los primeros programas computacionales para la planificación de proyectos escritos para las grandes computadoras.

Así, en 1960 se desarrolló en Inglaterra un grupo de programas muy potentes para la planificación de proyectos, como Plantrac [URL-1], los cuales fueron posteriormente exportados a los Estados Unidos de Norteamérica. Fue a partir de esta década que la gestión de proyectos fue reconocida internacionalmente como profesión. En este sentido, en 1967 fue creada la Asociación Internacional para la Gestión de Proyectos (IMPA, por sus siglas en inglés) y al final de los 60's se funda el PMI en los Estados Unidos de Norteamérica.

La década de los 70's se caracterizó por la realización de estudios empíricos sobre la gestión, de los que nacen modelos y métodos como el modelo de Putnam (1978) que introduce la ecuación de Raleigh-Norden para definir una segunda ecuación para calcular el esfuerzo y el tiempo dedicado a un proyecto, y la Revisión Programada de Información para la Determinación y Evaluación de Frank Freiman para predecir con bastante exactitud el costo de los nuevos sistemas militares antes de completar el diseño y obtener un desglose de las piezas para la estimación de costos [URL-2].

A partir de la década de los 80's se presenta una confrontación entre los modelos y las metodologías para la gestión a través de la controversia introducida sobre su utilidad en los proyectos pequeños y los grandes. Como consecuencia de esta discusión, durante este periodo surgen nuevos métodos y metodologías como los Puntos Función (Albrecht & Gaffney, 1983) que se orientaban en medir el tamaño funcional de los sistemas orientados a la gestión, y el Modelo Constructivo de Costos (o COCOMO, por sus siglas en inglés) propuesto por Boehm (1981) para estimar los costos de un proyecto de software.

Durante los 90's, las principales casas de software que desarrollaron todos los programas computacionales para brindar soporte a la gestión de los proyectos (e.g., MS Project y Excel de Microsoft, ERP de Oracle, HP y SAP), pusieron a disposición del público en general la automatización de las técnicas de gestión que fueron desarrolladas en décadas anteriores. Esto trajo como consecuencia la ampliación, y sobretodo el reconocimiento masivo e impulso renovado a la gestión de proyectos. Es importante resaltar que con la consolidación de la infraestructura de Internet también surge una nueva tendencia de colaboración, basada en web, de diferentes tipos de

---

<sup>8</sup> De acuerdo con el CMMI-DEV v1.3 la estructura de descomposición del trabajo (o WBS, por sus siglas en inglés) es una disposición de los elementos de trabajo y de sus relaciones entre ellos y con el producto final (CMMI, 2010).

equipos (de alta o baja proximidad), los cuales trabajan en diferentes tipos de proyectos (grandes y complejos o pequeños y simples) para obtener una amplia colaboración en los proyectos.

Al final de 1990 se presentaron cambios importantes en cuanto a la realización de la planificación de los proyectos y se introdujeron nuevas tendencias como el uso de redes neuronales (Dohi, Nishio & Osaki, 1999). Esta iniciativa planteaba que para decidir cuándo se debía entregar el producto de software a los usuarios, se debía usar un método para estimar el tiempo óptimo de liberación del software, de tal forma que se minimizara el criterio de costo relevante a través de redes neuronales artificiales.

El comienzo del año 2000 fue determinante para el avance y consolidación de la gestión de proyectos. En el 2005, por ejemplo, la Organización Internacional de Estándares (ISO, por sus siglas en inglés) creó un conjunto de normas para la gestión de la calidad que puede ser perfectamente aplicado a la gestión de los proyectos, la conocida norma ISO 9000 (Corbett, Montes-Sancho, & Kirsch, 2005). En este mismo contexto surgen dos metodologías que fueron desarrolladas en el SEI por Watts S. Humphrey y que se enfocaban principalmente en el recurso humano, PSP (*Personal Software Process*) (Humphrey, 1996) y TSP (*Team Software Process*) (Humphrey, 2000). Concretamente, el objetivo de PSP fue proveer al personal de una empresa los mecanismos de trabajo para que pudiera ser un miembro efectivo en los equipos de desarrollo de software, partiendo de la idea de que para alcanzar una disciplina a nivel de equipo es necesaria la disciplina personal. TSP, por su parte, pretende formar y desarrollar equipos exitosos con miembros exitosos, una vez que éstos han obtenido la disciplina individual.

Durante la primera década del siglo XX se crearon algunos de los modelos más importantes para las empresas de desarrollo de software que pretendían gestionar correctamente sus proyectos, con el objetivo de maximizar sus beneficios. Debido a que los proyectos eran cada vez más complejos, las empresas comenzaron a introducir técnicas de gestión como el Enfoque basado en el Vector de Predicción (Hastings & Sajeev, 2001), la Estimación de Expertos (Anda, Angelvik & Ribu, 2002; Anda, Benestad, & Hove, 2005), el Método Hermes [URL-3], PRINCE2 (OGC, 2005), COBIT, Método-V XT (Biffel, Winkler, Reinhard, & Wetzel, 2006), COCOMO II (Menzies, Chen, Hihn, & Lum, 2006), y los Puntos Función basados en Analogías (Ohsugi, Monden, Kikuchi, Barker, Tsunoda, Kakimoto, & Matsumoto, 2007), entre otras.

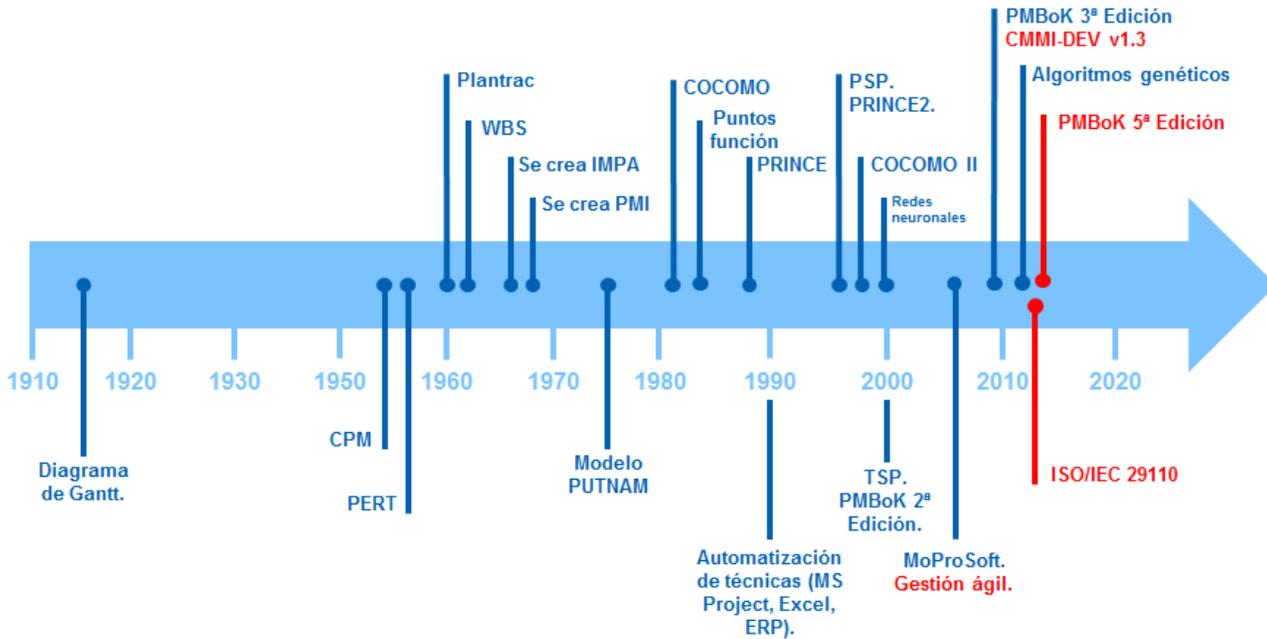
De acuerdo con (Bogado, Dapozo, & García-Martínez, 2011), la incorporación de los métodos de estimación basados en modelos empíricos agregó a la gestión de los proyectos las técnicas heurísticas consideradas “no algorítmicas”, es decir “no deterministas”, porque en éstas, los gestores no sólo consideran la solución, sino la probabilidad de que esa solución sea cierta. De aquí que años más tarde se haya dado la introducción de árboles de decisión, redes neuronales, algoritmos genéticos y otras técnicas de aprendizaje automático a la gestión de los proyectos para construir modelos de predicción de diferentes atributos de un producto de software.

En este sentido, Sabini (2014) argumenta que la estructura común de los diversos modelos para la gestión de los proyectos que se han desarrollado con el paso de los años requiere para su administración de un responsable (el director o jefe de proyectos), por lo que en la industria existe una demanda de habilidades específicas para la gestión, lo cual abre las oportunidades para los directores de proyectos calificados en todas las áreas de negocio, la industria, la academia, y el Gobierno. También es importante mencionar que a partir de todos los avances logrados en la investigación operativa sobre este periodo se ha podido dotar, de alguna manera, al director de proyectos con herramientas sofisticadas de TI para la planificación y comunicación de actividades.

Así pues, el principal logro de la historia de la gestión de los proyectos se relaciona con la combinación, la codificación y la estandarización de todas las técnicas que fueron desarrolladas principalmente para la planificación y el control de los proyectos complejos. Dichas técnicas se difundieron entre asociaciones catalogadas como profesionales (dentro del sector de la ingeniería, la

construcción y el sector militar principalmente), para posteriormente unificarse en cuerpos de conocimiento (Calabrese & Di Nauta, 2014; Sabini, 2014).

A modo de resumen, la Figura 2.1 muestra la línea de tiempo que ha seguido la evolución de la gestión de los proyectos de software. Cabe mencionar que esta línea de tiempo es producto de realizar un análisis exploratorio sobre la evolución que han seguido las técnicas, modelos y organizaciones relacionadas con la gestión de los proyectos de software. En esta figura se han resaltado propuestas relacionadas con los modelos CMMI-DEV v1.3, PMBoK, el estándar ISO/IEC 29110 y el enfoque ágil de gestión, puesto que son las propuestas que predominan en las bases de datos de publicaciones de prestigio sobre el avance de la Ingeniería de Software.



**Figura 2.1.** Evolución de la gestión de los proyectos de software.

Estos antecedentes permiten determinar que, a través del paso del tiempo, es posible identificar tres enfoques importantes en la evolución de la gestión de los proyectos: el *cumplimiento de plazos*, cuyo principal objetivo era el controlar el plazo de ejecución de los proyectos; el *control de costos*, que se basaba en la idea firme de realizar una estimación de presupuesto y efectuar el seguimiento y control de las actividades planeadas; y el enfoque basado en la *calidad*, que se ocupa, como su nombre lo expresa, en mantener la calidad del proceso de gestión (O'Sheedy, Xu, & Sankaran, 2010).

## 2.2. Un poco de historia sobre la gestión en la industria de software

De acuerdo con la información anterior, la gestión de un proyecto no es un concepto que se haya introducido recientemente a nuestra vida cotidiana. A lo largo de los siglos han existido proyectos de construcción a gran escala, incluyendo la elevación de las pirámides egipcias, la construcción de Stonehenge en el Reino Unido, y el establecimiento de una red de carreteras por el Imperio Romano, sólo por citar algunos ejemplos. Dado que los proyectos se hicieron cada vez más grandes y complejos, fue necesario que los gobiernos y los grandes consorcios empresariales comenzaran a introducir el uso de técnicas de gestión de proyectos, las cuales tenían como objetivo principal el controlar las tareas complejas de la ingeniería (e.g., para la construcción, la industria aeroespacial y aeronáutica, y la producción de nuevos y avanzados sistemas de misiles). Es así

como, con el paso del tiempo, surge un importante interés en la gestión de los proyectos de la industria y la academia dando lugar a la elaboración de normas de gestión durante las últimas décadas.

Dentro del contexto de la industria del software, se tiene conocimiento de que en los años 50 no existían metodologías que apoyaran la gestión de este tipo de proyectos, puesto que en esta década no se diferenciaba todavía el software del hardware. En este sentido, no fue hasta los 60's cuando IBM desarrolló el sistema operativo para una nueva línea de computadoras, el OS/360<sup>9</sup>, cuyo desarrollo se consideró como un proyecto de tamaño mediano dentro de un entorno universitario en el cual, vale la pena mencionar, no existía preocupación alguna por el costo o plazos de entrega. Así pues, el desarrollo del OS/360 de IBM, rebasó todo plazo establecido y elevó sus costos de manera considerable, dando como consecuencia la generación del primer gran atraso experimentado por la industria del software (Brooks, 1975). Este resultado fue la consecuencia de no planificar y controlar el proyecto considerando las limitaciones presupuestarias del producto a lanzar. Es importante mencionar que la mayoría de los autores consideran aplicable el término “planificar” (*planning*) a la primera etapa de un proceso de gestión, que incluye a la especificación de las tareas y las relaciones entre ellas y la asignación de recursos a las mismas, considerando así que el término “gestión” (*management*) abarca la totalidad del proceso, es decir a la planificación y al control de las tareas especificadas, sus relaciones, y los recursos asignados. Al introducir el concepto de gestión en el desarrollo del OS/360, con seguridad los resultados hubieran sido diferentes.

De acuerdo con el Instituto para la Gestión de los Proyectos (*Project Management Institute*)<sup>10</sup>, la gestión de los proyectos consiste en aplicar conocimientos, habilidades, herramientas y técnicas en las actividades de los mismos, a fin de satisfacer las necesidades de las partes interesadas (PMI, 2013). Es decir, es el arte de dirigir y coordinar los recursos humanos y materiales a lo largo de un proyecto con el fin de lograr sus objetivos dentro de los límites especificados. Así, el objetivo de la gestión de un proyecto de software consiste en mantener su desarrollo dentro de los objetivos de costo, calidad y duración, mediante la gestión de las interacciones entre la planificación, el control, el desarrollo técnico, y el aseguramiento de la calidad. Es posible explicar la relación entre estos objetivos mediante la analogía del Triángulo de Hierro (Atkinson, 1999), el cual demuestra que en los proyectos de software existen tres variables directamente relacionadas (tiempo, costo y alcance), y otra que permanece estática (calidad). Como su nombre lo indica, la variable *tiempo* indica el tiempo que tardará el proyecto en completarse, la variable *costo* hace referencia a los recursos humanos, materiales, etc., que se dedicarán al proyecto, y la variable *alcance* hace alusión a las características o tareas que se van a realizar dentro del proyecto. En este sentido, para mantener la *calidad* de cualquier proyecto de software es necesario que estas tres variables sean cuantitativamente proporcionales, en otro caso, si por ejemplo aumenta el número de características a desarrollar, se deberán de incorporar más recursos y consecuentemente, el costo del proyecto aumentará. Estas tres variables deben estar presentes en tres de las funciones principales dentro de la gestión de un proyecto de software: *estimación, planificación y monitorización y control* del estado del proyecto.

Sin embargo, es común observar que los modelos de proceso o ciclos de vida tradicionales de la Ingeniería de Software, y empleados frecuentemente dentro de un contexto industrial, definen múltiples actividades que incluyen desde la captura de los requisitos, el análisis de los procesos, la

---

<sup>9</sup> El OS/360 fue un sistema operativo producido por IBM entre 1965 y 1972. Desarrollado en 1964, el lanzamiento del OS/360 estaba dispuesto para 1965 para las versiones más simples y para 1966 para las versiones más complejas, pero no fue hasta 1967 cuando vio la luz oficialmente debido a los retrasos en su entrega (Kruchten & Stafford, 2006).

<sup>10</sup> El *Project Management Institute* (PMI) es una organización internacional sin fines de lucro que asocia a profesionales relacionados con la gestión de proyectos. Desde principios del 2011, esta organización es la más grande del mundo en su rubro, dado que se encuentra integrada por más de 700,000 miembros distribuidos en cerca de 170 países.

gestión de la calidad y los riesgos, hasta el diseño y codificación, la ejecución de pruebas, y el mantenimiento del producto resultante, sin mencionar la planificación del trabajo. En este sentido, una de las actividades más importantes en el desarrollo de un producto de software es la de “gestionar” el proyecto. De acuerdo con (Blomquist, Hällgren, Nilsson, & Söderholm, 2010), la gestión de un proyecto de software dentro de un contexto empresarial debe incluir cinco grandes fases: la planificación, la gestión del personal, la organización, el control, y la dirección. Estas fases son definidas brevemente a continuación:

1. **Planificación del proyecto.** La planificación consiste en establecer un conjunto de actividades de gestión que permitan elegir una relación adecuada de acciones a realizarse dentro del proyecto, además de indicar la manera en que éstas deben llevarse a cabo. Dado lo anterior, es importante especificar los objetivos del proyecto de forma priorizada junto con una serie de estrategias, políticas, planes y procedimientos para alcanzarlos. Así, las estrategias representan objetivos a largo plazo, las políticas son decisiones predeterminadas y ya aprobadas por la organización, los planes se generan como mecanismo de contingencia y control de riesgos, y los procedimientos engloban las acciones necesarias para corregir los posibles desvíos en dichos planes. Como resultado de las actividades de esta fase se genera un plan de proyecto que incluye una estructura de las tareas que se deberán realizar, una estimación del tamaño del proyecto, una estimación de los costos y recursos disponibles y necesarios para cada proyecto, así como la calendarización que especificará hitos y dependencias entre las actividades.
2. **Gestión del personal.** Consiste en conseguir a las personas adecuadas para desarrollar la estructura decidida de tareas incluyendo desde la selección del equipo de trabajo hasta su capacitación, con el objetivo de que pueda cumplir correctamente con las funciones asignadas; es decir, en esta etapa se realiza la selección del personal, su adiestramiento y desarrollo profesional para que las personas puedan encajar en un puesto de trabajo que beneficie a la culminación correcta del proyecto.
3. **Organización del proyecto.** Consiste en el establecimiento de una adecuada estructura organizativa para asignar y terminar las tareas de un proyecto, lo cual implica la definición de las entidades y responsabilidades asignadas a las personas. Para conseguir este objetivo es necesario listar las tareas y/o grupos de tareas y asignarlas de acuerdo a la estructura organizativa establecida.
4. **Control del proyecto:** El control consiste en verificar que el proyecto se desarrolla de acuerdo a lo previsto en los planes generados anteriormente. Para esto, es necesario recoger medidas para un conjunto de parámetros de tal forma que sea posible evaluar el rendimiento del proyecto y verificar la conformidad entre lo planificado y lo ejecutado. Esta fase comprende todas las actividades y tareas que permiten identificar los problemas en los retrasos del proyecto y, sobre todo, en la verificación de los plazos y costos establecidos.
5. **Dirección del proyecto.** Es en esta etapa en donde se acentúan las diferencias entre un líder de proyecto y su equipo de trabajo, puesto que se resaltan aquellos aspectos positivos para contribuir con la motivación de todo el personal de tal forma que se contribuya al valor agregado del proyecto. Aquí la función de liderazgo incluye la guía en el trabajo y la clasificación de las tareas asignadas, la motivación para mejorar el rendimiento de las personas, y conseguir la confianza de los implicados en el proyecto.

En este sentido, en el contexto de la industria de software las tareas anteriormente descritas son apoyadas por herramientas especializadas y técnicas para la gestión que son implementadas a

través del ciclo de vida del proyecto, y su uso correcto afecta directamente el éxito del mismo (Patanakul, Iewwongcharoen, & Milosevic, 2010). De esta manera, las empresas de software intentan reducir el tiempo de desarrollo de sus productos, gestionar sus limitados recursos, manejar la complejidad tecnológica, incrementar la satisfacción de sus clientes, y generar una ventaja competitiva en el mercado mundial. Sin embargo, si el líder o jefe de proyectos no posee el conocimiento técnico adecuado, las herramientas y técnicas no son de mucha ayuda en un entorno empresarial. La gestión de los proyectos de software, por ejemplo, requiere el uso de técnicas orientadas principalmente a la toma de decisiones estratégicas y al uso de tácticas para la gestión de nivel superior (directivo). En este contexto, tales técnicas plantean la creación de un repositorio de variables que midan el progreso en el desarrollo de un proyecto de software, de tal forma que sea posible controlar cualquier resultado no deseado. Así pues, la gestión de los proyectos de software involucra el control de múltiples decisiones a tomar en innumerables situaciones que se pueden presentar durante el desarrollo y mantenimiento de un producto. Por lo tanto, el uso de información cuantitativa y cualitativa contribuye a mejorar el proceso de toma de decisiones (Ferrer, Giráldez, & Ruiz, 2007).

Sin embargo, y de acuerdo con (Turner, Ledwith, & Kelly, 2010), la gestión de un proyecto de software, en el contexto de las pequeñas empresas, ha sido errónea en las actuales circunstancias y necesidades de dinamismo y adaptabilidad de los productos de software. Por tal motivo, ha surgido la necesidad de investigar y desarrollar nuevas técnicas que tienden hacia el rápido desarrollo de aplicaciones y la generación de mayores ventajas competitivas durante la vida útil del producto. En un entorno dinámico e inestable, como un pequeño equipo de trabajo, que tiene como principales protagonistas al cambio y la evolución rápida y continua, la ventaja competitiva se encuentra en aumentar la productividad para satisfacer las necesidades del cliente en el menor tiempo posible para agregar valor al negocio (Mohapatra & Sreejesh, 2014).

En este sentido, las metodologías “tradicionales” por lo regular son eficaces en el desarrollo de proyectos grandes, en relación al control del tiempo y recursos necesarios, y se caracterizan por su rigidez y documentación exhaustiva en cada una de las etapas de desarrollo. Sin embargo se han mostrado inadecuadas para proyectos pequeños desarrollados por las PyMES, donde predominan los entornos altamente cambiantes, y se requiere que el tiempo de desarrollo sea el mínimo y que la calidad sea mantenida a lo largo del proceso y en el producto final. Así pues, surge una alternativa a través de las denominadas metodologías “ágiles”, las cuales están orientadas principalmente a equipos pequeños puesto que ofrecen múltiples beneficios como su simplicidad y la calidad en el producto. En 2001 surge el término “ágil” como consecuencia de la reunión de un grupo de 17 expertos de la industria de software y creadores o impulsores de las metodologías ágiles, el objetivo fue el esbozar los valores y principios para el desarrollar rápidamente el software y además que se pudiera responder a los cambios a lo largo del desarrollo de todo el proyecto. Al cabo de esta reunión se alcanzaron dos cosas importantes, como son:

- El “Manifiesto ágil para el desarrollo de software”, el cual describe la filosofía de las metodologías ágiles, y
- La organización sin fines de lucro “The Agile Alliance”, la cual está dedicada a promover los conceptos relacionados con el desarrollo ágil de software, así como ayudar a organizaciones y empresas a adoptar estas metodologías (Canós, Letelier, & Penadés, 2003).

Así, los principales objetivos de la gestión ágil dentro de un contexto general son: el valor, la reducción del tiempo de desarrollo, agilidad y fiabilidad (O’Sheedy & Sankaran, 2013). Por lo que es importante la descripción de cada uno de estos:

- El *valor* es proporcional a la respuesta en cuanto a su innovación (novedoso para el mercado en que se esté ofertando) y flexibilidad (la capacidad de evolucionar o de adaptarse para mantener su posición en el mercado).
- La *reducción del tiempo de desarrollo* es una de las fortalezas más importantes hablando de competitividad en determinados sectores y el software no es la excepción, así que cuenta con las estrategias necesarias para la gestión ágil de tal manera que la producción de resultados se de en menor tiempo.
- La *agilidad* está caracterizada por adaptarse a un contexto cambiante donde los requisitos evolucionan continuamente.
- Los *resultados fiables* tienen como objetivo principal elevar el valor del resultado y agilizar el tiempo de salida al mercado.

Ahora bien, si la metodología de desarrollo de un proyecto es “ágil”, entonces la gestión del proyecto debe realizarse bajo el mismo enfoque. En este sentido, cabe mencionar que las fases de una metodología de este tipo son: planificación, ejecución y entrega, las cuales conforman un ciclo de vida iterativo e incremental, permitiendo así que el producto final sea entregado parcialmente en varios ciclos de desarrollo. Las metodologías ágiles resultan muy atractivas para las pequeñas y medianas empresas debido a que éstas les permiten tener procesos organizados, repetibles y mejorables sin una alta inversión de presupuesto y de tiempo en su implementación. Sin embargo, uno de los principales problemas con la utilización de las metodologías ágiles es la falta de una gestión efectiva, la monitorización inadecuada, la falta de generación de evidencias, el exceso de costos, la falta de precisión en los tiempos de entrega y los compromisos con el cliente (Rodríguez, Berdun, Soria, Amandi, & Campo, 2014; Mitre-Hernández, Ortega-Martínez, & Lemus-Olalde, 2014). Pero el problema principal es que una gestión ágil es de mucha utilidad en equipos pequeños bien preparados, aspecto que, como se mencionó anteriormente es uno de los principales problemas en las PyMES mexicanas. En este sentido, es conveniente que estos pequeños equipos aprendan primero a gestionar los proyectos, y después tomar experiencia poco a poco para implementar un enfoque ágil.

Así pues, con el fin de obtener una definición precisa de los dos procesos elementales que forman el área de gestión de proyectos, las siguientes secciones proporcionan una explicación detallada que establece la base de las actividades que cualquier empresa debería seguir para llevar un proyecto a buen fin.

### 2.3. La gestión de los proyectos de software

La gestión de los proyectos es un proceso fundamental para las empresas desarrolladoras de software puesto que, como se mencionó en el Capítulo 1 de esta tesis, de su ejecución puede depender el éxito o fracaso de un proyecto.

En este sentido, de acuerdo con el PMBoK la gestión de los proyectos es la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades relacionadas con la planificación, seguimiento y control de un proyecto, con el objetivo de cumplir con los requisitos del mismo a través de la aplicación e integración adecuada de cinco grupos de procesos: Inicio, Planificación, Monitoreo y Control, y Cierre (PMI, 2013). MoProSoft, por ejemplo, describe que el propósito de la gestión de proyectos es asegurar que los proyectos contribuyan al cumplimiento de los objetivos y estrategias de la organización, enfocándose en los proyectos externos, internos y oportunidades de negocio (generación y cierre, presentación de la propuesta y la firma del contrato).

Este proceso comprende las actividades para la Planificación, Realización, y Evaluación y Control (Oktaba, Esquivel, Su Ramos, Martínez, Quintanilla, Ruvalcaba, & Fernández, 2005).

De manera más formal, el CMMI-DEV V 1.3 (CMMI, 2010) define áreas de proceso<sup>11</sup> para la correcta ejecución de la gestión de proyectos, cubriendo así un amplio espectro de necesidades:

1. Gestión Integrada del Proyecto (IPM, por sus siglas en inglés),
2. Planificación del Proyecto (PP, por sus siglas en inglés),
3. Monitorización y Control del Proyecto (PMC, por sus siglas en inglés),
4. Gestión Cuantitativa del Proyecto (QPM, por sus siglas en inglés),
5. Gestión de Requisitos (REQM, por sus siglas en inglés),
6. Gestión de Riesgos (RSKM, por sus siglas en inglés),
7. Gestión de Acuerdos con Proveedores (SAM, por sus siglas en inglés)

Estas áreas de proceso para la gestión de proyectos son abordadas desde dos enfoques que dependen del nivel de madurez que pretenda alcanzar la organización. Es decir, las áreas básicas de proceso están orientadas a los Niveles 1 y 2 y contemplan a aquellas actividades relacionadas con el establecimiento y mantenimiento de un plan de proyecto y de los compromisos (PP), la monitorización de proceso del proyecto frente a dicho plan y la toma de acciones correctivas (PMC), y la gestión de acuerdo con los proveedores (SAM); y las áreas avanzadas de proceso están orientadas a los Niveles 3, 4, y 5 e involucran a las actividades que están enfocadas a establecer un proceso definido que se adapte a partir del conjunto de procesos estándar de la organización, establecer el entorno de trabajo de la organización, coordinar y colaborar con las partes interesadas relevantes (REQM), gestionar los riesgos (RSKM), crear y sostener a los equipos integrados para la conducción de los proyectos (IPM), y gestionar de forma cuantitativa el proceso definido del proyecto (QPM).

En este sentido, es lógico pensar que una empresa que recién comienza, que no tiene experiencia gestionando los proyectos, y que no dispone de suficientes recursos humanos y financieros (es decir, el contexto actual de una PyMES) debería empezar por seguir las áreas básicas de proceso. Además y en correspondencia con la definición anterior por el CMMI-DEV V 1.3, la planificación y control de los proyectos son las tareas básicas para comenzar a introducir la gestión de los proyectos en una empresa. Así, la Figura 2.2 muestra la relación existente entre las áreas de procesos de planificación y monitorización y control de un proyecto y su interacción con SAM. Esta figura muestra que PP desarrolla un conjunto de actividades para obtener un plan de proyecto que involucra correctamente a las partes interesadas, quienes deben comprometer con su ejecución y mantenimiento. PP inicia con los requisitos que componen tanto al producto como al proyecto, y las actividades de gestión y desarrollo que deberán realizarse en el proyecto. Por otro lado, PMC incluye actividades para la monitorización del proyecto y la toma de acciones correctivas en caso de presentarse desviaciones frente al plan generado por PP. De igual manera, se controla la frecuencia de las revisiones del progreso respecto al plan y las mediciones usadas en la monitorización. De esta manera, PMC permite determinar el estado real del proyecto y así, determinar cuándo se desvía el proyecto de forma significativa de los valores esperados establecidos en el plan, y tomar acciones correctivas según sea conveniente.

---

<sup>11</sup> Un “área de proceso” es un grupo de prácticas relacionadas que, cuando se implementan de manera conjunta, satisfacen un grupo de objetivos (CMMI, 2010).

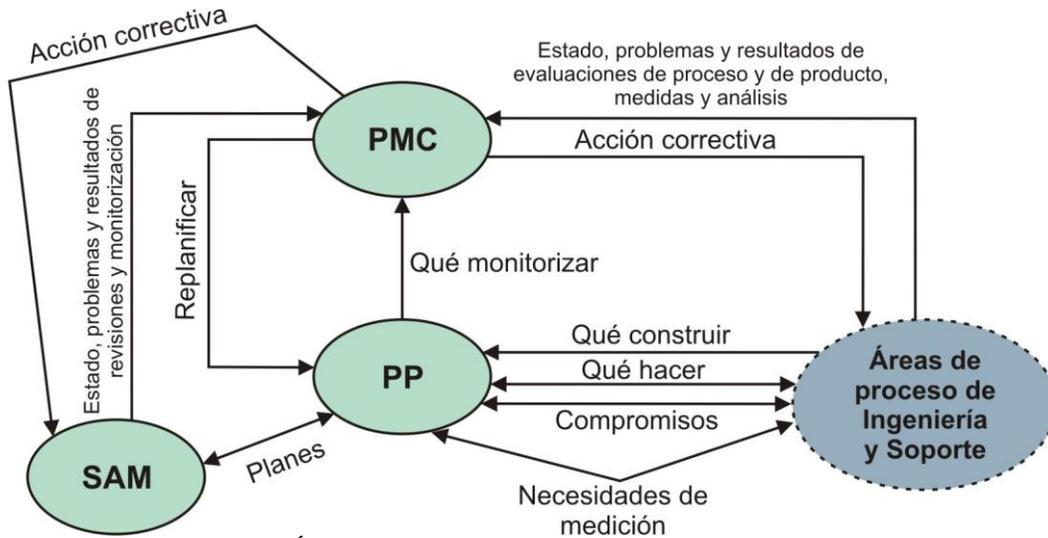


Figura 2.2. Áreas básicas para la gestión de proyectos (CMMI, 2010).

Así pues, la gestión de proyectos bajo este esquema impacta en el trabajo de las organizaciones, los sistemas y las personas, y el jefe de proyectos juega un papel importante en su ejecución, siendo éste el responsable de la planificación, supervisión de las tareas y aseguramiento de que el trabajo sea realizado de acuerdo a los estándares y dentro del tiempo y presupuesto establecidos.

### 2.3.1. La planificación de los proyectos

Para que las PyMES puedan llevar a buen fin un proyecto de software, es necesario que primero se definan cada una de las actividades o tareas a realizar, de tal modo que también sea posible definir a los responsables de ejecutarlas. Lo anterior implica el vislumbrar una idea concreta de qué va a desarrollarse y quién lo desarrollará para poder establecer así un marco de tiempo, un costo aproximado, el esfuerzo necesario (en términos de plantilla) para cumplir con dicho tiempo, los recursos (tanto financieros como de infraestructura y de habilidades), los riesgos y el alcance, e identificar el contexto del proyecto. Es decir, es necesario realizar una *planificación del proyecto*.

En este sentido, la planificación de los proyectos consta de una serie de actividades para establecer el alcance total del proyecto, definir y refinar los objetivos del mismo, y desarrollar una línea de acción que permita cubrir las metas y los objetivos definidos. Una de las metas principales de la planificación consiste en desarrollar un plan para la gestión del proyecto, y demás planes auxiliares para cumplirlo. Así, dependiendo del tipo de proyecto es posible determinar, de manera iterativa, la realimentación sobre el análisis del plan original, lo cual puede llevar a considerar la re-planificación si los desvíos, obtenidos como resultado del análisis, resultaran significativos. Como consecuencia, será necesario establecer estrategias y tácticas, líneas de acción, acciones correctivas o la ruta para completar con éxito el proyecto o fase del mismo, contribuyendo así al logro de las metas y objetivos planteados.

Lo anterior demuestra que la planificación es un proceso que es de suma importancia para el contexto de cualquier proyecto. En el PMBoK, por ejemplo, se establece que este proceso debe realizarse a través de un conjunto de actividades, asignando los recursos necesarios para establecer el alcance del proyecto, y considerando las guías y criterios para adaptar el conjunto de procesos y procedimientos estándar de la organización, con el fin de satisfacer las necesidades específicas del proyecto. Dicho conjunto de procesos y procedimientos estándar de la organización incluye a las políticas (e.g., políticas de recursos humanos, políticas de seguridad y salud, políticas de ética, y políticas de gestión de proyectos), ciclos de vida del producto y del proyecto, políticas y

procedimientos de calidad (e.g., auditorías de procesos, objetivos de mejora, listas de verificación y definiciones estandarizadas de procesos para su uso en la organización), y plantillas (e.g., plantillas de registro de riesgos, de estructura de desglose del trabajo, de diagramas de red del cronograma del proyecto, y de contratos) (PMI, 2013).

De manera más específica, de acuerdo con el CMMI-DEV v1.3 una pequeña empresa puede establecer un proceso de planificación que conste de las siguientes actividades:

1. Establecer estimaciones para la planificación,
2. Desarrollar un plan de proyecto y,
3. Obtener el compromiso con el plan.

La Figura 2.3 muestra que el proceso de planificación inicia con los requisitos establecidos por las partes interesadas para desarrollar el proyecto. Durante el establecimiento de las estimaciones pertinentes es necesario definir el alcance del proyecto, obtener estimaciones reales de los productos a obtener y de las tareas a realizar, y establecer el ciclo de vida del proyecto con el objetivo de determinar las estimaciones de esfuerzo y de costo sobre el producto final. Una vez que las estimaciones preliminares fueron obtenidas, es necesario definir un plan de proyecto en términos de presupuesto y calendario, recursos actuales (financieros y de infraestructura) para comenzar el proyecto, habilidades con las que se cuenta para realizar las actividades marcadas por el plan (en caso de no contar con alguna se deberá contemplar la subcontratación), y el nivel de participación de todos los interesados. Adicionalmente, el líder del equipo (llamando también jefe de proyectos) debe obtener planes adicionales para la identificación y análisis de los riesgos y la gestión de los datos (es decir, dónde será guardada toda la información del proyecto, quién tendrá acceso a ella, qué mecanismos de seguridad y comunicación serán establecidos, y demás). Finalmente, el plan del proyecto es la base fundamental para iniciar el proyecto y realizar el control de las actividades y compromisos del proyecto establecidas inicialmente. Así, es necesario revisar todos los planes que afectan al proyecto, reconciliar los niveles de trabajo y de recursos con la alta dirección, y obtener un compromiso con el plan.

En general, la parte más complicada en la formulación de un proyecto dependerá de la naturaleza de éste. Sin embargo entre los aspectos comunes a todos los proyectos destacan la resolución de incertidumbres y la integración del plan de forma que sea coherente y eficiente. En este sentido, esta complicación afecta también al proceso denotado en las actividades de la Figura 2.3 puesto que el nivel de detalle del plan está influenciado por la metodología de gestión de proyectos elegida, el grado de madurez, la capacidad para trabajar en equipo y la experiencia de los participantes en la realización de proyectos. Esto determinará si es necesario elaborar una planificación muy detallada en la que se especifiquen las tareas y fechas en las que cada miembro del equipo deberá dedicar un número de horas determinadas (típico de metodologías predictivas) o si, por lo contrario, se opta por una planificación más flexible donde se identifican los objetivos del proyecto y se permite que los miembros del equipo se organicen con mucha más autonomía en cuanto al reparto de tareas, el orden de ejecución y la dedicación necesaria (habitual con metodologías ágiles).

Por último, es lógico que cualquier plan que sea creado para realizar cualquier proyecto deba ser controlado, y la Ingeniería de Software no es la excepción. La planificación de proyectos está estrechamente ligada al proceso de *monitorización y control del proyecto*, puesto que no se puede afirmar que un plan esté siguiéndose exitosamente si no existe evidencia que demuestre, con datos cuantitativos, que no habrá desvíos en las estimaciones preliminares que dieron origen al plan.

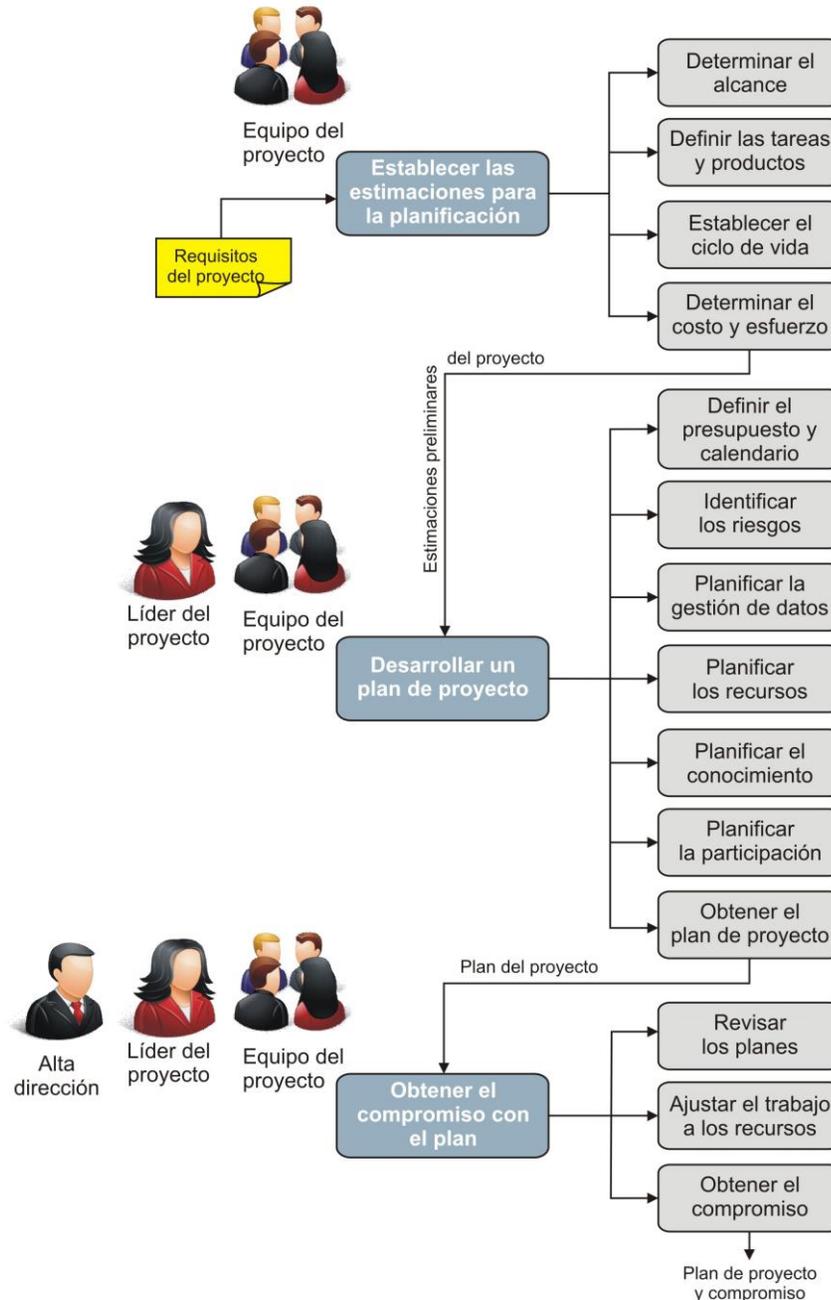


Figura 2.3. Síntesis del proceso de planificación de proyectos (CMMI, 2010).

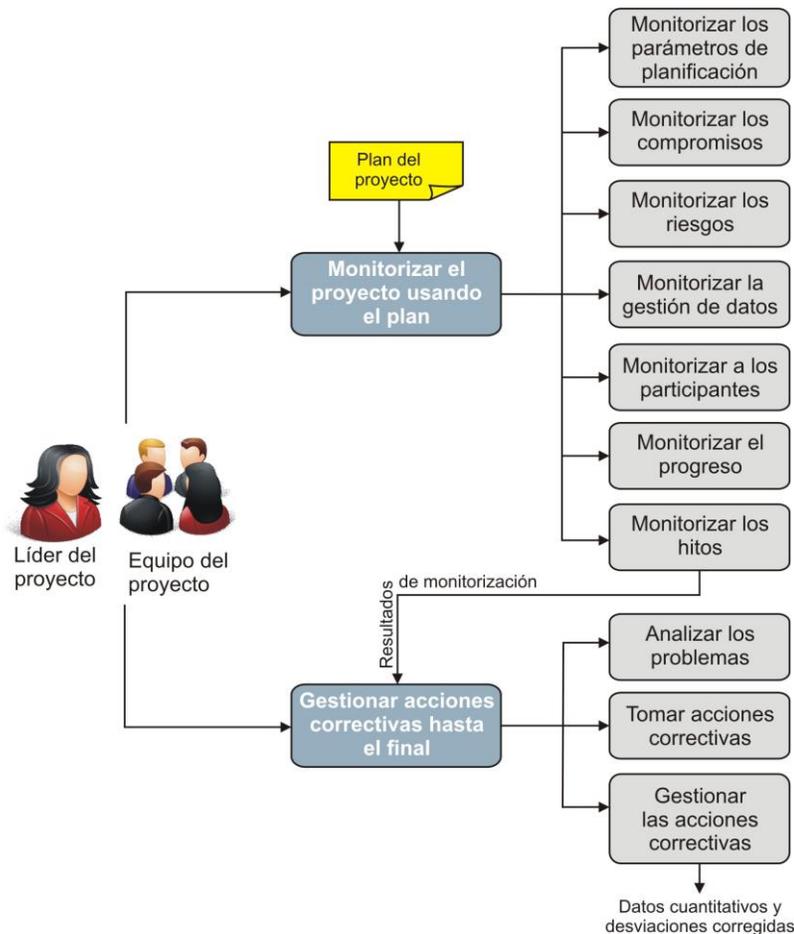
### 2.3.2. La monitorización y control de los proyectos

El proceso de monitorización y control de los proyectos tiene como base principal los planes descritos durante la planificación del proyecto, y tiene como propósito darle seguimiento a todas las actividades y compromisos definidos en éstos. Es decir, establece mecanismos de seguimiento y análisis para determinar el progreso del proyecto, de tal forma que sea posible definir las acciones preventivas o correctivas apropiadas con el único propósito de cumplir con los compromisos establecidos con el cliente. A través de este seguimiento es posible establecer una base comparativa entre lo planeado y lo real en relación al avance con los productos de trabajo y las tareas, el esfuerzo, el costo y el calendario. Esto permitirá que el líder del proyecto evalúe el estado real del progreso del proyecto con respecto a las estimaciones iniciales y analice la situación para definir

acciones pertinentes (preventivas o correctivas dependiendo del desvío y que ocasionarían, en el peor de los casos, que no se esté cumpliendo con las metas u objetivos planteados por el proyecto) que el equipo del proyecto deberá realizar. Así, el líder del proyecto puede realizar actualizaciones al plan e incluso se puede dar el caso de una re-planificación la cual implica desde la modificación del plan original, hasta definir y plantear nuevos acuerdos. De manera formal, de acuerdo con el CMMI-DEV v1.3 una pequeña empresa puede establecer un proceso de monitorización y control a través de las siguientes actividades:

1. La monitorización del proyecto usando el plan, y
2. La gestión de las acciones correctivas hasta el final.

La Figura 2.4 muestra que estas dos actividades dependen, en primera instancia, de un alto grado de supervisión de todas las actividades, compromisos, productos comprometidos y responsabilidades establecidas en el plan de proyecto. Esta monitorización abarca a los parámetros de la planificación (estimaciones frente a lo real), los compromisos de los participantes, los riesgos del proyecto, la gestión de los datos y la gestión de la participación del equipo del proyecto y demás partes interesadas. Como se ha mencionado anteriormente, si este seguimiento riguroso identificará desvíos importantes será necesario establecer y gestionar acciones correctivas hasta su cierre.



**Figura 2.4.** Síntesis del proceso de monitorización y control de proyectos (CMMI, 2010).

Es verdad que en el contexto de los pequeños equipos formados dentro de las PyMES, los procesos de *planificación del proyecto* y *monitorización y control del proyecto* representan una

fortaleza para cumplir con lo establecido en sus planes de trabajo y podrían llevar a buen término un proyecto cumpliendo no sólo con los compromisos establecidos con el cliente, sino también con los compromisos de calidad, rendimiento y productividad. Es decir, tal y como afirma el PMBoK “el monitoreo continuo proporcionará al equipo del proyecto el conocimiento sobre la salud del mismo y permitirá identificar las áreas que requieren más atención” (PMI, 2013). Sin embargo, llevar la implantación de estos dos procesos no es tarea fácil, mucho más en el contexto de las PyMES. En este sentido, es necesario realizar ajustes a los procesos para subsanar las deficiencias técnicas que las PyMES padecen desde su creación. Así pues, es necesario analizar los avances en este contexto a través de una revisión de literatura que permita identificar los métodos y enfoques que la comunidad internacional ha desarrollado para mejorar ambos procesos en el entorno de trabajo de un pequeño equipo formado dentro de una pequeña empresa.

## 2.4. Estudio del arte

Como se mencionó anteriormente, dentro del ámbito del desarrollo de software a nivel mundial las PyMES ocupan un porcentaje importante de la totalidad de empresas. Es por lo anterior que a lo largo del proceso de investigación sobre la Ingeniería de Software, se ha hecho énfasis en apoyar a este tipo de empresas a través de la implementación de modelos formales relacionados con las áreas del desarrollo de software.

De acuerdo con Richardson y Gresse von Wangenheim (2007), las pequeñas empresas de desarrollo de software –financiadas de manera independiente y conformadas por menos de 50 empleados– son fundamentales para el crecimiento de muchas economías nacionales. Por ejemplo, en Estados Unidos de Norteamérica, Brasil, México, Canadá, China, India, Finlandia, Irlanda, Hungría, y muchos otros países, las pequeñas compañías representan más del 85% de todas las empresas dedicadas al desarrollo de software. Sin embargo, para que estas empresas puedan persistir y crecer necesitan ser eficientes. No obstante, las pequeñas empresas suelen creer que la adopción de las “mejores prácticas” solamente se logra a través de un proceso caro y engorroso que se orienta principalmente a las grandes empresas. Desde luego que tanto las grandes empresas desarrolladoras de software como las pequeñas enfrentan diferentes retos relacionados con la Ingeniería de Software, como son: gestionar y mejorar sus procesos, afrontar los rápidos avances de la tecnología, mantener sus productos en el mercado, operar en un entorno global de software, y sostenerse activas a través del crecimiento del mercado. Pero también es verdad que lo que se crea para una empresa grande no necesariamente puede ser utilizado directamente en una pequeña; para esto se requieren diversos enfoques que se ajusten al contexto de cada organización, a los modelos de negocio, a los objetivos específicos, al nicho de mercado, al tamaño, la disponibilidad de recursos (tanto financieros como humanos), a la capacidad de gestión, y a las diferencias organizacionales, entre otras cosas.

Sin embargo, si bien cuesta trabajo crearlo, las pequeñas empresas poseen ventajas que las grandes no tienen, empezando por su principal ventaja competitiva: la flexibilidad. Es decir, su estructura organizativa flexible fomenta el espíritu empresarial y la innovación. Además, las pequeñas empresas por lo regular solamente se centran en un nicho de mercado (e.g., desarrollo de componentes para productos de otras empresas, desarrollo de productos innovadores, provisión de servicios o mantenimiento a los productos que ellos u otros producen), lo cual les permite generar experiencia y ser más competitivas frente al resto de empresas. En este sentido, cabe resaltar que en los últimos años se está poniendo mayor énfasis a mejorar la forma de trabajo de las empresas, con menos de 25 empleados, diferenciándolas de las empresas medianas y considerando sus características particulares. Así pues, es un error pensar que los modelos, metodologías y/o estándares existentes se puedan aplicar, en versiones reducidas, en este tipo de empresas puesto que

poseen características muy particulares y por lo tanto, necesitan de propuestas que se ajusten al contexto en el que se desarrollan. Es por esto que durante los últimos 10 años tanto la industria como la comunidad internacional relacionada con la Ingeniería de Software han centrado sus esfuerzos en el desarrollo de soluciones que buscan mejorar la calidad de los procesos de las pequeñas empresas y, en consecuencia, la calidad de sus productos o servicios.

Es así que con el objetivo de explorar cómo ha sido abordado el problema planteado al inicio de esta tesis (la gestión de los proyectos en sí), las siguientes secciones de este documento proporcionan una visión general sobre las investigaciones más relevantes que han sido creadas para el contexto de las pequeñas empresas. Este análisis sobre el estado de la práctica permitirá más adelante realizar una comparativa empírica que sitúe la solución propuesta en esta tesis en el contexto actual de la investigación.

## **2.4.1. Un programa de mejora de la gestión de proyectos de acuerdo al estándar ISO/IEC 29110 y PMBoK®**

### **2.4.1.1. Objetivo**

La investigación de Mesquida y Mas (2014) presenta los resultados y las lecciones aprendidas en la definición de un conjunto estandarizado de procesos y procedimientos, con el objetivo de mejorar la gestión de los proyectos en cuatro pequeñas empresas desarrolladoras de software. Para cumplir este propósito, esta investigación introdujo el desarrollo de una biblioteca de activos de procesos (PAL, por sus siglas en inglés) para inculcar el uso de las buenas prácticas dentro de las organizaciones participantes. En este sentido, la guía del PMBoK fue utilizada para complementar al estándar ISO/IEC 29110-5-1-2 (ISO/IEC, 2011) con el fin de proporcionar el conocimiento detallado sobre las entradas, salidas y mejores técnicas de la gestión de proyectos.

### **2.4.1.2. Descripción**

Con el fin de optimizar el uso de los recursos dentro de las PyMES y reducir los costos que implica la consultoría para mejorar los procesos de las mismas, en 2013 se creó un programa de mejora que involucró a cuatro empresas de este tipo, las cuales pertenecían al TurisTEC (Cluster de Compañías e Instituciones dedicadas a la Producción e Implementación de Soluciones Tecnológicas para la Industria del Turismo). En este sentido, esta investigación propuso la incorporación de un conjunto de “mejores prácticas”<sup>12</sup> a los proyectos, en base a la experiencia previa de los autores obtenida durante la implementación de programas de mejora en PyMES (e.g., Amengual & Mas, 2007; Mas, Mesquida, & Fluxà, 2012; Mas, Fluxà, Amengual, 2012). Así, dada la naturaleza de estas organizaciones, se sugirió la adopción de la norma internacional ISO/IEC 29110 para la mejora de los procesos de software, puesto que fue creada específicamente para las microempresas y sus necesidades (O’Connor & Laporte, 2012; Boucher, Perrouin, Deprez, & Heymans, 2012). El objetivo principal de este esfuerzo se enfocó únicamente en el establecimiento de los procesos para la gestión de proyectos, a través de la adición de mejores prácticas, y la mejora y agregación de nuevos activos de proceso<sup>13</sup> (e.g., políticas, medidas, descripciones de procesos y herramientas de soporte a la implementación de los procesos).

---

<sup>12</sup> El concepto de “mejores prácticas” se relaciona con todas aquellas actividades provenientes de metodologías, modelos y/o herramientas que mejoran consistentemente la productividad y calidad de los productos de software cuando son implementadas, y al aplicarlas se tiene un beneficio directo en el proyecto. Así, una actividad es considerada eficiente y agrupada con otras cuando ésta ha sido utilizada ampliamente en diferentes entornos industriales y se han reportado resultados satisfactorios con su uso (Walker, 2003).

<sup>13</sup> De acuerdo con el CMMI-DEV v1.3, un activo de proceso es cualquier cosa que la organización considera útil para alcanzar las metas en un área de proceso (CMMI, 2010). El término activos de proceso se utiliza para indicar que estos

La implementación del programa de mejora realizó, como primer paso, el análisis (o mapeo) de la relación entre las tareas del estándar ISO/IEC 29110-5-1-2 enfocadas al proceso de *Gestión del Proyecto* (quedando fuera el proceso de *Implementación de Software*), y las áreas de conocimiento de la guía del PMBoK (concretamente la *Gestión de la Integración del Proyecto*, *Gestión del Alcance del Proyecto*, *Gestión del Tiempo del Proyecto*, y *Gestión del Costo del Proyecto*), con el objetivo de mejorar el conocimiento y facilitar el desarrollo de los activos de proceso de estas organizaciones. Este análisis permitió determinar que era necesario considerar cierta información de las áreas de conocimiento del PMBoK antes de crear una solución para las pequeñas empresas, por ejemplo:

- *Gestión de la Integración del Proyecto*. Esta área incluye a los procesos y actividades de gestión, es decir: constituir formalmente un proyecto, planificarlo, monitorearlo, controlarlo, y dirigirlo al cierre. La *Gestión de la Integración* es un área transversal y fundamental, independientemente de cuál sea el nivel de gestión aplicada. Por lo tanto, y como era de esperarse, el estándar ISO/IEC 29110-5-1-2 incluye una referencia a tales actividades.
- *Gestión del Alcance del Proyecto*, *Gestión del Tiempo del Proyecto*, y *Gestión del Costo del Proyecto*. Tampoco es de extrañar que la norma ISO/IEC 29110-5-1-2 considere los procesos relacionados con esta triple restricción por lo que es necesario definir el trabajo necesario para completar el proyecto, subdividirlo en componentes más pequeños (actividades del proyecto), desarrollar el calendario, y estimar tanto los recursos y costos necesarios para llevar a cabo el proyecto.

Sin embargo, dado que existen otras disparidades entre el estándar ISO/IEC 29110 y la guía del PMBoK, fue necesario realizar algunos ajustes. Por ejemplo, el estándar incluye la identificación de los roles, responsabilidades y habilidades requeridas de los perfiles del equipo del proyecto y la selección apropiada de cada perfil. Sin embargo, el estándar no cubre otras tareas de la *Gestión de Recursos Humanos*, como la capacitación y el control del desempeño de los miembros del equipo. Además, el estándar considera la identificación de los riesgos que pudieran afectar al proyecto y su control durante su ejecución, pero la aborda de forma general sin mencionar específicamente las tareas relacionadas con el análisis cuantitativo y cualitativo de los riesgos, a diferencia de la *Gestión de Riesgos del Proyecto* de la guía del PMBoK. En este sentido, se determinó que las mejores prácticas para la gestión de proyectos de PMBoK que no son consideradas por el estándar ISO/IEC 29110-5-1-2 son las siguientes:

- *Gestión de la Calidad del Proyecto*. Si bien ISO/IEC 29110-5-1-2 considera como aspectos del aseguramiento de la calidad a la verificación, validación y revisión de los productos obtenidos, el área de *Gestión de Calidad del Proyecto* de PMBoK se refiere a los procesos y actividades que establecen las responsabilidades, objetivos y políticas de calidad para la mejora continua de los procesos ejecutados durante todo el proyecto.
- *Gestión de las Comunicaciones del Proyecto*. El estándar no incluye ni la identificación de las necesidades de información de las partes interesadas (*stakeholders*), ni la definición del plan de comunicación para todos los interesados en el proyecto.
- *Gestión de las Adquisiciones del Proyecto*. El estándar no cubre ninguno de los procesos relacionados con la compra o adquisición de productos, servicios, o resultados que no deban ser obtenidos por el equipo del proyecto.

---

artefactos se desarrollan o adquieren para cumplir los objetivos de negocio de la organización y representan inversiones que se espera proporcionen valor empresarial actual y futuro.

- *Gestión de las Partes Interesadas en el Proyecto*. El estándar no cubre los aspectos relacionados con la identificación de las personas u organizaciones (*stakeholders*) afectadas por el proyecto o con la documentación de la información relevante relacionada a sus intereses, participación, e impacto en el éxito del proyecto.

Dado que es difícil que las pequeñas empresas comprendan correctamente estas áreas de proceso y sus activos, se introdujo la idea de crear una PAL<sup>14</sup> para la gestión de los proyectos con el objetivo de agilizar la adopción del nuevo conocimiento. Esta PAL pretendió apoyar el correcto desempeño de las tareas como resultado del mapeo entre las entradas y productos de salida propuestos por el proceso de *Gestión del Proyecto* de ISO/IEC 29110-5-1-2, y las entradas, técnicas, y salidas propuestas por los procesos seleccionados de la guía del PMBoK. Así, los activos desarrollados fueron clasificados en cuatro grupos que corresponden a las cuatro actividades del proceso de *Gestión del Proyecto* del estándar ISO/IEC 29110-5-1-2: planificación del proyecto, ejecución del plan del proyecto, evaluación y control del proyecto, y cierre del proyecto. En este sentido, la Tabla 2 resume a los activos que fueron usados en la fase de planificación del proyecto.

**Tabla 2.** Activos de la planificación del proyecto (Mesquida & Mas, 2014).

Activos	Tareas del proceso de Gestión del Proyecto de ISO/IEC 29110-5-1-2	Procesos de PMBoK
Acta constitutiva del proyecto	PM.1.1 Revisar la <i>Declaración de Trabajo</i> .	4.1 Desarrollar el acta constitutiva del proyecto 5.1 Planificar la gestión del alcance
WBS (estructura desglosada de trabajo)	PM.1.3 Identificar las <i>Tareas</i> específicas a realizar para producir los <i>Entregables</i> y los <i>Componentes del Software</i> identificados en la <i>Declaración de Trabajo</i> . Incluir <i>Tareas</i> en el proceso de <i>Implementación de Software</i> junto con la verificación, validación y revisiones con el <i>Cliente</i> y <i>Equipo de Trabajo</i> para asegurar la calidad de los <i>Productos de Trabajo</i> . Identificar las <i>Tareas</i> para cumplir con las <i>Instrucciones de Entrega</i> . Documentar las <i>Tareas</i> .	5.4 Crear la WBS 6.1 Planificar la gestión del calendario
Calendario	PM.1.4 Establecer la <i>Duración Estimada</i> para realizar cada tarea. PM.1.7 Asignar las fechas estimadas de inicio y terminación para cada una de las <i>Tareas</i> con el fin de crear el <i>Calendario de las Tareas del Proyecto</i> , teniendo en cuenta los recursos asignados, la secuencia, y la dependencia de las <i>Tareas</i> . PM.1.8 Calcular y documentar el <i>Esfuerzo</i> y <i>Costo Estimados</i> para el proyecto.	6.2 Definir las actividades 6.3 Secuenciar las actividades 6.5 Estimar la duración de las actividades 6.6 Desarrollar el calendario
RBS (estructura desglosada de recursos)	PM.1.5 Identificar y documentar los <i>Recursos</i> : humanos, materiales, de equipamiento y herramientas, estándares, incluyendo la capacitación requerida para que el <i>Equipo de Trabajo</i> realice el proyecto. Incluir en el calendario las fechas en que necesitarán los <i>Recursos</i> y la capacitación. PM.1.6 Establecer la <i>Composición del Equipo de Trabajo</i> asignando los roles y responsabilidades de acuerdo a los <i>Recursos</i> .	6.4 Estimar los recursos de las actividades 9.1 Planificar la gestión del recurso humano 9.2 Formar al equipo del proyecto
Presupuesto	PM.1.8 Calcular y documentar el <i>Esfuerzo</i> y <i>Costo Estimados</i> para el proyecto.	7.1 Planificar la gestión de costos 7.2 Estimar los costos
Plan del	PM1.11 Generar el <i>Plan del Proyecto</i> integrando los	7.3 Determinar el presupuesto

<sup>14</sup> Una PAL es un repositorio organizado, indexado, y comprensible de activos de proceso que es de fácil accesibilidad por cualquiera que necesite información sobre el proceso en forma de guías, ejemplos, datos, plantillas, u otros materiales de apoyo (García, de Amescua, Sánchez, & Bermón, 2011).

proyecto Mapa inicial	<p>elementos previamente identificados y documentados.</p> <p>PM.1.12 Incluir la <i>Descripción del Producto, Alcance, Objetivos, y Entregables</i> en el <i>Plan del Proyecto</i>.</p> <p>PM.1.9 Identificar y documentar los riesgos que pueden afectar al proyecto.</p> <p>PM.1.10 Documentar la <i>Estrategia de Control de Versiones</i> en el <i>Plan del Proyecto</i>.</p> <p>PM.1.13 Verificar y obtener la aprobación del <i>Plan del Proyecto</i>. Verificar que todos los elementos del <i>Plan del Proyecto</i> sean viables y consistentes. Los resultados encontrados se documentan en los <i>Resultados de la Verificación</i>, y las correcciones se realizan hasta que el documento sea aprobado por el jefe de proyectos.</p> <p>PM.1.14 Revisar y aceptar el <i>Plan del Proyecto</i>. El cliente revisa y acepta el <i>Plan del Proyecto</i>, asegurándose de que sus elementos coincidan con la <i>Declaración de Trabajo</i>.</p> <p>PM.1.15 Establecer el <i>Repositorio del Proyecto</i> mediante la <i>Estrategia de Control de Versiones</i>.</p>	<p>4.2 Desarrollar el plan para la gestión del proyecto</p> <p>5.2 Recoger los requisitos</p> <p>5.3 Definir el alcance</p> <p>11.2 Identificar los riesgos</p>
--------------------------	---	---

De manera similar, la Tabla 3 resume los activos que fueron definidos para implementar el plan documentado del proyecto.

**Tabla 3.** Activos de la ejecución del plan del proyecto (Mesquida & Mas, 2014).

Activos	Tareas del proceso de Gestión del Proyecto de ISO/IEC 29110-5-1-2	Procesos de PMBoK
Registro del estado del progreso	PM.2.1 Monitorear la ejecución del <i>Plan del Proyecto</i> y registrar los datos reales en el <i>Registro del Estado del Progreso</i> .	4.4 Monitorear y controlar el trabajo del proyecto 4.5 Realizar la gestión integrada de cambios
Petición de cambio	PM.2.2 Analizar y evaluar las <i>Peticiones de Cambio</i> para costo, calendario, e impacto técnico. La <i>Petición de Cambio</i> puede ser iniciada de forma externa por el cliente o interna por el equipo de trabajo. Actualizar el <i>Plan del Proyecto</i> , si el cambio aceptado no afecta a los acuerdos con cliente. La <i>Petición de Cambio</i> , que afecta a tales acuerdos, debe ser negociada por ambas partes (ver PM.2.4).	
Minuta de reunión	PM.2.3. Realizar reuniones de revisión con el equipo de trabajo para identificar problemas, revisar el estado de los riesgos, registrar los acuerdos y seguirlos hasta su cierre. PM.2.4 Realizar reuniones de revisión con el cliente para registrar los acuerdos y seguirlos hasta su cierre. La <i>Petición de Cambio</i> iniciada por el cliente o por el equipo de trabajo, que afecte al cliente, necesita ser negociada para ser aceptada por ambas partes. Si es necesario, actualizar el <i>Plan del Proyecto</i> de acuerdo a los nuevos acuerdos con el cliente.	4.3 Dirigir y gestionar el trabajo del proyecto

En relación a la tercera actividad, la Tabla 4 muestra los activos definidos para evaluar el rendimiento del plan contra los acuerdos documentados.

**Tabla 4.** Activos de la evaluación y control del proyecto (Mesquida & Mas, 2014).

Activos	Tareas del proceso de Gestión del Proyecto de ISO/IEC 29110-5-1-2	Procesos de PMBoK
Reporte del desempeño del proyecto	PM.3.1 Evaluar el progreso del proyecto con respecto al <i>Plan del Proyecto</i> , comparando las <i>Tareas Actuales</i> contra las <i>Tareas Planeadas</i> , resultados actuales contra los <i>Objetivos</i> establecidos en el proyecto, la asignación actual de los <i>Recursos</i> contra los recursos planeados, el costo actual contra	4.4 Monitorear y controlar el trabajo del proyecto 5.6 Controlar el alcance 6.7 Controlar el calendario 7.4 Controlar los costos

	<p>las estimaciones del presupuesto, el tiempo actual contra el calendario planeado, y los riesgos actuales contra los identificados previamente.</p> <p>PM.3.2 Establecer acciones para corregir las desviaciones o los problemas y riesgos identificados concernientes al cumplimiento del plan, según sea necesario, documentarlos en el <i>Registro de Correcciones</i> y darles seguimiento hasta su cierre.</p> <p>PM3.3 Identificar los cambios a los requisitos y/o <i>Plan del Proyecto</i> para abordar las principales desviaciones, los riesgos potenciales, o problemas relacionados con el cumplimiento del plan, documentarlos en la <i>Petición de Cambio</i>, y seguirlos hasta su cierre.</p>	11.6 Controlar los riesgos
--	---	----------------------------

Por último, la Tabla 5 define los activos para obtener la documentación del proyecto y los productos de acuerdo con los requisitos convenidos.

**Tabla 5.** Activos del cierre del proyecto (Mesquida & Mas, 2014).

Activos	Tareas del proceso de Gestión del Proyecto de ISO/IEC 29110-5-1-2	Procesos de PMBoK
Registro de aceptación	PM.4.1 Formalizar la terminación del proyecto de acuerdo con las <i>Instrucciones de Entrega</i> establecidas en el <i>Plan del Proyecto</i> , proporcionado el apoyo para conseguir la firma del <i>Registro de Aceptación</i> .	5.5 Validar el alcance 4.6 Cerrar el proyecto o la fase
Información histórica	PM.4.2 Actualizar el <i>Repositorio del Proyecto</i> .	4.6 Cerrar el proyecto o la fase

#### 2.4.1.3. Resultados

A través de la implementación del estándar ISO/IEC 29110 en las cuatro empresas participantes en el programa de mejora, esta investigación logró identificar de forma general los siguientes aspectos:

- Estas pequeñas empresas estaban dedicadas plenamente a su trabajo productivo y resuelven los problemas del día a día para sobrevivir. A menudo éstas son incapaces (y de hecho no tienen interés) de dedicar tiempo y esfuerzo para definir sus procesos o activos. Estas empresas no contaban con un departamento de calidad, a diferencia de las grandes organizaciones, que se dedique a estas tareas. Lo anterior tiene sentido, puesto que en las organizaciones pequeñas, los ingenieros de software están más orientados al producto, servicio, o la gestión en lugar de establecer nuevas prácticas de trabajo.
- Estas empresas no tenían conocimiento de la existencia del estándar ISO/IEC 29110-5-1-2. Algunos empleados tenían conocimiento sobre la guía del PMBoK, pero sobre todo, sabían de la existencia de la certificación “Project Management Professional” (PMP). En cualquier caso, estas empresas tenían la creencia de que estos estándares eran demasiado complejos y que estaban fuera de su alcance. En este sentido, los resultados obtenidos en esta investigación coinciden con los argumentos de Boucher, Perrouin, Deprez, y Heymans (2012) quienes afirman que regularmente es todavía necesaria la intervención de los consultores del proceso de software dentro de las empresas. Así, en el contexto de esta tesis, este hallazgo soporta la idea de que las empresas pequeñas necesitan el asesoramiento externo que ofrezca el soporte para el desarrollo y mejora de sus procesos, temas que por lo general son desconocidos y considerados utópicos y lejanos.

- Las pequeñas empresas no sólo necesitan saber qué hacer cuando intentan mejorar sus procesos, también necesitan contar con procedimientos específicos que describan a detalle el trabajo que deben realizar, a través de un conjunto claro de “mejores prácticas” y un conjunto de activos que les ayuden a llevarlas a cabo. Estos procedimientos deben ser sencillos y aplicables a los tipos de proyectos que normalmente realizan.
- Estas compañías invierten muy poco esfuerzo para mejorar la capacitación de los empleados, y cuando lo hacen no siguen un plan de capacitación, sino corresponde a una acción ad hoc derivada de una necesidad detectada a corto plazo.
- Estas empresas generalmente no cuentan con procedimientos explícitos para la compra o adquisición de productos o servicios que se deban obtener fuera de la empresa. Además, no se detectó alguna mejor práctica del proceso de *Gestión del Proyecto* de ISO/IEC 29110-5-1-2 que estuviera relacionada con el área de *Gestión de Compras* de la guía del PMBoK.
- Por último, diferentes tareas provenientes del estándar ISO/IEC 29110-5-1-2 para la identificación de riesgos potenciales fueron establecidas en las empresas, puesto que tradicionalmente éstas no están acostumbradas a realizar las actividades para la gestión de los riesgos. Con regularidad los incidentes son asumidos y las empresas utilizan lo que pueden para reaccionar al momento.

En relación a los aspectos relevantes sobre la aplicación del estándar ISO/IEC 29110-5-1-2, la investigación reporta lo siguiente:

- Durante la definición del proceso de *Gestión del Proyecto* fue necesario establecer un nuevo mecanismo para la comunicación entre todas las partes interesadas en un proyecto. En este sentido, se observó que algunas de estas prácticas fueron realizadas de manera informal, y que no estaban definidas tampoco dentro de las organizaciones. Por lo tanto, aunque el estándar ISO/IEC 29110-5-1-2 no incluye la identificación de las necesidades de información de los participantes y la definición del plan de comunicación, estas buenas prácticas fueron consideradas en el programa de mejora.
- Existían tantos métodos para la gestión de proyectos como jefes de proyectos en las empresas. Dado que no existía un procedimiento establecido, cada empleado utilizaba el método que mejor conocía y que mejor se adaptaba a su forma de trabajar. A fin de obtener un procedimiento estandarizado, fue necesario considerar sus buenas prácticas y llegar a un acuerdo sobre la más adecuada. Con el objetivo de minimizar la resistencia al cambio, y con el fin de generar un proceso útil, esta investigación recomienda el consenso sobre la definición de un método para la gestión de los proyectos, puesto que es mucho mejor alcanzar un acuerdo que imponer una decisión.
- Las empresas necesitan contar con herramientas para ser más eficientes y productivas en su trabajo diario. Así pues, se menciona que fue necesario seleccionar, proponer, y acordar un conjunto de herramientas para apoyar el trabajo colaborativo y las tareas de gestión del proyecto. Debido a que se indica que algunas de las empresas participantes ya habían desarrollado internamente alguna herramienta de software interno para apoyar algunos de estos aspectos, cada empresa seleccionó el subconjunto de herramientas que mejor se adaptó a sus necesidades (e.g., Wiggio, TeamLab, Teambox, y Zoho, entre otras).
- A pesar de que el estándar ISO/IEC 29110-5-1-2 no establece ninguna práctica específica relacionada con el monitoreo del desempeño de los miembros del equipo, se estableció un conjunto de indicadores de desempeño y la forma de monitorearlos.

Continuando con los hallazgos, la investigación concluye que la valoración final sobre esta iniciativa para implementar el estándar ISO/IEC 29110-5-1-2 en las cuatro empresas pequeñas fue totalmente positiva. Como principales fortalezas, claves para el éxito del proyecto, la investigación destaca los siguientes aspectos:

- La participación activa, la motivación y la conciencia de todas las empresas participantes.
- La voluntad de compartir el conocimiento entre las empresas, puesto que es importante mencionar que las cuatro empresas se encuentran dentro del mismo sector, y en algunas ocasiones compiten para obtener un nuevo proyecto/cliente.
- El establecimiento de un plan detallado y su conformidad con sólo algunas pequeñas desviaciones. El programa definió una agenda muy clara que incluía la vigilancia periódica en todas las empresas. Sin estas revisiones conjuntas y la comprensión de las acciones de mejora que el equipo debía realizar, el costo de la implementación del estándar hubiera sido mucho mayor.

Por último, desde un punto de vista técnico, la investigación resalta lo siguiente:

- La importancia de seleccionar a un representante en cada empresa. Esta persona debe canalizar las necesidades y requerimientos de su empresa y exponerlos en las reuniones celebradas entre todos los representantes y consultores. Los acuerdos alcanzados en estas reuniones requieren posteriormente la aprobación de cada empresa.
- La selección de una herramienta de trabajo colaborativo (e.g., TeamLab) para apoyar la comunicación y el intercambio de archivos entre todos los representantes y consultores.

## **2.4.2. Gestión cuantitativa de proyectos en pequeñas y medianas empresas de software**

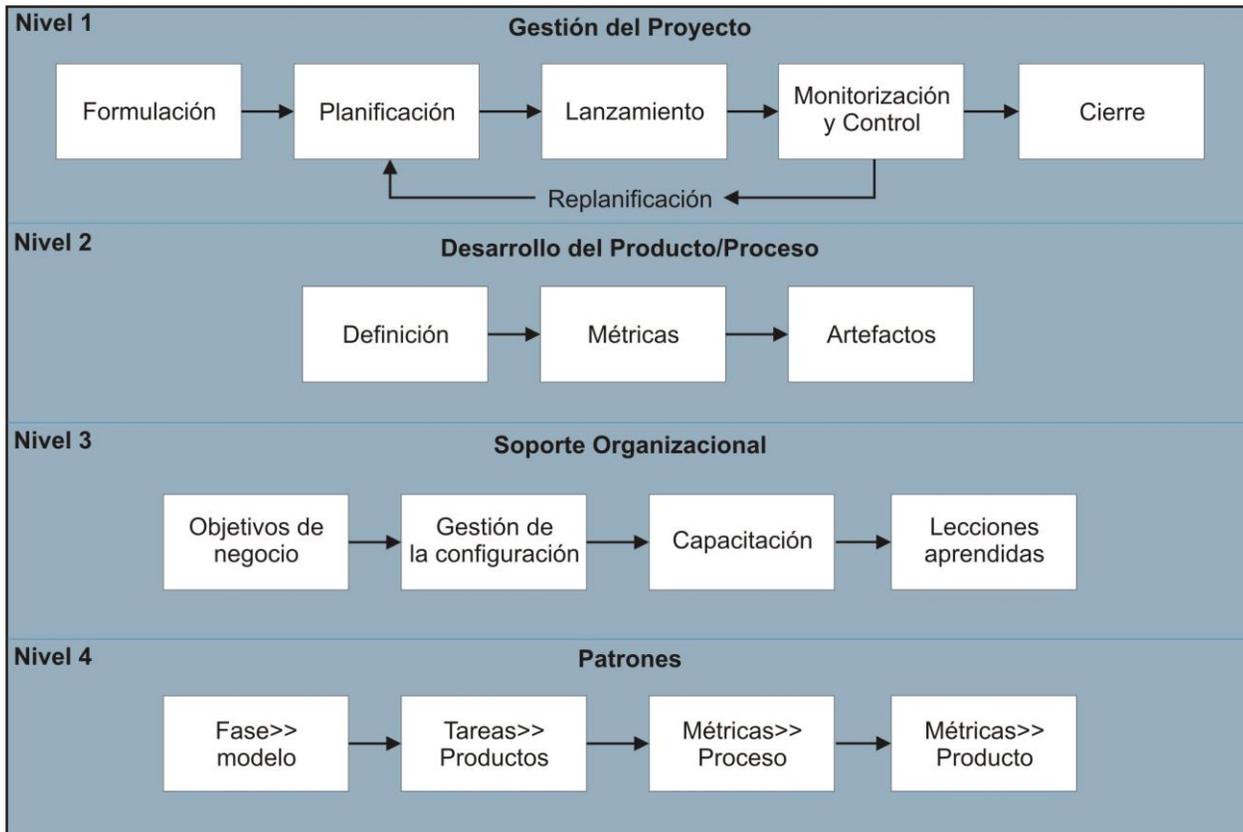
### **2.4.2.1. Objetivo**

La investigación realizada en (García, Pacheco, & Calvo-Manzano, 2014) plantea la introducción de la gestión cuantitativa de los proyectos de software, usualmente implantada únicamente en las empresas grandes con altos niveles de madurez, para cubrir las necesidades particulares de las PyMES y brindar un soporte para realizar las tareas relacionadas con la gestión de sus proyectos. Así, esta investigación diseñó un Marco de Trabajo para la Gestión Cuantitativa de Proyectos en Pequeñas Empresas, denominado FQPMSE por sus siglas en inglés. Adicionalmente, con el objetivo de mostrar la aplicabilidad de este marco de trabajo la investigación realizó un caso de estudio que demuestra la mejora sustancial de tres PyMES.

### **2.4.2.2. Descripción**

Este trabajo introdujo a FQPMSE con el objetivo de agregar mayor granularidad a las actividades de planificación del proyecto y monitorización y control del proyecto; como ya se ha mencionado anteriormente, ambos son procesos básicos de la gestión de los proyectos. Para lograr este objetivo fue necesario establecer dos componentes principales de trabajo: un marco conceptual y un entorno de Ingeniería de Software. En este sentido, el *marco conceptual* sería el encargado de proporcionar el soporte necesario para representar y gestionar el conocimiento relacionado con los dos procesos relacionados con la gestión de los proyectos, considerando la perspectiva cuantitativa de su integración. Para esto, se diseñó una arquitectura conceptual de metadatos compuesta por cuatro capas de abstracción (véase Figura 2.5) cuyo objetivo fue representar el aspecto cuantitativo mediante la definición de las relaciones entre los elementos de las diferentes capas de abstracción y

la integración de activos de proceso. Dicha integración requirió el diseño de un patrón<sup>15</sup> para la definición de los procesos de la organización, la representación de los modelos de proceso con mayor renombre y aplicabilidad dentro del contexto de las PyMES, y la definición de la información necesaria para representar cualquier artefacto relacionado con el proceso de software que pudiera ser candidato para la gestión. Es importante mencionar que esta arquitectura también incluye las mejores prácticas para la definición y medición de los procesos de software, correspondientes al Nivel 2 de CMMI-DEV v1.3.



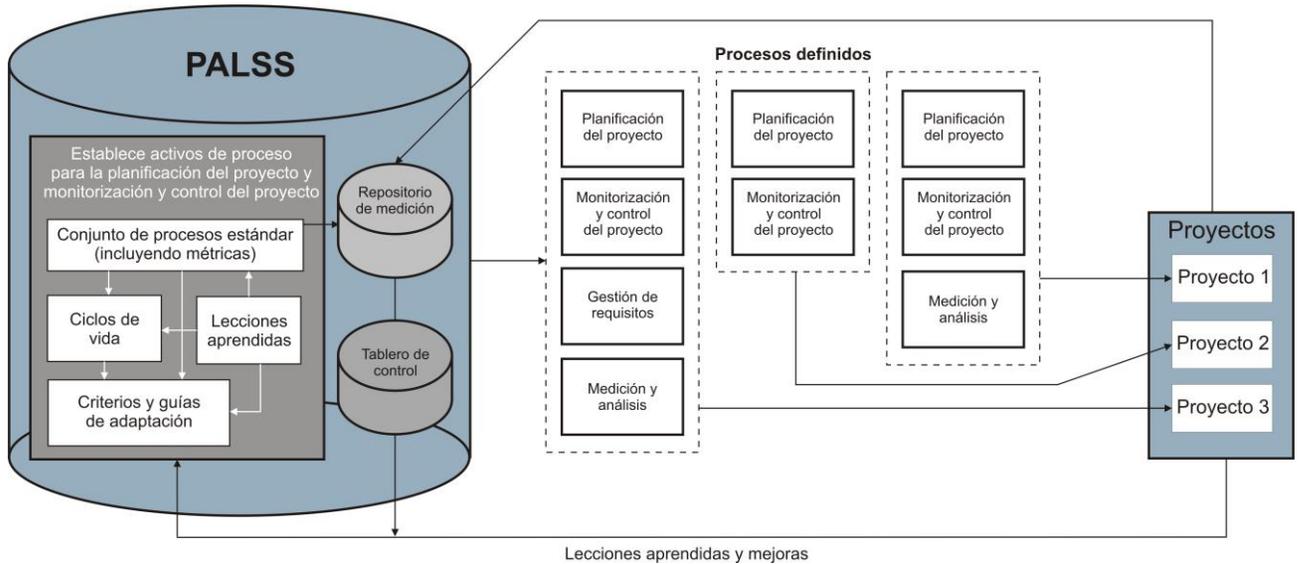
**Figura 2.5.** Capas de abstracción del marco conceptual de FQPMSE (García, Pacheco, & Calvo-Manzano, 2014).

La implantación de este marco de trabajo requirió la introducción de una PAL para apoyar a las PyMES en la definición de sus procesos. Como ya se mencionó anteriormente, la PAL es un repositorio de activos comunes que facilita la estandarización de los procesos, puesto que permite que todos los proyectos utilicen el mismo proceso estándar o las diferentes adaptaciones aprobadas para este proceso. En este sentido, cada proyecto debe seleccionar de la PAL su propio proceso para realizar la gestión considerando los criterios y las directrices de adaptación. Adicionalmente, la PAL almacena en el repositorio de medición los valores estimados y reales de los esfuerzos en relación con las diferentes tareas de los proyectos, permitiendo de esta manera que los diferentes proyectos puedan estimar rendimientos futuros de acuerdo a datos históricos (véase Figura 2.6).

Por otro lado, el *entorno de Ingeniería de Software* de FQPMSE se integró con dos herramientas computacionales que brindan el soporte tecnológico al marco conceptual: PALSS

<sup>15</sup> En el contexto de la Ingeniería de Software, el concepto de patrón se relaciona con un esqueleto de soluciones a problemas comunes en el desarrollo de software. En otras palabras, el patrón proporciona una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares (van den Bergh, Van Haeren, Van Baelen, Berbers, & Joosen, 2013).

(PAL para las pequeñas empresas de software) la cual, como ya se indicó, se enfoca en definir y gestionar la arquitectura conceptual, y PROMEP la cual está diseñada para apoyar a las PyMES con la gestión de sus proyectos utilizando el concepto de patrón definido por PALSS. Cabe señalar que PALSS define las actividades a desarrollar en base a otra herramienta que proporciona apoyo a las PyMES para la evaluación de sus procesos y que fue denominada SysProVal (García, Pacheco, & Calvo-Manzano, 2010).



**Figura 2.6.** Representación de la PAL requerida para la implantación de FQPMSE (García, Pacheco, & Calvo-Manzano, 2014).

Así, todos los metadatos gestionados por PROMEP y PALSS son almacenados en un repositorio que permite el intercambio de información durante todo el proceso y entre los proyectos. En este sentido, el entorno de ingeniería de FQPMSE cuenta con tres elementos principales:

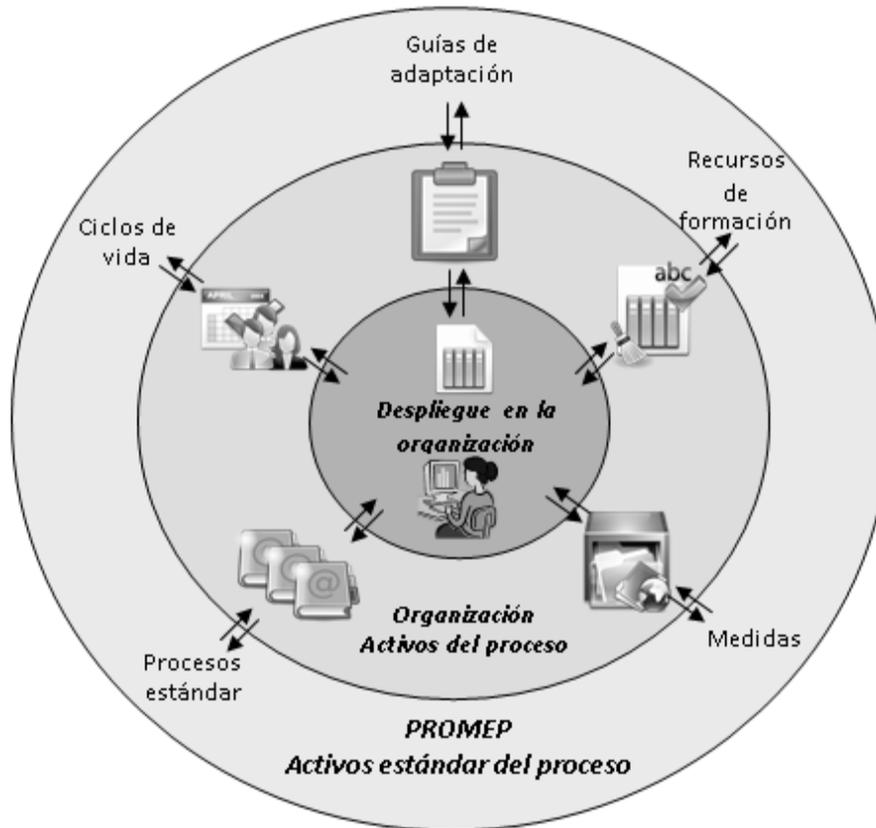
- Repositorio de metadatos. La representación de los metadatos, como se mencionó anteriormente, es gestionada a través de diferentes niveles de abstracción puesto que, de acuerdo a los autores, es el medio más acorde para manejar la complejidad y diversidad de la información necesaria para soportar la definición y evaluación de los procesos de software en el contexto de las PyMES.
- Repositorio de medición del software. Este repositorio proporciona el apoyo cuantitativo requerido para la gestión de los proyectos.
- Un tablero de control. Este mecanismo permite gestionar el conocimiento de los proyectos de software a través de la definición de procesos estándar, actividades para los jefes de proyectos, catálogos de ciclos de vida, guías de adaptación, material de capacitación, y medidas (véase Figura 2.7). De esta manera, el tablero de control promueve el uso de los activos de proceso relacionados con la planificación del proyecto y la monitorización y control del proyecto y brinda soporte para la gestión de los riesgos y la definición de un proceso que puede ser controlado cuantitativamente.

Es así como en esta investigación se planteó la introducción de FQPMSE con el objetivo de promover entre las PyMES la gestión y medición de sus procesos de software. Dicha gestión es

apoyada mediante la inclusión de un repositorio de datos en el que se establece una base de conocimiento sobre la gestión de los procesos orientada a su evaluación y mejora.

### 2.4.2.3. Resultados

Los resultados de esta investigación plantearon el diseño de un caso de estudio que se enfocó en evaluar la implementación de FQPMSE en tres PyMES (véase Tabla 6).



**Figura 2.7.** Entorno de Ingeniería de Software de FQPMSE (García, Pacheco, & Calvo-Manzano, 2014).

**Tabla 6.** Perfil de las PyMES participantes (García, Pacheco, & Calvo, 2014).

PyMES	Tamaño de la plantilla	Actividad principal
1	20	Desarrollo de software a medida
2	15	Desarrollo de software a medida y empaquetado
3	17	Desarrollo de software a medida y diseño de páginas web

El caso de estudio inició con la aplicación de cuestionarios de evaluación para obtener una línea base inicial. El equipo de evaluación fue integrado por tres investigadores y un jefe de proyectos de cada PyMES. Además, se seleccionaron dos jefes de proyectos más por cada empresa (para un total de seis jefes de proyectos participantes) a fin de evaluar los procesos relacionados con la gestión de los proyectos. La aplicación de estos cuestionarios permitió que los investigadores identificaran, por ejemplo, a las personas con conocimientos sobre la estimación de software y la creación de planes y éstas fueron invitadas a participar en la definición del nuevo proceso. Así, la línea base inicial permitió que los investigadores determinaran cuáles actividades habían sido cubiertas por el equipo de desarrollo y cuáles habían sido desplegadas en toda la organización como un proceso institucionalizado. De acuerdo a los autores, los resultados obtenidos mostraron un 42.0% de cobertura media para la planificación del proyecto y un 30.5% para la monitorización y

control del proyecto. Al final de la evaluación, la información obtenida fue cotejada con la documentación proporcionada por cada empresa y se condujeron entrevistas como parte de la revisión de evidencia objetiva. Tres meses después de completar los proyectos piloto bajo el enfoque establecido por FQPMSE se obtuvieron los resultados mostrados en la Tabla 7. Estos datos muestran que todas las empresas aumentaron el nivel de cobertura de los procesos en un 11.0%, en comparación con los valores obtenidos en la primera evaluación. Por ejemplo, la PyMES 1 presentó una cobertura inicial de 54.4% para el proceso de planificación del proyecto antes de la implementación de FQPMSE. Una nueva evaluación, después de usar el marco de trabajo y adoptar la PALSS en las actividades diarias, mostró un incremento del 9.0% sobre la cobertura. De acuerdo a los autores, el sesgo fue controlado mediante el uso de revisiones sobre la documentación y las entrevistas con los grupos de gestión y técnicos superiores. Además, en relación al uso de la PALSS como herramienta de soporte a la implementación de FQPMSE, la Tabla 8 muestra datos que fueron obtenidos considerando el esfuerzo total (en horas) del personal centrado en la mejora y el esfuerzo individual para cada PyMES. En este sentido, los autores destacan que fue interesante observar la relación entre el esfuerzo individual y la mejora obtenida (cuando se utilizó PALSS). Por ejemplo, la PyME 3 invirtió más horas en utilizar PALSS y logró el mayor incremento en la mejora de sus procesos.

**Tabla 7.** Comparación de los resultados entre las evaluaciones (García, Pacheco, & Calvo-Manzano, 2014).

PYME	Cobertura			
	Proceso	Antes de FQPMSE	Después de FQPMSE	% incremento
1	Planificación	54.0%	58.9%	9.0%
	Monitorización y control	38.0%	42.2%	11.0%
2	Planificación	30.0%	33.3%	11.0%
	Monitorización y control	20.0%	22.2%	11.0%
3	Planificación	41.0%	46.0%	12.2%
	Monitorización y control	33.0%	37.0%	12.1%

**Tabla 8.** Información de mejora por empresa (García, Pacheco, & Calvo-Manzano, 2014).

PYME	Tamaño de personal	Total esfuerzo (horas)	Esfuerzo individual	Promedio de mejora
1	20	221	8.76	0.100
2	15	400	25.18	0.111
3	17	600	35.40	0.120
Total	17	407	24.40	0.110

De esta forma, y de acuerdo a sus creadores, FQPMSE define a detalle los elementos necesarios para orientar a las PyMES en la realización de las actividades de formulación y ejecución de los planes de proyecto, a través de activos de proceso propios de la gestión de los proyectos. Así, a través del concepto de PAL, FQPMSE facilita la institucionalización de las prácticas efectivas para la gestión de proyectos en el contexto de las PyMES.

### 2.4.3. Definición de prácticas para la planificación de proyectos en pequeñas y medianas empresas en base a sistemas de planificación de recursos

#### 2.4.3.1. Objetivo

Los objetivos de esta investigación se enfocan en examinar los planes para la implementación de proyectos de Planificación de Recursos Empresariales (ERP, por sus siglas en inglés) y estudiar el efecto de las prácticas recomendadas para la planificación del proyecto sobre el

éxito de este tipo de proyectos (Tasevska, Damij, & Damij, 2014). Para cubrir estos objetivos fue necesario considerar cuatro medidas de la planificación de un proyecto: desarrollo de casos de negocio, la planificación del alcance, el desarrollo del plan de línea base, y la planificación de riesgos, junto con tres medidas de éxito de un proyecto: la satisfacción del cliente, la calidad percibida del proyecto, y el éxito del proceso de implementación. Los resultados de este estudio se obtuvieron mediante la aplicación de una encuesta a 30 PyMES de la República de Macedonia.

### 2.4.3.2. Descripción

Actualmente muchas PyMES están implementando proyectos de ERP con el fin de mejorar sus procesos de negocio empleando una cantidad mínima (o por lo menos costeable) de recursos, y algunas otras simplemente lo están haciendo por cumplir con requisitos legales. Por lo general, las PyMES solamente implementan pocos módulos de este tipo de sistemas, concretamente aquellos enfocados a cubrir funciones financieras, de inventario y contabilidad. En este contexto, esta investigación plantea el análisis de las actividades realizadas en la implementación de proyectos de ERP, particularmente de las prácticas relacionadas con la planificación de los proyectos, para determinar si éstas impactan en el éxito del proyecto.

En este sentido, fue necesario identificar las prácticas relacionadas con la planificación de los proyectos y considerarlas como variables independientes para someterlas a prueba. Así, 21 variables fueron utilizadas para medir el esfuerzo de las PyMES en la planificación de los proyectos de ERP, tal como se muestra en la Tabla 9. Estas variables fueron organizadas en cuatro dimensiones: prácticas para el desarrollo de los casos de negocio<sup>16</sup>, prácticas para la planificación del alcance, prácticas para el desarrollo del plan de línea base, y prácticas para la planificación de los riesgos.

**Tabla 9.** Dimensiones de medición de la planificación de proyectos (Salomo, Weise, & Gemunden, 2007).

Casos de negocio (CN)	Planificación del alcance (PA)	Plan de línea base (PLB)	Planificación de riesgos (PR)
En general, el análisis que se realizó antes de implementar el ERP fue exhaustivo y metódico (CN1)	Se definieron las metas que se pretendía alcanzar con la implementación del ERP (PA1)	Se definieron todas las actividades necesarias para ejecutar la implementación del ERP (PLB1)	Se llevó a cabo el análisis de los riesgos y sus consecuencias (PR1)
Se identificaron los principales controladores en la implementación del ERP (CN2)	Se definieron todos los resultados que deberían entregarse durante la implementación (PA2)	No se definió la secuencia de las actividades (PLB2)	Se crearon planes detallados para la reducción de la incertidumbre (PR2)
Se identificaron, de forma sistemática, soluciones alternativas de ERP (CN3)	No se definieron los eventos importantes que debían ocurrir durante la implementación (PA3)	Se definió la duración de las actividades (PLB3)	Se crearon planes detallados de respuesta a los riesgos (PR3)
Se seleccionó, de forma sistemática, la solución preferida de ERP (CN4)	Se definieron los requisitos que el software debía cumplir (PA4)	No se definieron los recursos necesarios para la ejecución de las actividades (PLB4)	
Se evaluó el ajuste entre la implementación del ERP y la estrategia corporativa (CN5)	No se consideraron todas las restricciones que se pudieran presentar durante la	Se estableció un calendario detallado para la implementación del ERP (PLB5)	

<sup>16</sup> El caso de negocio es un documento que se construye a partir de un diagnóstico previo, tanto de una situación interna a resolver como de un objetivo común de negocio. Este documento describe prácticamente la justificación de un proyecto en términos de si los beneficios previstos compensan los costos y riesgos estimados. Un caso de negocio es a menudo complejo y puede requerir análisis financiero, análisis técnico, análisis del impacto de la organización y un estudio de viabilidad (Söderlund & Müller, 2014).

	implementación (PA5)		
Los departamentos relevantes participaron en el proceso de planificación (CN6)		Se estableció un presupuesto detallado para la implementación del ERP (PLB6)	
El equipo/persona responsable se comprometió con las metas del proyecto (CN7)			

En relación a la primera dimensión, el desarrollo de un caso de negocio, se midieron las variables definidas involucradas en la planificación inicial que resulta de la creación del caso de negocio. En este sentido, los autores utilizaron siete de los nueve puntos definidos en la escala de medición establecida por Salomo et al. (2007), puesto que dos puntos (Escenarios alternativos de mercado y Ajuste con las competencias básicas) no aplicaban para el caso de estudio. Dado que, de acuerdo a los autores, no se identificaron escalas comprensibles para la segunda y tercera dimensión (es decir, las prácticas relacionadas con la planificación del alcance y el desarrollo del plan de línea base, respectivamente), se desarrolló una escala propia de acuerdo con las recomendaciones publicadas en (Saunders, Lewis, & Thornhill, 2003) y (Hair, Black, Babin, & Anderson, 2010). La cuarta dimensión, planificación de riesgos, fue medida con tres elementos definidos en el estudio de Salomo et al. (2007).

La investigación requirió que los encuestados respondieran, a través de un determinado nivel de acuerdo o desacuerdo, a afirmaciones relacionadas con el uso de una determinada práctica de planificación (con valores que iban desde 1 - Muy en desacuerdo hasta 7 - Muy de acuerdo). Para determinar si las empresas implementaban ciertas prácticas de planificación o no, las variables fueron codificadas y los valores fueron clasificados en dos grupos: *no implementado* (respuestas que iban desde 1 - Completamente en desacuerdo hasta 4 - Neutral) e *implementado* (respuestas que iban desde 5 - Parcialmente de acuerdo a 7 - Totalmente de acuerdo). Así, los cuestionarios y las entrevistas fueron las técnicas utilizadas para la recolección de los datos primarios. Posteriormente, el éxito de un proyecto como una variable dependiente fue medido mediante el uso de una escala que constó de 12 ítems que evaluaban tres dimensiones del éxito: la satisfacción del cliente, la calidad percibida, y el éxito del proceso de implementación (véase Tabla 10). De acuerdo a los autores, esta escala fue utilizada puesto que ya había sido utilizada previamente por varios autores para analizar proyectos de Sistemas de Información (SI) específicamente, y su fiabilidad ha sido reportada en la revista *Project Management Journal* del PMI.

**Tabla 10.** Dimensiones de medición de los proyectos exitosos (Mahaney & Lederer, 2006).

Satisfacción del cliente (SC)	Calidad percibida (CP)	Proceso de implementación (PI)
El software ERP que fue implementado funcionó (SC1)	El software ERP implementado fue la mejor opción de un conjunto de alternativas (CP1)	La implementación del ERP se realizó dentro del calendario original (PI1)
El software ERP es utilizado por los usuarios previstos (SC2)	El uso del software ERP condujo directamente a mejorar la toma de decisiones y el rendimiento de los usuarios (CP2)	La implementación del ERP se realizó dentro del presupuesto original (PI2)
El software ERP benefició directamente a los usuarios previstos tanto en el aumento de la eficiencia como en la eficacia de los empleados (SC3)	Este software tuvo un impacto positivo en aquellos que lo usaron (CP3)	Estoy satisfecho con el proceso por el cual se completó el software ERP (PI3)
Los usuarios importantes, directamente afectados por el software ERP, son los que lo	Los resultados de la implementación de este software ERP representan una mejora definitiva en la forma en que	

utilizan (SC4)	los usuarios realizan estas actividades (CP4)	
Se tiene plena confianza que los problemas (no técnicos) de puesta fueron mínimos, ya que el software ERP fue fácilmente aceptado por sus usuarios previstos (SC5)		

En este sentido, la primera dimensión intentó medir el nivel de aceptación del proyecto a través de los beneficios previstos por los usuarios. La segunda dimensión midió el efecto del proyecto en términos de la mejora del desempeño. La tercera dimensión intentó averiguar si el proyecto fue terminado a tiempo y dentro del calendario, y si las metas técnicas fueron cumplidas. De manera similar, se pidió a los encuestados que respondieran en qué medida estaban de acuerdo o en desacuerdo con cada afirmación hecha a través de una escala Likert de siete puntos (con valores que iban desde 1 - Muy en desacuerdo hasta 7 - Muy de acuerdo). Con el fin de descubrir si los proyectos tuvieron éxito o no, las variables también fueron codificadas y los valores de las respuestas fueron clasificados en dos grupos: no exitoso (respuestas que iban desde 1 - Completamente en desacuerdo hasta 4 - Neutral) y exitoso (respuestas que iban desde 5 - Parcialmente de acuerdo a 7 - Totalmente de acuerdo). En consecuencia, las hipótesis de este caso de estudio se establecieron como:

- Hipótesis 1: El nivel de esfuerzo destinado al desarrollo del caso de negocio está relacionado positivamente con el éxito de un proyecto en términos de la satisfacción del cliente, la calidad percibida, y el proceso de implementación.
- Hipótesis 2: El nivel de esfuerzo destinado en la planificación del alcance está relacionado positivamente con el éxito de un proyecto en términos de la satisfacción del cliente, la calidad percibida, y el proceso de implementación.
- Hipótesis 3: El nivel de esfuerzo destinado al desarrollo del plan de línea base está relacionado positivamente con el éxito de un proyecto en términos de la satisfacción del cliente, la calidad percibida, y el proceso de implementación.
- Hipótesis 4: El nivel de esfuerzo destinado al desarrollo del plan de riesgos está relacionado positivamente con el éxito de un proyecto en términos de la satisfacción del cliente, la calidad percibida, y el proceso de implementación.

Antes de la aplicación del cuestionario se realizaron entrevistas exhaustivas a tres propietarios de las PyMES. El propósito de estas entrevistas consistió en obtener una visión general de la forma en que las empresas se enfocaron en la implementación del ERP y las metas que perseguían con esto. A través de esta técnica (entrevistas) se obtuvo información sobre cómo se realizaba la planificación de un proyecto y cómo se evalúa el éxito de la implementación de un ERP. Durante las entrevistas también se discutieron las preguntas iniciales del cuestionario inicial y se examinó la terminología que los jefes o líderes de proyectos utilizan cuando se habla de la planificación en general. La entrevista inició con tres preguntas abiertas, pero durante el debate se añadieron otras más con el objetivo de alcanzar los objetivos de esta investigación. Estas preguntas fueron definidas de la siguiente manera:

- ¿Cómo se decidió a implementar un sistema ERP en general?
- ¿Qué prácticas de planificación realizaba antes de la actual implementación?
- ¿Está satisfecho con el sistema ERP después de que fue implementado?

Otro cuestionario fue utilizado para recoger datos estandarizados de los representantes de las PyMES a través de preguntas que recogen su opinión sobre las medidas relacionadas con la planificación de un proyecto y su relación con el éxito del mismo. Dicho cuestionario fue estructurado en seis secciones: la primera sección se enfocó a preguntas de tipo general; las siguientes cuatro secciones recogieron información sobre el nivel de implementación de cada una de las dimensiones de la planificación de proyectos de TI; y la sexta sección contenía los ítems que median el éxito del proyecto. El cuestionario fue distribuido electrónicamente y aplicado, en algunos casos, de manera personal durante julio de 2012.

De acuerdo con los autores, se realizó un análisis de factores con el fin de confirmar las dimensiones que definen tanto a la planificación de un proyecto como el éxito del mismo, y para diseñar las escalas a utilizar en análisis posteriores. El efecto independiente de las dimensiones de la planificación de un proyecto sobre las dimensiones de su éxito fue medido simplemente a través de un análisis de correlación mediante el coeficiente de Pearson. El efecto de todas las dimensiones de la planificación de un proyecto sobre cada una de las dimensiones de su éxito fue analizado mediante un análisis de regresión. Dicho análisis fue utilizado con el objetivo de comprender mejor la relación entre los factores de la planificación de un proyecto y los factores de éxito de un proyecto como sugiere Dvir (2005).

Por último, la investigación utilizó una combinación de los enfoques de comodidad y muestreo de bola de nieve (*snowball sampling*) para seleccionar 30 PyMES de diferentes industrias. Este método particular de muestreo no probabilístico fue elegido debido a la falta de una lista de PyMES, incluidas las microempresas, que hubieran introducido soluciones de ERP recientemente.

#### **2.4.3.3. Resultados**

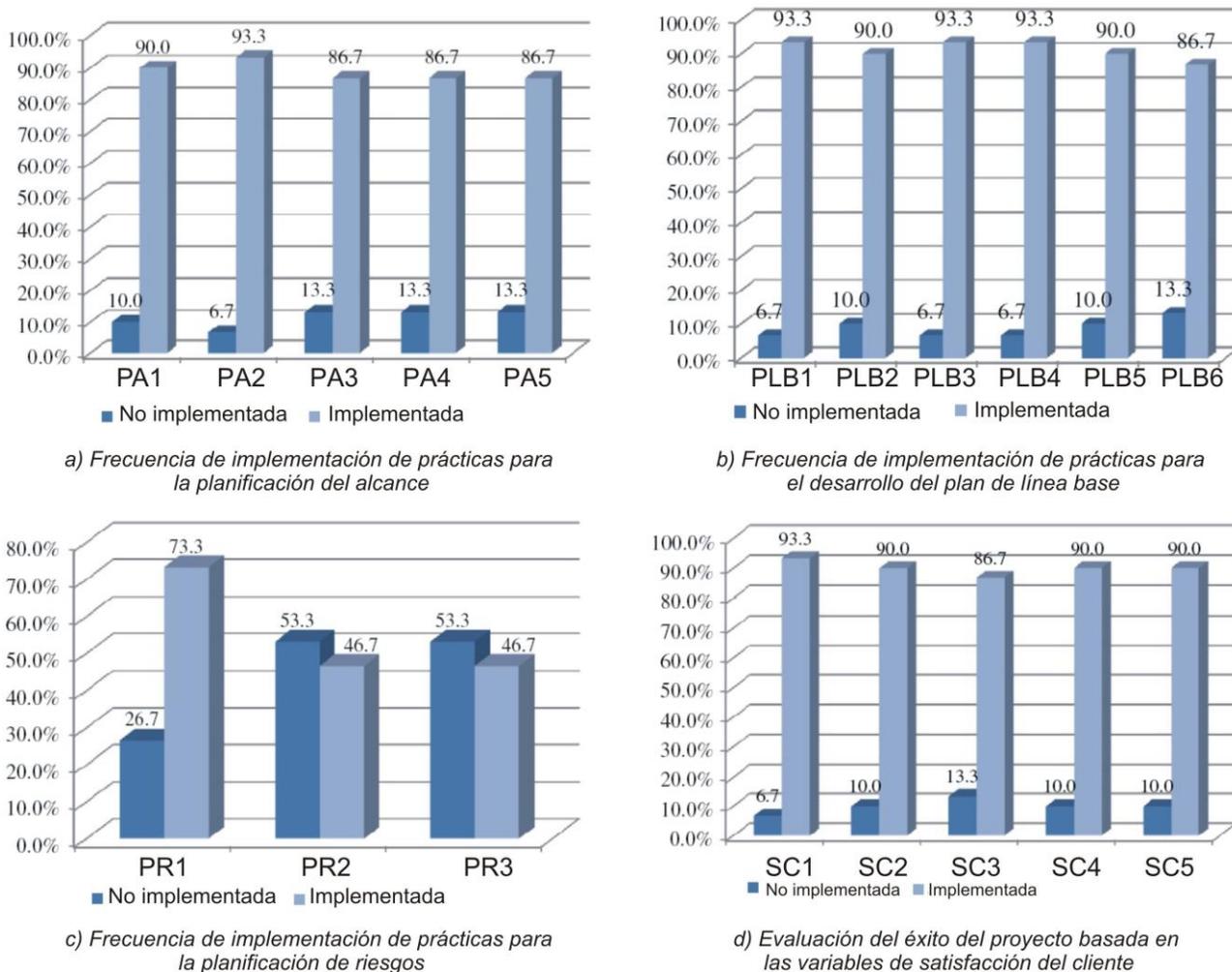
Los resultados reportados por esta investigación mostraron que los representantes de las PyMES no utilizaban el término ERP para referirse a la solución que habían implementado, pero era común que se refirieran a los términos “software” o “programa de computadora”. Sin embargo, cuando los investigadores pidieron describir las funcionalidades de su software, estaba claro que se trataba de un ERP. Dado que únicamente uno de los representantes utilizó el término ERP, se agregó al cuestionario una definición para este término indicando que se refería a aquellos programas informáticos utilizados para la gestión de los recursos internos o software destinado a la gestión de las operaciones internas de las empresas.

En base a las respuestas del cuestionario, concretamente para la segunda pregunta, el estudio concluyó que las PyMES en efecto utilizan algunas prácticas de planificación, pero no las reconocen como tal (i.e., no consideran que la planificación sea una fase separada en el desarrollo de sus proyectos). Por otra parte, las empresas no utilizaban alguna herramienta de planificación en particular, como el diagrama de Gantt o la WBS, incluso los representantes no estaban familiarizados con estos términos. Por lo tanto, el cuestionario fue modificado para formular preguntas sobre las prácticas generales de la planificación de proyectos y evitar entrar en detalles acerca de las herramientas específicas utilizadas como soporte.

En relación a la tercera pregunta, dos de los representantes de las PyMES consideraron que habían realizado una implementación exitosa de su ERP. Incluso uno de ellos informó estar satisfecho con los beneficios que el ERP proporcionó a su gestión, pero no estaba satisfecho con el proceso de implementación en sí, ya que tomó demasiado tiempo para que el software se ejecutara y fuera aceptado por todos los usuarios. Más tarde se determinó que esto fue una consecuencia de no controlar las actividades dentro del calendario del proyecto.

Con el fin de proporcionar conclusiones sobre la frecuencia de aplicación de las prácticas analizadas durante la planificación de un proyecto, los autores examinaron las respuestas a través de un análisis de frecuencia y representaron a través de diagramas de barras sus principales hallazgos

(véase Figura 2.8). Estas gráficas muestran que las PyMES implementaron algunas prácticas relacionadas con la planificación de los proyectos, tal y como es sugerido en la literatura actual. Por ejemplo, al menos el 86.7% de los encuestados afirmó que sus empresas implementan las prácticas de planificación que intervienen en el desarrollo de un caso de negocio, el alcance del proyecto y el diseño de un plan de línea base (véase Figura 2.8 (a)). Sin embargo, el estudio también mostró que las prácticas relacionadas con la planificación de los riesgos fueron las menos implementadas, en comparación con otras prácticas de la planificación de proyectos (véase Figura 2.8 (c)). En este sentido, el establecimiento de planes detallados para la reducción de la incertidumbre (PR2) y planes detallados para la mitigación de los riesgos (PR3) fueron implementados por solamente el 46.7% de los encuestados. De acuerdo a los investigadores este resultado tiene sentido, puesto que es común que los jefes de proyectos de una PyMES consideren las actividades de gestión de los riesgos como un trabajo y gasto extras, y por lo tanto evitan su ejecución. El análisis de riesgos, por otro lado, y sus consecuencias (PR1) lo realizaron el 73.3% de los encuestados.



**Figura 2.8.** Principales resultados sobre el estudio de planificación en base a ERPs (Mahaney & Lederer, 2006).

Con respecto a las evaluaciones sobre el éxito de un proyecto, la mayoría de los encuestados (al menos el 85%) valoraron los proyectos como exitosos en base a las dimensiones de satisfacción del cliente y calidad percibida (véase Figura 2.8 (d)). Mientras que para la última dimensión,

relacionada con el proceso de implementación, se obtuvieron respuestas negativas puesto que ésta (la implementación en sí) no se realizó dentro del tiempo y presupuesto establecidos. La mayoría de las respuestas negativas se dieron para la variable PI2 (33.3%), que mide si el proyecto se completó dentro del presupuesto. Considerando estos resultados, los investigadores argumentan que los principales problemas que enfrentaron estas PyMES durante la implementación de su proyecto se relacionaron con los excesos en el presupuesto y, en menor frecuencia, con la replanificación de las actividades. Es decir, las PyMES experimentaron excesos de presupuesto y el desbordamiento del calendario.

Por último, el estudio argumenta que la principal recomendación sería que, a pesar de que todas las prácticas para la planificación de un proyecto son importantes para alcanzar el éxito del mismo, las PyMES deben poner más esfuerzo en el desarrollo de un modelo de negocio y un plan de línea base. Estas dos medidas parecen tener un efecto importante en el éxito de los proyectos de las PyMES incluidas en la muestra del estudio, pues to que abarcan actividades tangibles y mensurables que son más fáciles de entender si se comparan con la planificación del alcance y las actividades de planificación de los riesgos. Esto es especialmente válido para las empresas que no poseen los recursos humanos con habilidades adecuadas sobre la gestión de los proyectos, como las PyMES de este caso de estudio.

#### **2.4.4. Gestión de proyectos de software en pequeñas y micro empresas: un enfoque de integración de ISO/IEC 12207, ISO/IEC 29110 y PMBoK**

##### **2.4.4.1. Objetivo**

La investigación realizada por Mas y Mesquida (2013) identificó un conjunto de mejores prácticas para facilitar la incorporación de los procesos de la gestión de los proyectos en micros y pequeñas empresas desarrolladoras de software mediante el análisis de las relaciones existentes entre diferentes estándares que proporcionan una guía para realizar la gestión, y la unión del proceso de Gestión de proyectos del estándar ISO/IEC 29110-5-1-2 y el conocimiento de áreas específicas de la guía del PMBoK.

##### **2.4.4.2. Descripción**

Esta investigación parte de la idea de que es bien sabido que la gestión de proyectos resulta crucial para la mayoría de las empresas de software, sea cual sea su tamaño o el tipo de aplicaciones que desarrollen. Así pues, para una empresa que desee iniciarse en la gestión de proyectos es conveniente adoptar un marco conocido y probado, que cubra todo el ciclo de vida de un proyecto y que aborde los diferentes aspectos a gestionar. Pero, ¿Cuál de todos ellos escoger? ¿Qué estándar seleccionar? ¿Cuál es el modelo que mejor se adapta a las necesidades de una pequeña o micro empresa? El estudio realizado, y que ahora se resume, pretende dar respuesta a estas preguntas. En este sentido, los autores analizaron los siguientes estándares de ISO:

- La norma ISO/IEC 12207 que agrupa todos los procesos del ciclo de vida del software, algunos de éstos están relacionados con la gestión de los proyectos.
- La norma ISO/IEC 29110-5-1-2, que es un estándar específico para empresas de desarrollo de software de hasta 25 empleados que incluye un subconjunto de procesos de la norma ISO/IEC 12207 adaptados a las necesidades de este tipo de empresas.

Sin embargo, la gran variedad de estándares existentes puede dificultar la elección de la norma que mejor se adapte a la situación y objetivos de una pequeña o micro empresa que desee implantar la gestión de los proyectos. El mapa de procesos de la norma ISO/IEC 12207:2008, por

ejemplo, establece un marco común para los procesos del ciclo de vida del software. El problema para la pequeña empresa, es que éste contiene un conjunto de buenas prácticas relacionadas con la adquisición, suministro, desarrollo, operación, mantenimiento y retirada de productos y servicios de software a través de 43 procesos, los cuales son difíciles de entender si el personal de la empresa no está relacionado con las cuestiones técnicas a las que ahí se hace referencia. Esta norma suministra además un modelo de procesos (o de referencia) que puede ser utilizado para realizar una evaluación de la capacidad de los procesos de acuerdo al estándar de evaluación y mejora de procesos ISO/IEC 15504-2 (SPICE). La Figura 2.9 muestra que estos procesos se encuentran agrupados en dos grandes categorías y, a su vez, en siete grupos. En esta figura se han sombreado aquellos procesos de la norma ISO/IEC 12207 que guardan algún tipo de relación con las buenas prácticas propias de la gestión de proyectos.

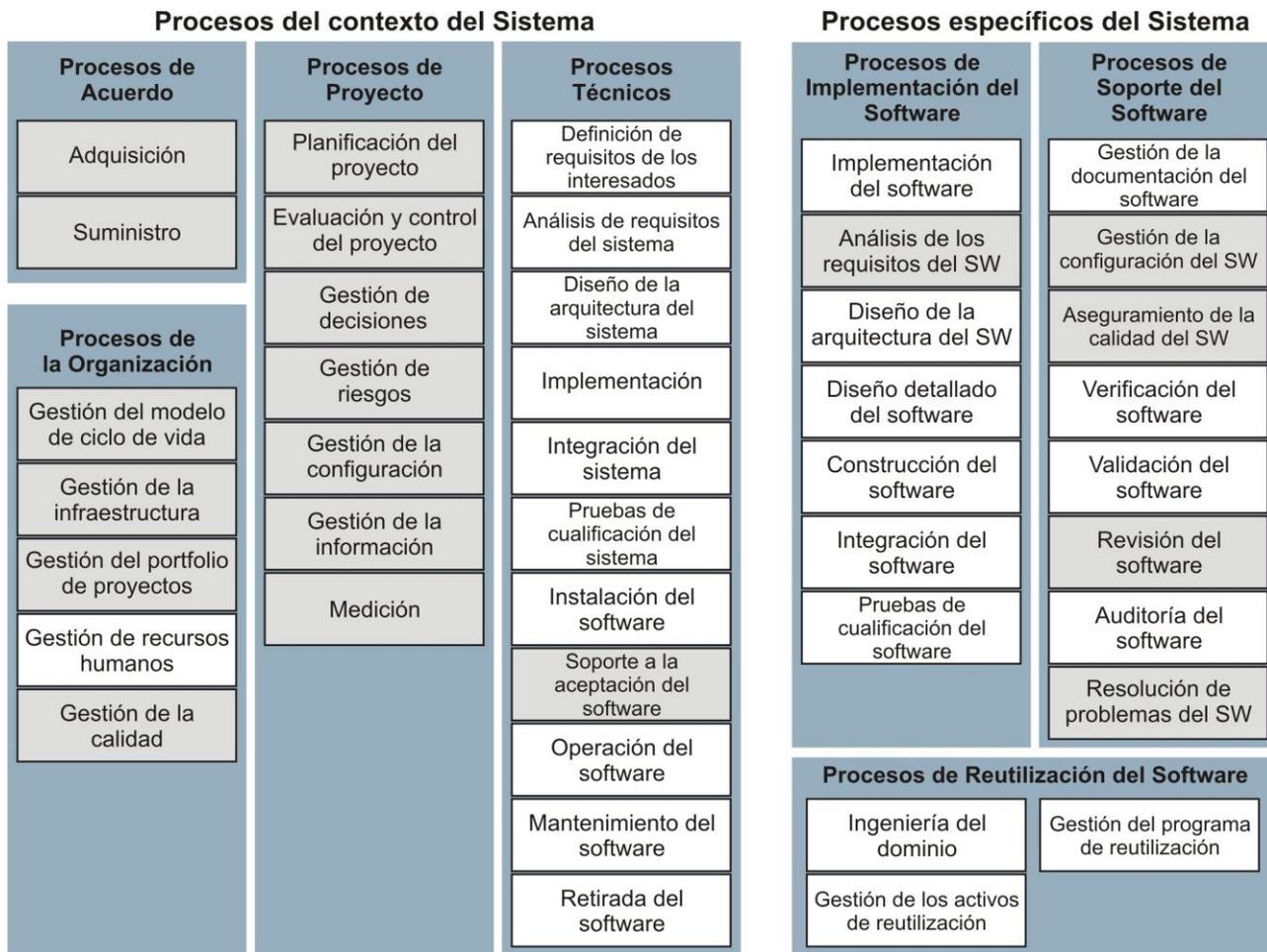
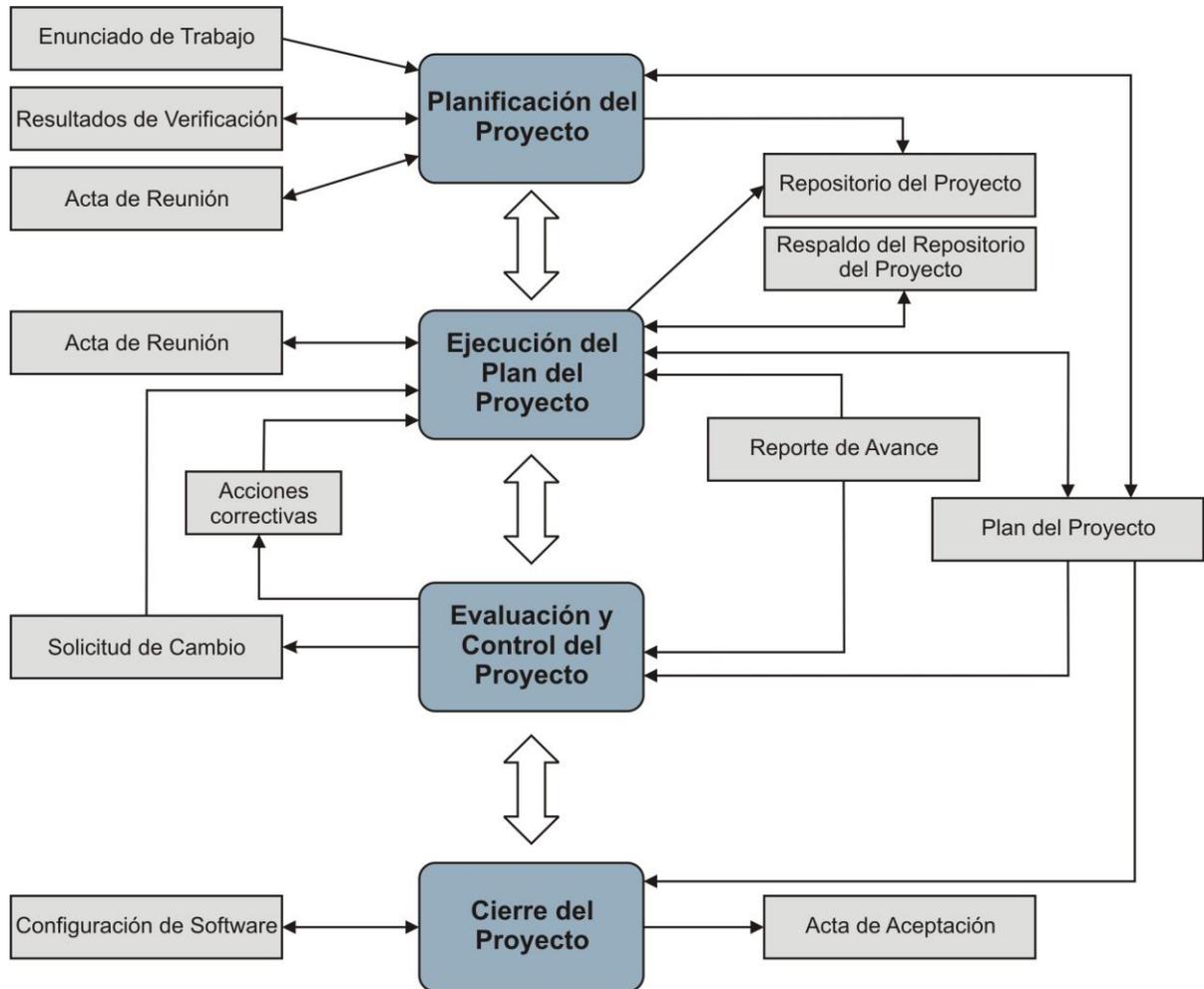


Figura 2.9. Procesos de la norma ISO/IEC 12207 (Mas & Mesquida, 2013).

Por otro lado, el estándar ISO/IEC 29110-5-1-2:2011 define una guía para el desarrollo de software y la gestión de los proyectos que es aplicable a la amplia mayoría de pequeñas y micro empresas que no desarrollan software crítico. Tal y como se indicó en secciones anteriores de este capítulo, esta norma define dos procesos, uno denominado *Implementación del Software*, cuyo propósito es la realización sistemática de las actividades de análisis, diseño, construcción, integración y pruebas de los productos de software y otro, denominado *Gestión de Proyectos*, cuyo propósito es establecer y llevar a cabo de manera sistemática las tareas de implementación de software que permitan cumplir con los objetivos del proyecto en calidad, tiempo y costo esperados.

La Figura 2.10 muestra el flujo de información entre las cuatro actividades que conforman el proceso de *Gestión de Proyectos* y las cuales se identifican como:

- GP.1 Planificación del Proyecto.
- GP.2 Ejecución del Plan del Proyecto.
- GP.3 Evaluación y Control del Proyecto.
- GP.4 Cierre del Proyecto



**Figura 2.10.** Procesos de gestión de proyectos de la norma ISO/IEC 29110-5-1-2 (Mas & Mesquida, 2013).

Considerando ambas normas, los autores han identificado las relaciones que existen entre los estándares con el objetivo de facilitar su comprensión en el entorno de las PyMES. Así, se determinó que la norma ISO/IEC 29110-5-1-2 contiene un subconjunto de únicamente 10 de los 19 procesos relacionados con la gestión de proyectos que define la norma ISO/IEC 12207 (véase columna derecha de la Tabla 11).

**Tabla 11.** Relación entre ISO/IEC 12207 e ISO/IEC 29110 (Mas & Mesquida, 2013).

Objetivos del proceso de Gestión de proyectos de ISO/IEC 29110-5-1-2	Procesos de ISO/IEC 12207
GP.O1 Se desarrolla el <i>Plan de Proyecto</i> de acuerdo al <i>Enunciado del Trabajo</i> y se revisa y	Planificación del proyecto. Medición.

acepta por el Cliente. Se dimensionan y estiman las <i>Tareas</i> y los <i>Recursos</i> necesarios para completar el trabajo.	
GP.O2 Se monitoriza el avance del proyecto contra el <i>Plan de Proyecto</i> y se registra en el <i>Registro de Estado del Avance</i> . Cuando los objetivos del proyecto no son logrados, se realizan las correcciones para resolver los problemas y desviaciones respecto del plan. Se ejecuta el cierre del proyecto para conseguir la aceptación del Cliente documentada en el <i>Registro de Aceptación</i> .	Evaluación y control del proyecto. Medición. Soporte a la aceptación del software. Resolución de problemas del software.
GP.O3 Se atienden las <i>Solicitudes de Cambio</i> mediante su recepción y análisis. Se evalúan los cambios a los requisitos de software por su impacto técnico, en coste y en cronograma.	Análisis de los requisitos del software.
GP.O4 Se realizan <i>Reuniones de Revisión</i> con el Equipo de Trabajo y el Cliente. Se documentan y siguen los acuerdos que surgen de estas reuniones.	Revisión del software.
GP.O5 Se identifican los <i>Riesgos</i> a medida que se desarrollan y durante la realización del proyecto.	Gestión de riesgos. Revisión del software.
GP.O6 Se desarrolla una <i>Estrategia de Control de Versiones</i> de software. Se identifican, definen e incorporan a una línea base los elementos de la <i>Configuración del Software</i> . Se controlan y ponen a disposición del Cliente y del Equipo de Trabajo las modificaciones y liberaciones de los elementos. Se controla el almacenamiento, la manipulación y la entrega de los elementos.	Gestión de la configuración del software.
GP.O7 Se realiza el Aseguramiento de Calidad del Software para proporcionar garantía de que los productos y procesos cumplen con el <i>Plan de Proyecto</i> y la <i>Especificación de Requisitos</i> .	Aseguramiento de la calidad del software.

Adicionalmente, teniendo en mente la simplificación de dichos estándares, los autores analizaron la guía del PMBoK puesto que, como se ha indicado anteriormente, ésta identifica un conjunto de mejores prácticas relacionadas con la gestión de los proyectos y proporciona y promueve un vocabulario común en el ámbito de la profesión.

Las prácticas contenidas en el PMBoK abarcan todo el ciclo de vida de un proyecto, desde que se realiza la propuesta al cliente hasta su entrega, aceptación final y cierre del mismo. Los procesos de gestión de proyectos de este estándar se agrupan en cinco categorías, conocidas como grupos de procesos de la gestión de proyectos, que son: Iniciación, Planificación, Ejecución, Monitorización, y Control y Cierre. Por otra parte, los procesos de gestión de proyectos se agrupan en diez áreas de conocimiento, de hecho cada una de éstas es un capítulo del PMBoK y son:

1. *Gestión de la Integración*. Se refiere a los procesos requeridos para asegurar que los elementos de un proyecto están coordinados apropiadamente. Así, consiste del desarrollo de un plan de proyecto, la ejecución del plan de proyecto, y el control de cambios en general.
2. *Gestión del Alcance*. Se refiere al proceso requerido para asegurar que el proyecto incluye todo el trabajo necesario, y sólo el trabajo necesario, para completar el proyecto de manera exitosa. Así, consiste de la iniciación, planificación del alcance, definición del alcance, verificación del alcance, y control de cambio al alcance.

3. *Gestión del Tiempo*. Se refiere a los procesos requeridos para asegurar la terminación a tiempo del proyecto. Así, consiste de la definición de las actividades, establecimiento de la secuencia de las actividades, estimación de la duración de las actividades, desarrollo del cronograma y control de la calendarización.
4. *Gestión de los Costos*. Se refiere a los procesos requeridos para asegurar que el proyecto es completado dentro del presupuesto aprobado. Así, consiste de la planificación de recursos, la estimación de costos, el presupuesto de costos, y control de los mismos.
5. *Gestión de la Calidad*. Se refiere a los procesos requeridos para asegurar que el proyecto va a satisfacer las necesidades para las cuales fue desarrollado. Así, consiste de la planificación de la calidad, y el aseguramiento y control de la misma.
6. *Gestión de los Recursos Humanos*. Se refiere a los procesos requeridos para hacer el uso más eficiente de las personas involucradas en el proyecto. Así, consiste de la planificación organizacional, la adquisición de staff, y el desarrollo del equipo.
7. *Gestión de las Comunicaciones*. Se refiere a los procesos requeridos para asegurar la generación apropiada y a tiempo, recolección, diseminación, almacenamiento, y la disposición final de la información del proyecto. Así, consiste de la planificación de la comunicación, la distribución de la información, la generación de reportes de desempeño, y el cierre administrativo.
8. *Gestión de los Riesgos*. Se refiere a los procesos concernientes con la identificación, análisis, y respuesta al riesgo del proyecto. Así, consiste en la identificación del riesgo, la cuantificación del riesgo, desarrollo de la respuesta al riesgo, y en el control de la respuesta al riesgo.
9. *Gestión de las Adquisiciones*. Se refiere a los procesos requeridos para adquirir bienes y servicios fuera de la organización ejecutora. Así, consiste en la planificación de la adquisición, la selección de los proveedores, la gestión y cierre de contratos.
10. *Gestión de los Interesados*. Se refiere a los procesos para identificar y gestionar a los interesados del proyecto y sus requerimientos y expectativas.

En este sentido, las Tablas 12 a 15 muestran las relaciones que fueron identificadas entre las tareas de las cuatro actividades del proceso de Gestión de Proyectos de la norma ISO/IEC 291105-1-2 y los procesos de la guía del PMBoK.

**Tabla 12.** Relación las tareas de GP.1 Planificación del Proyecto y PMBoK (Mas & Mesquida, 2013).

Tareas de la actividad GP.1 de ISO/IEC 29110-5-1-2	Procesos de PMBoK
GP.1.1 Revisar el <i>Enunciado del Trabajo</i> .	4.1 Desarrollar el acta de constitución del proyecto 5.1 Planificar la gestión del alcance
GP.1.2 Definir con el Cliente las <i>Instrucciones de Entrega</i> para cada uno de los <i>Entregables</i> especificados en el <i>Enunciado del Trabajo</i> .	5.2 Recopilar los requisitos
GP.1.3 Identificar las <i>Tareas</i> específicas a realizar para producir los <i>Entregables</i> y sus <i>Componentes de Software</i> identificados en el <i>Enunciado del Trabajo</i> . Incluir las <i>Tareas</i> en el proceso de Implementación de Software, junto con las <i>Tareas</i> de verificación, validación y revisiones con el Cliente y el Equipo de Trabajo, para asegurar la calidad de los productos de trabajo. Identificar las <i>Tareas</i> para realizar las <i>Instrucciones de Entrega</i> .	5.4 Crear la Estructura Detallada de Trabajo (EDT) 6.1 Planificar la gestión del cronograma 6.2 Definir las actividades 6.3 Secuenciar las actividades

Documentar las <i>Tareas</i> .	
GP.1.4 Establecer la <i>Duración Estimada</i> para realizar cada tarea.	6.5 Estimar la duración de las actividades
GP.1.5 Identificar y documentar los <i>Recursos</i> : humanos, materiales, equipos y herramientas, estándares, incluyendo la capacitación requerida para que el <i>Equipo de Trabajo</i> pueda realizar el proyecto. Incluir en el cronograma las fechas cuando se vayan a necesitar los <i>Recursos</i> y la capacitación.	6.4 Estimar los recursos de las actividades 9.1 Planificar la gestión de los recursos humanos
GP.1.6 Establecer la <i>Composición del Equipo de Trabajo</i> , asignando los roles y responsabilidades acordes a los <i>Recursos</i> .	6.4 Estimar los recursos de las actividades 9.2 Adquirir el equipo del proyecto
GP.1.7 Asignar las fechas de inicio y fin estimadas a cada <i>Tarea</i> con el fin de crear el <i>Cronograma de las Tareas del Proyecto</i> considerando los <i>Recursos</i> asignados, la secuencia y dependencia de las <i>Tareas</i> .	6.6 Desarrollar el cronograma
GP.1.8 Calcular y documentar el <i>Esfuerzo</i> y <i>Costo estimado</i> del proyecto.	6.5 Estimar la duración de las actividades 7.1 Planificar la gestión de costos 7.2 Estimar los costos 7.3 Determinar el presupuesto
GP.1.9 Identificar y documentar los riesgos que pueden afectar al proyecto.	11.2 Identificar los riesgos
GP.1.10 Documentar la <i>Estrategia de Control de Versiones</i> en el <i>Plan de Proyecto</i> .	4.2 Desarrollar el plan para la dirección del proyecto
GP.1.11 Generar el <i>Plan de Proyecto</i> integrando los elementos previamente identificados y documentados.	4.2 Desarrollar el plan para la dirección del proyecto
GP.1.12 Incluir la <i>Descripción del Producto</i> , el <i>Alcance</i> , los <i>Objetivos</i> y los <i>Entregables</i> en el <i>Plan de Proyecto</i> .	4.1 Desarrollar el acta de constitución del proyecto 5.3 Definir el alcance
GP.1.13 Verificar y obtener aprobación del <i>Plan de Proyecto</i> . Verificar que todos los elementos del <i>Plan de Proyecto</i> son viables y consistentes. Los resultados encontrados son documentados en un <i>Resultado de Verificación</i> y las correcciones son realizadas hasta que el documento es aprobado por el jefe del proyecto.	--
GP.1.14 Revisar y obtener la aprobación del <i>Plan de Proyecto</i> . El cliente revisa y acepta el <i>Plan de Proyecto</i> , asegurándose que los elementos del <i>Plan de Proyecto</i> se corresponden con el <i>Enunciado del Trabajo</i> .	--
GP.1.15 Establecer el <i>Repositorio del Proyecto</i> usando la <i>Estrategia de Control de Versiones</i> .	4.2 Desarrollar el plan para la dirección del proyecto.

**Tabla 13.** Relación las tareas de GP.2 Ejecución del Plan del Proyecto y PMBoK (Mas & Mesquida, 2013).

<b>Tareas de la actividad GP.2 de ISO/IEC 29110-5-1-2</b>	<b>Procesos de PMBoK</b>
GP.2.1 Monitorizar la ejecución del <i>Plan de Proyecto</i> y registrar la información actual en el <i>Informe de Avance</i> .	4.3 Dirigir y gestionar la ejecución del proyecto 4.4 Monitorizar y controlar el trabajo del proyecto
GP 2.2 Analizar y evaluar el impacto en coste, cronograma e impacto técnico de la <i>Solicitud de Cambio</i> . La <i>Solicitud de Cambio</i> puede ser propuesta externamente por el Cliente o internamente por el Equipo de Trabajo. Actualizar el <i>Plan de Proyecto</i> si los cambios aceptados no afectan a los acuerdos con el Cliente. La <i>Solicitud de Cambio</i> que afecta a los acuerdos previos debe ser negociada por ambas partes (véase GP.2.4).	4.5 Realizar el control integrado de cambios
GP.2.3 Realizar reuniones de revisión con el <i>Equipo de Trabajo</i> , identificar problemas, revisar el estado de los riesgos, registrar acuerdos y darles seguimiento hasta su	4.4 Monitorizar y controlar el trabajo del proyecto 11.6 Controlar los riesgos

conclusión.	
GP.2.4 Realizar reuniones de revisión con el Cliente, registrar acuerdos y dares seguimiento hasta su conclusión. La <i>Solicitud de Cambio</i> propuesta por el Cliente o por el Equipo de Trabajo, que afecte al Cliente, debe ser negociada para alcanzar un acuerdo entre ambas partes. Si es necesario, se debe actualizar el <i>Plan de Proyecto</i> conforme a los nuevos acuerdos con el Cliente.	5.6 Controlar el alcance
GP.2.5 Realizar el <i>Respaldo del Repositorio del Proyecto</i> de acuerdo a la <i>Estrategia de Control de Versiones</i> .	--
GP.2.6 Realizar la recuperación del <i>Repositorio del Proyecto</i> utilizando el <i>Respaldo del Repositorio del Proyecto</i> , en caso de ser necesario.	--

**Tabla 14.** Relación las tareas de GP.3 Evaluación y Control del Proyecto y PMBoK (Mas & Mesquida, 2013).

Tareas de la actividad GP.3 de ISO/IEC 29110-5-1-2	Procesos de PMBoK
GP.3.1 Evaluar el progreso del proyecto con respecto al <i>Plan de Proyecto</i> , comparando: <ul style="list-style-type: none"> <li>Tareas reales contra las <i>Tareas</i> planificadas.</li> <li>Resultados reales contra los <i>Objetivos</i> del proyecto establecidos.</li> <li>Recursos reales asignados contra los <i>Recursos</i> planificados.</li> <li><i>Costos</i> reales contra el presupuesto estimado.</li> <li>Tiempo real utilizado contra el programado.</li> <li><i>Riesgos</i> reales contra los identificados previamente.</li> </ul>	4.4 Monitorizar y controlar el trabajo del proyecto 5.6 Controlar el alcance 6.7 Controlar el cronograma 7.4 Controlar los costos
GP.3.2 Establecer acciones para corregir desviaciones o problemas y riesgos identificados que amenacen el cumplimiento del plan. En caso de ser necesario, documentarlas en el <i>Registro de Acciones Correctivas</i> y darles seguimiento hasta su conclusión.	5.6 Controlar el alcance 6.7 Controlar el cronograma 7.4 Controlar los Costos 11.2 Identificar los riesgos
GP.3.3 Identificar cambios en requisitos y/o en el <i>Plan de Proyecto</i> para hacer frente a desviaciones importantes, riesgos potenciales o problemas relativos al cumplimiento del plan; documentarlos en una <i>Solicitud de Cambio</i> y darles seguimiento hasta su conclusión.	4.5 Realizar el control integrado de cambios

**Tabla 15.** Relación las tareas de GP.4 Cierre del Proyecto y PMBoK (Mas & Mesquida, 2013).

Tareas de la actividad GP.4 de ISO/IEC 29110-5-1-2	Procesos de PMBoK
GP.4.1. Formalizar la conclusión del proyecto de acuerdo a las <i>Instrucciones de Entrega</i> establecidas en el <i>Plan de Proyecto</i> , proporcionando el apoyo para su aceptación y obteniendo el <i>Acta de Aceptación</i> firmada.	5.5 Validar el alcance 4.6 Cerrar el proyecto o fase
GP.4.2 Actualizar el <i>Repositorio del Proyecto</i> .	4.6 Cerrar el proyecto o fase

#### 2.4.4.3. Resultados

Como consecuencia del análisis realizado en esta investigación, los autores obtuvieron un conjunto de recomendaciones para las PyMES que pretenden implantar los procesos para la gestión de los proyectos en sus labores cotidianas y que no han tenido experiencia alguna con estas actividades. Estas recomendaciones se resumen de la siguiente manera:

- Se concluyó, en primer lugar, que la norma ISO/IEC 29110-5-1-2 considera y cubre completamente las mejores prácticas de los procesos de las siguientes cuatro áreas de conocimiento de PMBoK:

- Gestión de la Integración. En esta área se encuentran los procesos y actividades relacionadas con la gestión, como son: constituir formalmente un proyecto, planificarlo, monitorizarlo, controlarlo y dirigirlo hasta su cierre. También se determinó que la Gestión de la Integración es un área transversal y básica, no importando el nivel de gestión aplicado. Por lo tanto, el estándar específico para pequeñas y micro empresas incluye una referencia a dichas actividades.
- Gestión del Alcance, Gestión del Tiempo y Gestión de los Costos. Tampoco resulta sorprendente que la investigación haya indicado que la norma ISO/IEC 29110-5-1-2 considere los procesos relacionados con la triple restricción: Definir el trabajo requerido para completar el proyecto, subdividirlo en componentes más pequeños (o bien las actividades del proyecto), realizar el cronograma y estimar tanto los recursos necesarios para llevarlas a cabo como el costo del proyecto.
- En segundo lugar, se observó que la norma ISO/IEC 29110-5-1-2 cubre, aunque no en su totalidad, algunas de las mejores prácticas recomendadas en otras dos áreas de conocimiento de PMBoK:
  - Gestión de los Recursos Humanos. La norma ISO/IEC 29110-5-1-2 contempla la identificación de los roles, responsabilidades y habilidades requeridas para los perfiles necesarios del equipo del proyecto, así como la selección de las personas más adecuadas a cada perfil. Sin embargo, no cubre otras tareas propias de la gestión de los recursos humanos contempladas en PMBoK, como son la capacitación de las personas y el control del rendimiento de los miembros del equipo del proyecto.
  - Gestión de los Riesgos. La norma ISO/IEC 29110-5-1-2 considera la identificación de los riesgos que puedan afectar al proyecto, así como el control de los mismos durante la ejecución del proyecto, pero esto de manera muy general, sin mencionar específicamente tareas relacionadas con el análisis cuantitativo y cualitativo que si se especifican dentro de PMBoK.
- Finalmente, las mejores prácticas de la gestión de los proyectos de las cuatro áreas de conocimiento restantes del PMBoK, no son consideradas por la norma ISO/IEC 29110-5-1-2:
  - Gestión de la Calidad. Si bien la norma ISO/IEC 29110-5-1-2 considera como aspectos relativos al aseguramiento de calidad, la verificación, validación y revisión de los productos obtenidos, el área de Gestión de la Calidad de PMBoK se refiere a los procesos y actividades que establecen responsabilidades, objetivos y políticas de calidad para la mejora continua de los procesos llevados a cabo durante todo el proyecto.
  - Gestión de las Comunicaciones. La norma no incluye la identificación de las necesidades de información de los interesados en el proyecto ni la definición del plan de comunicación entre todos los interesados en el proyecto.
  - Gestión de las Adquisiciones. La norma no cubre los procesos relacionados con la compra o adquisición de los productos, servicios o resultados que es necesario obtener fuera del equipo del proyecto.
  - Gestión de los Interesados. La norma no trata ningún aspecto relacionado con la identificación de las personas u organizaciones afectadas por el proyecto, ni con la documentación de información relevante relativa a sus intereses, participación e impacto en el éxito del proyecto.

## 2.4.5. Un marco de trabajo para la gestión ágil de proyectos en PyMES

### 2.4.5.1. Objetivo

La investigación de (O'Sheedy & Sankaran, 2013) argumenta que los modelos y estándares actuales orientados a la gestión de los proyectos son frecuentemente percibidos por las PyMES como complicados y excesivamente burocráticos; algo no deseable considerando el tiempo limitado o el bajo presupuesto del que disponen estas empresas para realizar sus proyectos. Así, los autores vislumbran al desarrollo ágil como una posible solución al problema de uso de los métodos “excesivamente” complejos recomendados por los modelos y estándares y, en base a éste, presentan un marco ágil de trabajo compuesto por cuatro fases: inicio (centrada en el establecimiento de los objetivos), planificación (enfocada en el *backlog*<sup>17</sup> del proyecto), ejecución y control (vigiladas a través de ciclos iterativos del proyecto), y cierre (dirigida a los entregables del proyecto). Adicionalmente, esta investigación resume algunas sugerencias relacionadas con el éxito en el desarrollo de un marco de trabajo para la gestión ágil de los proyectos.

### 2.4.5.2. Descripción

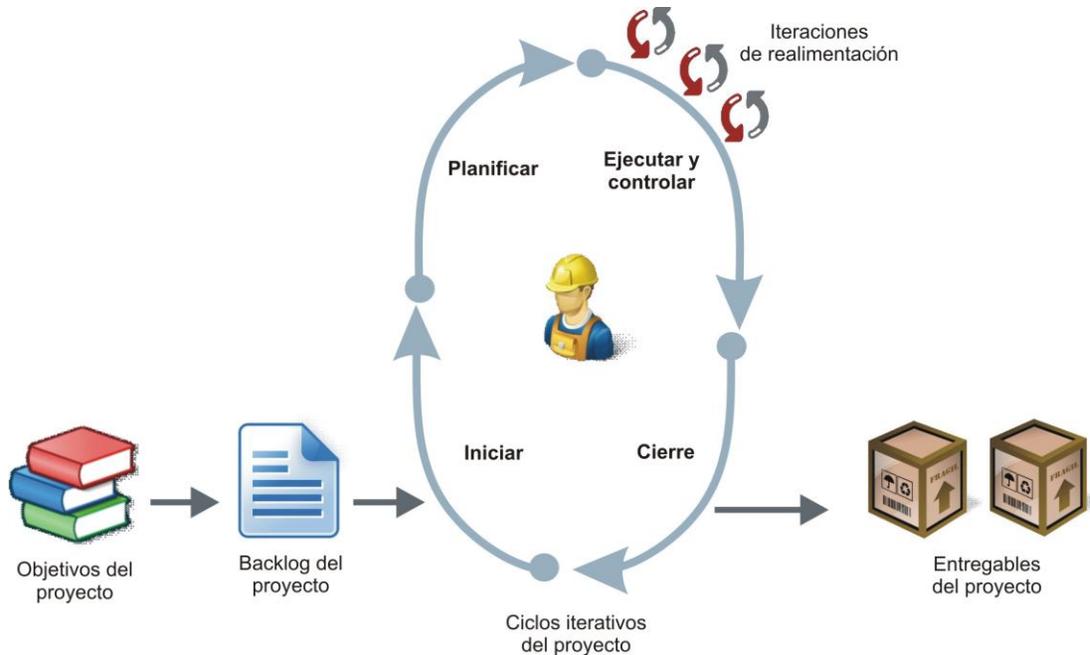
La propuesta de los autores combina los fundamentos de la gestión tradicional de proyectos con herramientas y técnicas modernas del desarrollo ágil. Así, la guía del PMBoK fue elegida como el fundamento teórico de los procesos y procedimientos para la gestión de los proyectos (es decir el cómo de la práctica), y los componentes ágiles, principalmente de Scrum y eXtreme Programming, fueron utilizados para representar el qué. La integración de ambos permitió la creación de un marco de trabajo para la gestión ágil de los proyectos (véase Figura 2.11), que fue probado en un entorno real de TI a través de dos PyMES.

A continuación se describen cada una de las fases de este marco de trabajo:

- La fase de inicio establece los objetivos del proyecto. De acuerdo a los autores, si se tratase de un proyecto grande estos objetivos deberían plasmarse en el *acta constitutiva del proyecto*. En este sentido este artefacto, que también es un componente necesario para el éxito de un proyecto ágil aunque puede seguir un formato ligeramente diferente, debe contener mínimamente tres componentes:
  - La *visión*, que define por qué se está realizando el proyecto. Estos motivos pueden responder a una necesidad de mejora tanto del negocio como de los procesos de TI, o bien simplemente son la acción realizada con el fin de mejorar la eficiencia de una organización.
  - La *misión*, que establece el trabajo necesario para lograr la visión del proyecto, y
  - Los *criterios de éxito* que establecen de manera formal la definición de “éxito de los proyectos” a través de la colaboración con los principales interesados (*stakeholders*).

---

<sup>17</sup> En el contexto de la terminología de un marco ágil de trabajo, como Scrum por ejemplo, el *backlog* representa una estructura de pila que resume una lista de tareas u objetivos que deben ser satisfechos durante cada iteración (Salo & Abrahamsson, 2008).



**Figura 2.11.** El marco de trabajo para la gestión ágil de los proyectos (O’Sheedy & Sankaran, 2013).

- La fase de planificación del proyecto. Durante esta fase el proyecto es planificado y se establece la secuencia de las tareas necesarias para desarrollarlo. En este sentido, el primer paso de la fase consiste en la recopilación de los requisitos para el proyecto a través de la interacción con las partes interesadas. Así, el alcance del proyecto es construido utilizando estos requisitos. Este alcance es una descripción detallada del proyecto y lista el trabajo requerido para satisfacer los requerimientos del proyecto. El siguiente paso de este proceso se enfoca en la creación del *backlog* del proyecto, es decir una descomposición del trabajo que será necesario para cumplir con el alcance establecido. Este *backlog* puede verse como una lista de “fragmentos de trabajo” que están sin terminar, pero que tienen una prioridad asignada para dicho fin. La meta es centrarse, en primer lugar, en el trabajo con la prioridad más alta y proporcionar el mejor valor para el esfuerzo dedicado a cada iteración. Algunas tareas podrían realizarse antes que otras en la línea de tiempo, y existirán dependencias, por lo tanto, que también deberán de considerarse. En este sentido, los autores recomiendan el uso de herramientas como la WBS, para la descomposición del trabajo en tareas con mayor detalle, y Microsoft Project® para gestionar la línea de tiempo.
- Las fases de ejecución y control del proyecto. Ambas fases se enfocan en realizar las actividades del proyecto y conducir revisiones constantes con el objetivo de asegurar que el trabajo está siendo realizado de manera apropiada. De acuerdo a los argumentos de los autores, la fase de ejecución se basa en las actividades tradicionales de un proceso de investigación que consta de cuatro etapas: identificar, planificar, actuar y evaluar. En esta el *backlog* del proyecto es utilizado como entrada para cada ciclo de ejecución. Así, un fragmento del trabajo del *backlog* es tomado por un equipo de trabajo, quien identifica su objetivo final. Posteriormente, se determina el trabajo necesario para lograr dicho objetivo y se formula un plan de acción que el equipo debe ejecutar. Durante esta ejecución, el equipo debe reunirse periódicamente para discutir el progreso y asegurarse de que el proyecto va en la dirección correcta. Estas reuniones aseguran que el equipo se mantiene enfocado en el objetivo de salida del ciclo, y permite ajustes menores cuando se producen cambios en el entorno del proyecto. Dependiendo de la cantidad de esfuerzo requerida para completar una

parte del trabajo, un ciclo puede durar días o semanas, y será necesario ajustar el tiempo de las iteraciones. Una vez que el trabajo se ha completado, es necesario evaluar el resultado. Si existiera una desviación en el proyecto sobre sus objetivos a cumplir, se asume que se ha presentado un problema imprevisto o que se ha producido un cambio en el negocio durante la ejecución del proyecto, y esto debe realimentarse en los objetivos y *backlog* del proyecto (dependiendo del tamaño y alcance de los cambios necesarios). Estos ciclos de trabajo pueden ocurrir de forma serial, o incluso simultáneamente si varios equipos están trabajando en diferentes partes del proyecto. Así, la característica clave en las fases de ejecución y control es que deben existir múltiples iteraciones de realimentación para controlar el progreso del proyecto. Esto permite que el equipo pueda evaluar dos aspectos importantes: (1) si el trabajo realizado está alineado con los objetivos del proyecto, es decir si el equipo determina que sus esfuerzos se desviaron del objetivo del proyecto, entonces será necesario que dicho esfuerzo sea redirigido hacia la meta original; (2) si el entorno en el que se está produciendo el proyecto ha cambiado significativamente, de tal forma que el objetivo del proyecto ya no es pertinente, entonces el trabajo tendrá que ajustarse a un objetivo más relevante. En este sentido, el enfoque ágil difiere de muchos métodos tradicionales para la gestión de los proyectos, particularmente en la forma en que los cambios son realizados (incluso en etapas avanzadas del ciclo); sin embargo, es importante reconocer que ya sea a través de un enfoque ágil o tradicional, todo cambio genera un costo adicional.

- La fase de cierre. En esta última fase el equipo de trabajo compara, una vez más, los objetivos del proyecto con el trabajo entregado. Esto asegura que el proyecto comprometido ha sido liberado y aceptado por las partes interesadas. Si existieran algunos resultados imprevistos del proyecto, u otros problemas, éstos deben tratarse y rectificarse. Además, el equipo de trabajo deberá documentar las lecciones aprendidas en el proyecto, de tal forma que el conocimiento sea almacenado para que equipos de otros proyectos futuros sean beneficiados.

### 2.4.5.3. Resultados

La implementación de este marco de trabajo para la gestión ágil de los proyectos evidenció beneficios importantes para las PyMES que participaron en el caso de estudio. Uno de los beneficios fue, por ejemplo, que dicho marco ayudó a facilitar la introducción de cambios en los requisitos del proyecto, incluso en etapas avanzadas del proyecto. También, este marco proporcionó un conjunto flexible y no burocrático de procesos para la gestión de los proyectos en el entorno PyME, al mismo tiempo que proporcionó los procesos adecuados para guiar y controlar los proyectos. Así, los autores lograron resumir una lista de herramientas y procedimientos que pueden usarse con tal fin (véase Tabla 1).

**Tabla 16.** Herramientas y procedimientos usados por el marco de gestión ágil (O'Sheedy & Sankaran, 2013).

Fase del Proyecto	Herramientas y procedimientos implementados
Inicio	Acta constitutiva del proyecto
Planificación	Alcance del proyecto <i>Backlog</i> del proyecto Tablero de tareas para desarrollo ágil WBS Gráficas de Gantt
Ejecución	Tablero de tareas para desarrollo ágil Reuniones de actualización sobre desarrollo ágil

Monitorización y Control	Graficas de tipo <i>burn down</i> <sup>18</sup>
Cierre	Base de conocimientos con lecciones aprendidas

De acuerdo a sus creadores, el marco ágil ayudó a las empresas a finalizar exitosamente los proyectos, al equipo de trabajo a responder rápidamente a los cambios a medida que éstos surgían, y a gestionar la complejidad de un proyecto en un entorno cambiante. Adicionalmente, se argumenta que este marco ágil es capaz de ayudar a mejorar el flujo y la gestión de los proyectos, especialmente en situaciones donde existe un alto grado de incertidumbre y no se conocen todos los factores que pueden afectar al proyecto. En este sentido, se hace la recomendación de que dicho marco no sea considerado un manual de proceso, es decir, su propósito principal no es prescribir el camino “correcto” para gestionar los proyectos. Más bien, debe ser visto como un sistema flexible para la gestión de los proyectos, que es adaptable y sensible a los cambios del entorno.

En resumen, los investigadores han planteado una serie de sugerencias relacionadas con su propuesta y la forma de implementar un enfoque ágil dentro de las PyMES, las cuales se pueden resumir de la siguiente manera:

1. *Existen beneficios reales que se pueden obtener a partir de la implementación de un estilo de gestión ágil de los proyectos:* Diferentes niveles de un proceso pueden aplicarse a un proyecto cuando se implementa un método ágil de gestión. En este sentido, es posible cambiar el nivel de los procesos si se utiliza un marco adaptable de gestión ágil de los proyectos. Por lo tanto, los procesos pueden reducirse en cantidad y complejidad si se trata de proyectos pequeños y no complejos, mientras que para proyectos más complejos se puede utilizar una combinación avanzada de PMBoK y procesos ágiles.
2. *La gestión ágil de los proyectos necesita el soporte de los dueños del negocio, especialmente en el entorno de las PyMES:* La gestión ágil de los proyectos difiere de los métodos estándar de gestión en muchos aspectos. Debido a la naturaleza innovadora de este tipo de gestión, si el dueño del negocio no apoya totalmente la introducción de los procesos para la gestión de los proyectos, entonces existe una mayor probabilidad de que se presenten problemas de aceptación dentro de la organización. Los problemas de aceptación por lo general se deben a que los empleados no pueden percibir el beneficio o la necesidad de usar procesos diferentes. A pesar de que dicho dueño no puede oponerse activamente a los nuevos procesos, si no los apoyan completamente éstos pueden fallar debido a la falta de compromiso. Además, dado que la gestión de los proyectos requiere a menudo recursos adicionales para su implantación, si la alta dirección de una empresa no los proporciona, o si algunas partes de la compañía se oponen activamente a la introducción de los nuevos procesos, el jefe de proyectos encontrará dificultades en la introducción de los procedimientos y procesos adecuados para poner en práctica un estilo ágil de gestión.
3. *La gestión ágil de los proyectos es más adecuada para entornos muy pequeños:* En base a los resultados obtenidos en el estudio realizado por los autores, se argumenta que un marco de gestión ágil funciona mejor en el contexto de los pequeños y medianos proyectos realizados en un entorno PyMES. Sin embargo, se argumenta también que a medida que los proyectos crecieron en complejidad, el marco empezó a mostrar varias deficiencias. Por ejemplo, el seguimiento de los proyectos grandes y complicados puede ser más difícil a través de un enfoque ágil (e.g., la coordinación de los horarios de trabajo cuando existe un

<sup>18</sup> Una gráfica *burn down* o diagrama de quemado es una representación gráfica del trabajo por hacer en un proyecto dentro de una línea de tiempo. Usualmente el *backlog* se muestra en el eje vertical y el tiempo en el eje horizontal. Así, el diagrama representa una serie temporal del trabajo pendiente y predecir cuándo será completado (Salo & Abrahamsson, 2008).

número importante de elementos participando en un proyecto). Esta es una demostración de las repetidas preocupaciones de los especialistas sobre la poca efectividad de un enfoque ágil cuando la escalabilidad de un proyecto aumenta su complejidad. También se observó que los equipos tradicionalmente más grandes son más resistentes al cambio, y puede ser necesario que los métodos ágiles sean introducidos, al principio, a través de objetivos más conservadores.

4. *El enfoque ágil no es la panacea y necesita ser utilizado como una herramienta en el entorno adecuado:* Los autores recalcan que aunque la gestión de proyectos ágil es una herramienta útil, no puede resolver todos los problemas, y no es adaptable a cualquier situación. Uno de los problemas que los jefes de proyectos enfrentan son las muchas prioridades que compiten por su atención, y aunado a esto si este jefe no posee la habilidad para comunicar sus solicitudes de recursos, el problema se agrava aún más. Es decir, sin la formación adecuada el enfoque ágil carece de efectividad.
5. *El marco ágil de proyectos debe ajustarse a la estructura y cultura de la empresa para trabajar de forma eficiente:* Una recomendación importante se relaciona con las condiciones bajo las cuales sería idóneo implementar un enfoque ágil. En este sentido, a pesar de que la gestión ágil de los proyectos es flexible y adaptable, el marco de trabajo propuesto no puede ser implementado de la misma forma en todas las empresas, sino que debe ser ajustado a la estructura y cultura de éstas para tener mayor impacto. Además, se hace énfasis en que este estilo de gestión será más difícil de introducir si la empresa se encuentra dentro de un contexto que es rígido y burocrático.
6. *El marco de trabajo debe ser personalizado para incorporar elementos ágiles y de PMBoK:* Como un componente del marco ágil, el PMBoK es también un marco valioso para la gestión de los proyectos, pero existe la necesidad de que este marco sea personalizado para incluir el conocimiento “suficiente” para obtener una gestión exitosa. Así, el nivel de detalle requerido para los procesos de la gestión de los proyectos depende del nivel de complejidad dentro de un proyecto. Es decir, un proyecto que es simple en su ámbito de aplicación y ejecución, no tendrá los mismos controles y procesos que se requieren para un proyecto que es más complejo o difícil de ejecutar. En la medida que el marco ágil de gestión de proyectos es adaptable, también lo puede ser el nivel de los procesos que se aplican al proyecto. Por lo tanto, los procesos de control pueden ser simples y limitados en los proyectos pequeños y no complejos. Sin embargo, en proyectos más complejos y difíciles puede ser necesario aplicar a las tareas de gestión un rango más amplio de actividades del PMBoK y procesos ágiles.
7. *Los equipos pequeños trabajan mejor en un entorno ágil, aunque los equipos puedan escalar para proyectos más grandes mediante la adición de equipos extras:* Esta investigación determinó que en los entornos de las PyMES, en donde se trabaja con equipos pequeños y proyectos pequeños, los miembros del equipo pueden realizar varias tareas que corresponden a varias funciones y a su vez suelen estar involucrados en la ejecución de varias etapas del proyecto. En este sentido, es muy importante la coherencia en el equipo para completar con éxito el proyecto, algo que regularmente es conseguido gracias a los esfuerzos y habilidades de todos los integrantes. Así pues, si el equipo del proyecto tiene un promedio de alta calidad en sus miembros, es posible lograr un mayor nivel de éxito en la consecución de los proyectos que con solamente uno o dos miembros altamente capacitados.
8. *Para llegar a ser jefe de proyectos se requiere de muchas horas de experiencia y formación:* Uno de los aspectos cruciales para llegar a ser jefe de proyectos es la experiencia adquirida con el paso del tiempo a través de la práctica. Es decir, para ser capaz de tomar decisiones,

un jefe de proyectos tiene el respaldo de muchas horas de práctica que le proporcionan el conocimiento de experto a través de una combinación de experiencia práctica y el aprendizaje formal.

9. *El estilo de gestión del jefe de proyectos necesita empatar con el proyecto, y si es necesario capacitarse para ocupar este rol:* Un jefe de proyectos debe poseer la formación adecuada para incrementar la posibilidad de éxito del proyecto. Además, la personalidad y el estilo de gestión del jefe del proyecto tienen que coincidir con el estilo del proyecto, de lo contrario la tasa de éxito se verá afectada negativamente cuando se intente aplicar un estilo ágil de gestión. Es decir, el jefe del proyecto necesita las habilidades que le permitan investigar las herramientas tradicionales para la gestión de los proyectos, y ponerlas en práctica de manera equilibrada con los procesos ágiles de desarrollo. A través de estos procesos estos jefes serán capaces de determinar el mejor curso de acción para un proyecto en particular, y serán capaces de ofrecer también una estrategia personalizada de gestión para cada entorno del proyecto. Así pues, además de poseer una amplia experiencia en la gestión de proyectos, el jefe de proyectos requiere del conocimiento del dominio (e.g., los expertos en un dominio están mejor preparados para tomar decisiones en entornos que están menos estructurados, lo cual es una situación común en la aplicación de los métodos ágiles).

#### **2.4.6. Gestión de proyectos de software en microempresas usando ISO/IEC 29110**

##### **2.4.6.1. Objetivo**

La investigación de O'Connor y Laporte (2012) analiza el papel y la estructura de la gestión de proyectos en el estándar *ISO/IEC 29110 - Ciclos de vida del Proceso de Software para Entidades muy Pequeñas*, así como su implicación práctica. Concretamente, este trabajo se enfoca en el diseño y desarrollo de la documentación de soporte para la gestión de proyectos y su uso en la implementación de dicho estándar.

##### **2.4.6.2. Descripción**

El estándar ISO/IEC 29110 (ISO/IEC, 2011) está orientado a abordar los problemas que ocasionan el fracaso de un proyecto (e.g., planificación, estimación, control de cambios, control de calidad) y las necesidades específicas de las microempresas, además procura hacer frente a las cuestiones que originan su pobre adopción en las pequeñas empresas. Como se ha mencionado anteriormente, el enfoque utilizado para el desarrollo de ISO/IEC 29110 inició con el estándar internacional ISO/IEC 12207, desarrollado previamente y que está dedicado al establecimiento de un ciclo de vida para el desarrollo de software. Así pues, el desarrollo de este nuevo estándar consistió de tres etapas: (1) seleccionar un subconjunto de procesos del ISO/IEC 12207 aplicable a los pequeños entornos de desarrollo que usualmente no rebasan los 25 empleados; (2) adaptar dicho subconjunto para que se ajuste a las necesidades de las microempresas; y (3) desarrollar guías para las microempresas de desarrollo.

De acuerdo a los autores, los requerimientos básicos que un proceso de desarrollo de software debe cumplir son que éste debe ajustarse a las necesidades del proyecto y además debe contribuir con el éxito del mismo. En este sentido, este proceso debe ser informado por el contexto situacional en el que el proyecto opera y, por lo tanto, ser más contingente en dicho contexto (Clarke & O'Connor, 2012). Así, el *tamaño* es la característica situacional más importante de las entidades objetivo del estándar ISO/IEC 29110, sin embargo existen otros aspectos y características de las microempresas que pueden afectar la preparación o la selección de un perfil, tales como: el modelo de negocio (comercial, contratación, desarrollo interno, etc.), factores situacionales (como la criticidad, incertidumbre, etc.), y niveles de riesgo. La creación de un perfil para cada posible

combinación de valores de las varias dimensiones mencionadas por estos autores, daría como resultado un conjunto inmanejable de perfiles. Esta investigación afirma que los perfiles de las microempresas están agrupados de tal manera que son aplicables a más de una categoría. La Tabla 17 ilustra un grupo que contiene tres perfiles (con las etiquetas A, B y C) que son mapeados a nueve combinaciones de modelos de negocio y factores situacionales.

**Tabla 17.** Asignación de características de la microempresa a un grupo de perfiles (O’Connor & Laporte, 2012).

Modelos de negocio	Factores de perfil situacional		
	Crítico	Incertidumbre de usuario	Cambio en el entorno
Contratación	Perfil A	Perfil A	Perfil A
Desarrollo interno	Perfil B	Perfil B	Perfil A
Comercial	Perfil C	Perfil A	Perfil A

Estos grupos representan una colección de perfiles que están relacionados ya sea por la composición de procesos (i.e., actividades, tareas), o por el nivel de capacidad, o ambos. Una de las principales contribuciones de O’Connor y Laporte es la definición de un grupo “genérico” de perfiles que es aplicable a la vasta mayoría de microempresas que no desarrollan software crítico y cuentan con factores situacionales típicos. En este contexto, la Tabla 18 muestra a este grupo como una colección de cuatro perfiles que proporcionan un enfoque progresivo para satisfacer los requerimientos del grupo de perfiles.

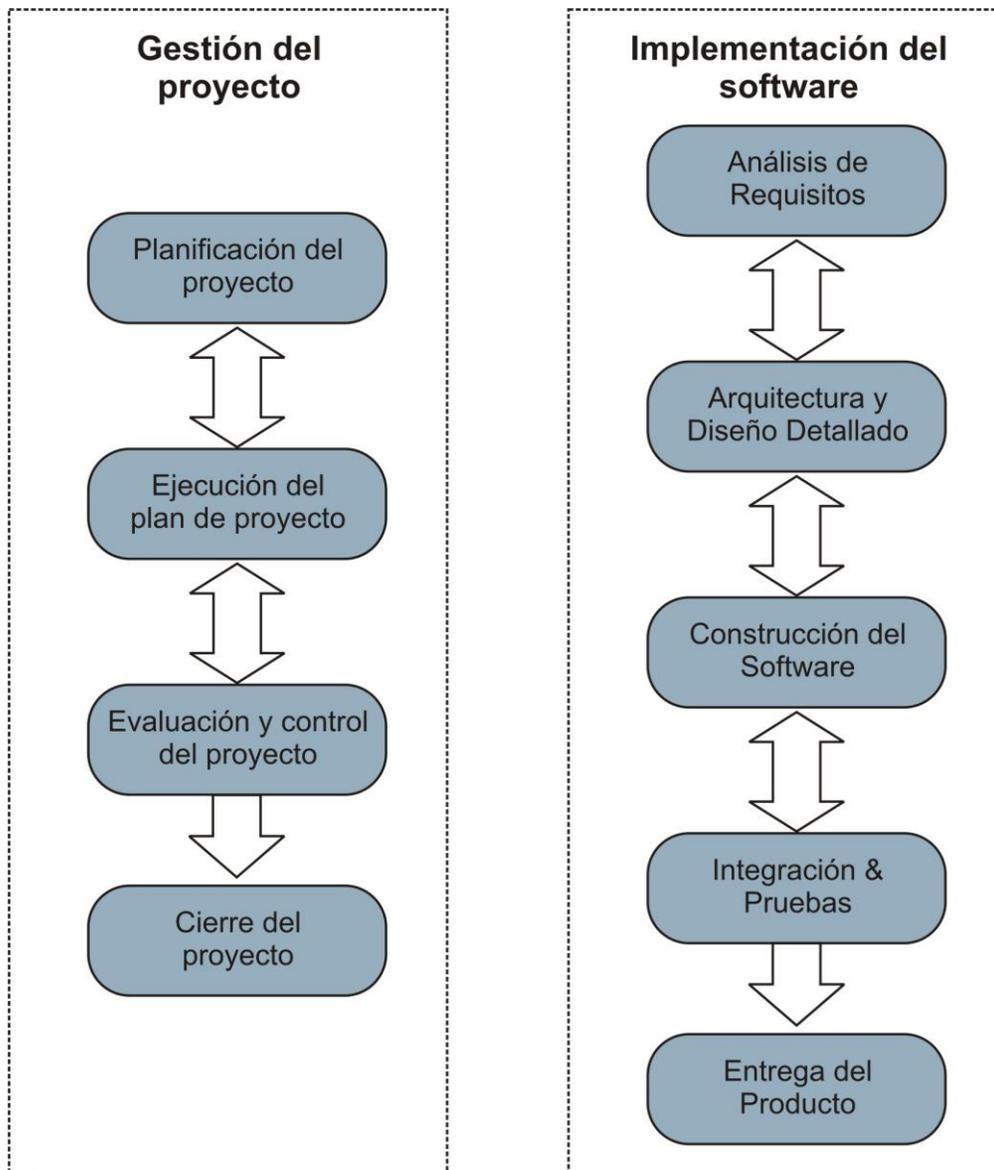
**Tabla 18.** Perfil graduado del grupo de perfiles genérico (O’Connor & Laporte, 2012).

Entrada	Factores de perfil situacional		
	Básico	Intermedio	Avanzando

Hasta la fecha, la finalidad de este perfil básico es definir un guía para el desarrollo de software y la gestión de los proyectos con el objetivo de realizar un proyecto a la vez. En este sentido, la parte medular del estándar es la Guía de Gestión e Ingeniería (ISO/IEC 29110-5) que se enfoca en la *Gestión de Proyectos* y la *Implementación de Software*, procesos que se muestran en la Figura 2.12. El propósito del proceso para la *Gestión de Proyectos* es establecer y realizar de manera sistemática las tareas de implementación de un proyecto de software, de tal forma que cumpla con los objetivos del proyecto en términos de calidad, tiempo y costo. Para esto, la *Gestión de Proyectos* genera un *Plan del Proyecto* que dirige el proyecto de software. Durante la ejecución del proyecto las *Peticiones de Cambio* pueden causar revisiones al *Plan del Proyecto*. El proyecto es sujeto a revisión por la *Evaluación y Control de Proyectos* durante la vida del proyecto, hasta que la *Implementación del Software* es completada y ocurre el *Cierre del Proyecto*. La *Implementación de Software* produce un sistema específico de software implementado como un servicio o producto. Este proceso inicia con el establecimiento de los *Requerimientos del Software*, después se producen la *Arquitectura y el Diseño Detallado* del mismo. Así, el software es *Construido* y verificado usando procedimientos de *Integración y Pruebas*. En la etapa final el *producto* es entregado al cliente.

Dentro del estándar ISO/IEC 29110, el propósito de la *Gestión de Proyectos* consiste en establecer y realizar de manera sistemática las *Tareas* del proyecto para la implementación del software, lo que permite cumplir con los objetivos del proyecto en términos de la calidad esperada, y los tiempos y costos establecidos. Teniendo este objetivo en mente es que se pretende que este estándar sea usado por las microempresas para establecer procesos para implementar cualquier

enfoque de desarrollo o metodología considerando sus necesidades o las del proyecto, por ejemplo, los enfoques ágil, evolutivo, incremental, desarrollo guiado por pruebas (TDD, por sus siglas en inglés), etc.



**Figura 2.12.** Diagramas de procesos del perfil básico de ISO/IEC 29110 (O'Connor & Laporte, 2012).

Siguiendo esta lógica, los objetivos del proceso de *Gestión de Proyectos*, de acuerdo al estándar ISO/IEC 29110-5-1-2, son los siguientes:

- Desarrollar el *Plan del Proyecto* para la ejecución del proyecto de acuerdo con el *Enunciado de trabajo*. Dicho plan debe ser revisado y aceptado por el *Cliente*, para posteriormente determinar y estimar las *Tareas* y los *Recursos* necesarios para completar el trabajo.
- Monitorizar el progreso del proyecto contra el *Plan del Proyecto* y actualizar el *Registro del Estado del Proyecto* de acuerdo a lo observado. Es necesario realizar las correcciones para remediar los problemas y desviaciones en el plan cuando los objetivos del proyecto no son cubiertos. Por último, realizar el cierre del proyecto para obtener la aceptación del *Cliente* y documentarla en el *Registro de Aceptación*.

- Abordar las *Peticiones de Cambio* a través de su recepción y análisis. En este sentido, es primordial evaluar los cambios a los requisitos del software para determinar su costo, tiempo (afectación al calendario), e impacto técnico.
- Sostener reuniones de revisión con el *Equipo de Trabajo* y el *Cliente* y registrar y monitorizan los acuerdos alcanzados.
- Identificar y vigilar los riesgos durante la realización del proyecto.
- Desarrollar una *Estrategia de Control de Versiones del software*, donde los elementos de la *Configuración del Software* sean identificados, definidos y establecidos como línea base. Adicionalmente, es necesario controlar y poner a disposición del *Cliente* y *Equipo de Trabajo* las diferentes versiones generadas en el proyecto.
- Realizar el *Aseguramiento de la Calidad del Software* para proporcionar garantías de que los productos de trabajo y procesos cumplen con el *Plan del Proyecto* y la *Especificación de Requisitos*.

Ahora bien, con el fin de ayudar a las microempresas con la implementación del estándar ISO/IEC 29110 y proporcionar una orientación sobre su aplicación efectiva, esta investigación ha propuesto una serie de *Paquetes de Implementación* y *Guías de Implementación* para ampliar la explicación de los procesos definidos en los perfiles del estándar. En este sentido, el conjunto de *Paquetes de Implementación* (PI) (que por cierto es gratuito), reúne una colección de artefactos desarrollados para facilitar la implementación de un conjunto de prácticas del marco de trabajo seleccionado en la microempresa. Un PI no es un modelo de referencia de proceso (es decir, no es prescriptivo). Los elementos típicos de un PI son: descripción de procesos, actividades, tareas, roles y productos, plantillas, listas de comprobación, ejemplos, referencias y mapeos a modelos y estándares, y listas de herramientas. Estos paquetes fueron diseñados de tal manera que una microempresa pueda implementar su contenido, sin tener que implementar, al mismo tiempo, el marco de trabajo en su totalidad. Así, la Figura 2.13 muestra un ejemplo del contenido de un paquete de implementación para la gestión de los proyectos.

- |  |
|--|
| <ol style="list-style-type: none"> <li>1. Descripción técnica <ul style="list-style-type: none"> <li>Importancia de la gestión del proyecto</li> <li>Éxito y fracaso de la gestión del proyecto</li> </ul> </li> <li>2. Definiciones (definiciones genéricas y específicas)</li> <li>3. Relaciones con ISO/IEC 29110 <ul style="list-style-type: none"> <li>Procesos para la gestión de proyectos</li> <li>Tareas y roles</li> </ul> </li> <li>4. Descripción detallada <ul style="list-style-type: none"> <li>Roles, productos y artefactos</li> </ul> </li> <li>5. Plantillas <ul style="list-style-type: none"> <li>WBS, plantilla para el estado del proyecto, etc.</li> </ul> </li> <li>6. Ejemplos <ul style="list-style-type: none"> <li>Prácticas sobre el ciclo de vida de la gestión de proyectos, etc.</li> </ul> </li> <li>7. Listas de comprobación <ul style="list-style-type: none"> <li>Lista de comprobación para la revisión del plan de proyecto, etc.</li> </ul> </li> <li>8. Herramientas</li> <li>9. Referencia a otros estándares y modelos <ul style="list-style-type: none"> <li>ISO 9000, ISO/IEC 12207 y CMMI para Desarrollo</li> </ul> </li> <li>10. Referencias</li> <li>11. Formulario de evaluación del paquete de implementación</li> </ol> |
|--|

**Figura 2.13.** Contenido de un paquete de implementación de la gestión del proyecto (O'Connor & Laporte, 2012).

Adicionalmente, una serie de *Guías de Implementación* ha sido desarrollada para ayudar a implementar un proceso específico a través del soporte de una herramienta computarizada ya existente. Estas guías cubren aspectos como los siguientes:

- Control de Versión con Concurrent Versions System (CVS).
- Control de Versión con Subversion (abreviado frecuentemente como SVN, por el comando svn).
- Gestión de proyectos con GForge.
- Seguimiento de problemas con GForge.
- Mejora del Proceso de Software con OpenOffice Calc.

#### 2.4.6.3. Resultados

El grupo de trabajo que está detrás de la elaboración del estándar ISO/IEC 29110 (denominado Comité ISO/IEC JTC1/SC7 WG 24) se enfoca en la realización de proyectos piloto en microempresas de todo el mundo, como medio para probar las soluciones creadas para acelerar la adopción y utilización del estándar. Así, los proyectos piloto representan un medio importante para reducir los riesgos y aprender más sobre los temas organizativos y técnicos asociados a la implementación de nuevas prácticas de Ingeniería de Software. Un proyecto piloto es también un medio eficaz para demostrar la adopción correcta de nuevas prácticas dentro de un entorno empresarial. Los proyectos piloto que se han realizado para probar las *Guías de Gestión e Ingeniería* y los *Paquetes de Implementación*, entre los que destacan los siguientes:

- En Canadá se realizó un estudio piloto con un departamento de TI de cuatro personas: un analista y tres desarrolladores. Este grupo ha participado en la implementación de tres paquetes de implementación, concretamente para Requisitos de Software, Control de Versiones, y Gestión de Proyectos.
- En Bélgica una microempresa de 25 personas realizó una evaluación con el objetivo de identificar fortalezas y debilidades en sus procesos de desarrollo. Actualmente, esta compañía está trabajando en la implementación de acciones de mejora basándose principalmente en los siguientes paquetes de implementación: Análisis de Requisitos, Control de versiones, y Gestión de Proyectos.
- En Francia se realizó otro estudio piloto con una microempresa de 14 personas que desarrolla y comercializa sistemas de conteo para determinar la frecuencia de asistencia a espacios naturales y sitios públicos. Así, esta empresa logro implementar con éxito el estándar ISO/IEC 29110 a través de las prácticas definidas en los paquetes de implementación disponibles.

De acuerdo con los autores, estos proyectos piloto les han permitido identificar algunas deficiencias de los PI relacionadas con aspectos de infraestructura y procesos de soporte adicionales que están siendo contemplados para la futura evolución de los perfiles de proceso del estándar. Adicionalmente, se están realizando nuevos proyectos piloto en diferentes laboratorios de investigación y empresas de Canadá, Irlanda, Bélgica y Francia. Por ejemplo, en la Escuela Superior de Tecnología (ETS) de Montreal, los estudiantes de los niveles de licenciatura y posgrado han utilizado el proceso de *Gestión de Proyectos* del perfil básico para implementar las actividades en las microempresas existentes. A modo de ejemplo, un estudiante graduado actualmente contratado como ingeniero profesional en una compañía que tiene más de 400 ingenieros, está utilizando el

perfil básico para desarrollar un proceso para la gestión de los proyectos muy pequeños que son desarrollados en dicha empresa. En esta empresa los proyectos son divididos en tres categorías, como se ilustra en la Tabla 19. Un proceso formal para la gestión de los proyectos ya había sido utilizado previamente para proyectos de tamaños mediano y grande, pero solamente se había utilizado un proceso informal para los pequeños. Así pues, la tarea de este caso de estudio es definir un proceso para la gestión de los proyectos pequeños utilizando el perfil básico y sus PI, obtener la aprobación del vicepresidente de la compañía, y ayudar a implementar el nuevo proceso.

**Tabla 19.** Categorías para los proyectos de ingeniería (O'Connor & Laporte, 2012).

Categoría	Proyecto pequeño	Proyecto mediano	Proyecto grande
Duración del proyecto	Menor de 2 meses	Entre 2 y 8 meses	Más de 8 meses
Tamaño del equipo	Igual o menor de 4 personas	Entre 4 y 8 personas	Más de 8 personas
Número de especialidades de la ingeniería	Un especialista	Más de una especialidad	Muchas especialidades
Pagos de ingeniería	Entre \$5,000 y \$70,000 USD	Entre \$50,000 y \$350,000 USD	Cerca de \$350,000 USD

En otro proyecto piloto una microempresa de dos personas está desarrollando un sitio web para que los viajeros compartan con sus amigos sus experiencias desde la planificación inicial hasta el término de un viaje específico. Este sitio web será capaz de construir un perfil personalizado para cada usuario con el fin de proponer temas relevantes, tales como actividades de viaje o lugares de alojamiento. Lo relevante de este caso de estudio es que se está utilizando el conjunto de funciones propuestas por el perfil básico (véase Tabla 20), de tal forma que sea posible que un miembro de la microempresa, equipo B, juegue el rol de jefe de proyectos.

**Tabla 20.** Asignación de roles a dos miembros de la microempresa (O'Connor & Laporte, 2012).

Rol	Proyecto pequeño
Analista	A
Diseñador	B
Programador	A/B
Jefe de Proyectos	B
Líder Técnico	A
Equipo de Trabajo	A/B

El jefe de proyectos utiliza los procesos del perfil básico para la gestión de los proyectos para gestionar el proyecto y producir o revisar los documentos listados en la Tabla 21.

De esta forma, en la medida que la microempresa irá creciendo, el conjunto de roles será asignado entre el personal utilizando los mismos procesos del perfil básico para la gestión de los proyectos y la implementación de software. Si la microempresa decidiera trabajar en más de un proyecto a la vez, será necesario utilizar entonces el proceso del perfil intermedio para la gestión de los proyectos del estándar ISO/IEC 29110.

**Tabla 21.** Asignación de documentos a dos personas de la microempresa (O'Connor & Laporte, 2012).

Nombre del documento	Autor principal	Revisor (en caso de aplicar)
Petición de cambio	A	B
Acciones correctivas	B	A
Mantenimiento de la documentación	B	A
Acta de reunión	A	

Guía de operación del producto	B	B
Reporte de avance	B	
Plan de proyecto	B	A
Repositorio del proyecto	B	
Respaldo del repositorio del proyecto	B	
Especificación de requisitos	A	B
Software	A/B	
Componentes de software	A/B	
Configuración de software	A/B	
Diseño del software	B	A
Documentación de usuario del software	A	B
Enunciado de trabajo	A	B
Casos y procedimientos de pruebas	A	B
Reporte de pruebas	A	
Registro de trazabilidad	B	A
Verificación de resultados	A/B	
Validación de resultados	A/B	

## 2.5. Análisis comparativo de propuestas analizadas

De acuerdo con (Pfleeger, 1995), dentro de la Ingeniería de Software el método comparativo suele ser un método eficiente en una etapa temprana de una investigación, cuando los investigadores intentan pasar del nivel inicial de una revisión exploratoria o búsqueda exhaustiva de información a un nivel más avanzado que le permita establecer estructuras teóricas generales. En este sentido, el análisis comparativo es un método eficaz y simple que permite explicar o utilizar conocimiento tácito para estudiar, por ejemplo, elementos relevantes que pertenecen a un mismo grupo pero que difieren en algunos aspectos. Así, tales diferencias llegan a ser el objeto de la examinación. Con este objetivo en mente, en esta sección se siguen los pasos establecidos por el método comparativo para evaluar las propuestas analizadas en el estado del arte e identificar elementos comunes que permitan ubicar a la solución propuesta en esta tesis en un contexto real. Estos cuatro pasos se desarrollan de la siguiente manera:

1. Definición de los criterios de comparación. Se establecen tres categorías de criterios que corresponden con los aspectos relevantes de la revisión exhaustiva de información (véase Tabla 22). Estas categorías son definidas como:
  - *Definición del enfoque*: Esta categoría reúne a los criterios relacionados con las características y suposiciones teóricas que fundamentan cada propuesta.
  - *Ejecución del enfoque*: Esta categoría identifica a los criterios relacionados con los requerimientos prácticos que se asume deben existir para poder implantar cada propuesta.
  - *Validación del enfoque*: Esta categoría establece los criterios que valoran el entorno de validación de cada propuesta.
2. Establecimiento de una escala de valoración. Dado que es necesario definir una escala que permita calificar los criterios establecidos anteriormente, este análisis comparativo establece una escala de tipo cualitativo que le permita determinar, en base a la información recogida, qué aspectos son cubiertos por cada categoría de criterios. A diferencia de una escala cuantitativa, basada en la inducción probabilística, la escala cualitativa es de carácter

exploratorio, inductivo y descriptivo que se orienta al proceso (a la forma de hacer las cosas) no al resultado.

**Tabla 22.** Definición de criterios para realizar el análisis comparativo.

<b>Criterios</b>	<b>Descripción</b>
<b><i>Categoría: Definición del enfoque</i></b>	
I. Procesos de la gestión de proyectos que son cubiertos	Identifica cuáles de los procesos de la gestión de proyectos son abordados por cada propuesta revisada. En este sentido, es importante determinar el alcance de las propuestas actuales para justificar el alcance de la solución propuesta en esta tesis.
II. Modelos, métodos y/o estándares involucrados	Determina el conjunto de mejores prácticas que se relacionan con la implantación de un proceso que busca mejorar significativamente la realización de la gestión de los proyectos.
III. Enfoque de realización	Establece si la propuesta es creada para un contexto tradicional, ágil, o la combinación de ambos. A pesar de que parezca imposible, es importante determinar si estos dos enfoques (ágil y tradicional) están siendo utilizados al mismo tiempo para mejorar la gestión de los proyectos.
IV. Habilidades técnicas requeridas	Dado que es importante considerar que la solución propuesta en esta tesis debe ajustarse a las características de los pequeños equipos formados en las pequeñas empresas, este criterio identifica los conocimientos mínimos con los que el personal debe contar para realizar una implantación exitosa.
<b><i>Categoría: Ejecución del enfoque</i></b>	
V. Objetivo de la implantación	Identifica las metas a alcanzar con la implantación de cada propuesta. Es claro que dicho objetivo está alineado con la mejora interna de la empresa y por ende busca proveer de herramientas a la misma para mejorar la productividad de su personal.
VI. Bases teórica para la implantación	Determina, considerando el objetivo de la implantación, cuál es el fundamento teórico que soporta a la propuesta. Esto permitirá determinar la característica de repetitividad en caso de que se trate de modelos o guías de implantación.
VII. Tipo de herramienta de soporte a la implantación	Evidencia el tipo de herramienta que soporta la base teórica definida. Esta herramienta puede estar en forma de documentos o plantillas, herramientas computarizadas, páginas web, o cualquier otro medio que facilite la adopción de las mejoras a la gestión de proyectos.
<b><i>Categoría: Validación del enfoque</i></b>	
VIII. Tamaño de empresas/proyectos	Reconoce la importancia de analizar si la investigación actual le está prestando atención a la pequeña empresa y a los pequeños equipos que son integrados en ésta para desarrollar los proyectos de software.
IX. Entorno de aplicación	Establece el contexto de validación donde se ponen a prueba las ideas y conceptos generados con cada propuesta. En este sentido, es importante determinar si se trata de un esfuerzo académico o industrial.

3. Desarrollo de un cuadro comparativo con la información recogida sobre el mapeo criterio/factor de valoración. En este sentido, la Tabla 23 resume los hallazgos del análisis

comparativo y proporciona una visión clara del contexto, definición, ejecución y validación de las propuestas revisadas en el estado del arte.

**Tabla 23.** Análisis comparativo de propuestas enfocadas a la gestión de proyectos.

Criterios	Propuestas revisadas					
	(O'Connor & Laporte, 2012)	(O'Sheedy & Sankaran, 2013)	(Mas & Mesquida, 2013)	(Tasevska, Damij, & Damij, 2014)	(García, Pacheco, & Calvo-Manzano, 2014)	(Mesquida & Mas, 2014)
I.	Planificación del Proyecto, Evaluación y Control del Proyecto	Planificación del Proyecto, Ejecución y Control del Proyecto	Planificación del Proyecto, Monitorización y Control del Proyecto	Planificación del Proyecto	Planificación del Proyecto, Monitorización y Control del Proyecto, Definición de Procesos de la Organización, Medición y Análisis, Gestión Cuantitativa del Proyecto	Planificación del Proyecto, Evaluación y Control del Proyecto
II.	ISO/IEC 29110	Scrum, PMBoK	ISO/IEC 12207, ISO/IEC 29110, PMBoK	ERP	CMMI-DEV v1.3	ISO/IEC 29110
III.	Tradicional	Combinación de tradicional y ágil	Tradicional	Tradicional	Tradicional	Tradicional
IV.	Experiencia práctica en la gestión, técnicas de gestión, conocimiento sobre estándar	Experiencia práctica en la gestión, técnicas de gestión, conocimiento sobre Scrum	Experiencia práctica en la gestión, técnicas de gestión, conocimiento sobre estándares y PMBoK	Experiencia práctica en la planificación, técnicas de gestión, conocimiento sobre ERPs	Experiencia práctica en la gestión, técnicas de gestión, conocimiento sobre CMMI-DEV v1.3	Experiencia práctica en la gestión, técnicas de gestión, conocimiento de estándar
V.	Simplificar la complejidad de la gestión de proyectos	Agilizar la ejecución de la gestión de proyectos	Simplificar la complejidad de la gestión de proyectos	Probar prácticas exitosas en el desarrollo de ERPs en la planificación	Definir procesos para mejorar la gestión de los proyectos	Simplificar la complejidad de la gestión de proyectos
VI.	Guías de Gestión e Ingeniería	Marco de Trabajo	Guías de mapeo	Guías de mapeo	Marco de Trabajo	Guías de adopción y herramientas
VII.	Repositorio de Paquetes de Implementación	Documentación	Documentación	Documentación	Software (PALSS, PROMEP, SysProVal)	Software (PAL)
VIII.	Micro/pequeños	PyMES/pequeños	Micro, PyMES/pequeños	PyMES/pequeños	PyMES/pequeños	Pequeñas/pequeños
IX.	Académico/industrial	Industrial	Académico	Académico/industrial	Industrial	Industrial

4. Redactar las conclusiones sobre el análisis realizado. A través del análisis comparativo se alcanzaron las siguientes conclusiones:
  - La mayoría de las propuestas analizadas aborda dos procesos de la gestión de los proyectos de software: la Planificación del Proyecto y la Monitorización y Control del Proyecto. Cabe recordar que se mencionó anteriormente que ambos procesos representaban la base de cualquier esfuerzo inicial de mejora dentro de una empresa, puesto que, independiente del modelo de procesos, estándar o metodología que se pretenda implantar, estos procesos se encuentran en los niveles iniciales que definen a las actividades de mayor importancia para que una empresa pueda realizar y gestionar correctamente su trabajo. En este sentido, no es extraño que las investigaciones actuales sobre la gestión aborden estos procesos medulares para el desarrollo exitoso de los proyectos. Así, los estudios revisados coinciden en que aquellas PyMES que desean mejorar la gestión de sus proyectos, o que poseen poca experiencia en esta área, deben comenzar por estos dos procesos básicos.
  - Otra observación importante se relaciona con el uso de modelos, metodologías y estándares como medio de soporte en la implantación de la gestión de los proyectos. En este contexto, la mayoría de las propuestas se enfocan a la combinación de un estándar creado específicamente para las empresas pequeñas (ISO/IEC 29110) y PMBoK. Dicha combinación representa un mecanismo inteligente de simplificación, puesto que el estándar establece qué hacer en la empresa para gestionar un proyecto y PMBoK proporciona el cómo. De esta manera las PyMES que carecen de conocimiento sobre las técnicas y/o herramientas de gestión reciben recomendaciones más precisas para implantar la gestión de forma eficiente. De manera similar, se observa que la combinación de una metodología ágil como XP o Scrum con PMBoK intenta simplificar la complejidad de la gestión y por ende mejorar la eficiencia de la empresa. Por último, es importante hacer notar que no se encontró alguna propuesta orientada a mejorar los procesos de gestión relacionada con el modelo MoProSoft, creados específicamente para las PyMES mexicanas.
  - En este sentido, la mayoría de las propuestas analizadas se enfocan en la definición de procesos con un enfoque tradicional, y solamente una aborda la combinación de los enfoques ágil y tradicional. Se considera que esto tiene sentido, puesto que si se desea que las PyMES den sus primeros pasos en la gestión de sus proyectos, éstas deben comenzar con un enfoque tradicional dado que la capacitación en el uso de las técnicas tradicionales de gestión proporcionará el conocimiento requerido para aplicar posteriormente cualquier técnica ágil.
  - Abundando un poco más sobre la necesidad de capacitación (una de las principales deficiencias de las PyMES), se observa que todas las propuestas analizadas requieren cierto grado de experiencia a nivel técnico y de gestión. En este sentido, se considera que un profesionalista con un mínimo de tres años de experiencia posee el conocimiento técnico necesario para aplicar una técnica tradicional de gestión, mientras que un jefe de proyectos con cinco años de experiencia tiene la capacidad de realizar y controlar las actividades que implica la gestión completa de un proyecto y, por ende, es capaz de tomar decisiones efectivas sobre el proceso de desarrollo.
  - Se observa también que el uso de herramientas computacionales brinda el soporte necesario para que las PyMES realicen de forma exitosa la implantación de un proceso de gestión de proyectos. Dado que se pretende ayudar a estas empresas con sus

problemas cotidianos, estas herramientas son creadas con una arquitectura bastante sencilla y eficaz que cubre prácticamente la necesidad de gestión de conocimiento en el entorno de desarrollo. A pesar de esto, la mayoría de las propuestas analizadas se enfoca a establecer un soporte documental a través de la definición de guías y recomendaciones teóricas que buscan simplificar el uso del modelo/estándar, algo que complica sin duda alguna la labor de las PyMES puesto que todo el proceso de mejora se vislumbra como altamente burocrático.

- Por último, se observa un marcado énfasis por investigar y resolver los problemas que las PyMES tienen al realizar la gestión de sus proyectos pequeños. Era de esperarse cuando se ha mencionado que este tipo de empresas rebasa el 90% de todas las organizaciones de software en países como Estados Unidos de Norteamérica, India, Canadá, México, Irlanda y demás.

## 2.6. Conclusiones finales del capítulo

La gestión de los proyectos contribuye a incrementar las probabilidades de éxito de los proyectos desarrollados no solamente en el contexto de la industria de software, sino en el ámbito general (Patanakul et al., 2010). De acuerdo con el CMMI-DEV v1.3 (CMMI, 2010), la gestión de los proyectos puede ser implantada en la industria del software a través de dos enfoques: el inicial que corresponde a la definición de áreas básicas de proceso orientadas al establecimiento y mantenimiento de un plan de proyecto y de compromisos (Planificación del Proyecto), la monitorización del progreso del proyecto frente a dicho plan y la toma de acciones correctivas (Monitorización y Control del Proyecto), y la gestión de acuerdos con los proveedores; y el enfoque especializado que corresponde a la definición de áreas avanzadas de proceso que involucran el establecimiento de un proceso definido que se adapte a partir de un conjunto de procesos estándar de la organización, la gestión de los riesgos, la creación y gestión de equipos integrados para la conducción de los proyectos, y la gestión cuantitativa del proceso definido en el proyecto. En este sentido, es lógico pensar que para implantar de forma exitosa la gestión de los proyectos dentro de las PyMES que cuentan con poca o nula experiencia en este ámbito, es necesario desarrollar propuestas que estén orientadas a dos de los procesos de las áreas básicas de gestión: la *Planificación del Proyecto* y la *Monitorización y Control del Proyecto*.

Por otro lado, la revisión de literatura expuesta a lo largo de este capítulo ha permitido confirmar que la gestión de los proyectos en el ámbito de las PyMES debe contemplar factores como el contexto y las características de este tipo de empresas, el entorno dinámico en el que se desarrollan, el alto grado de adaptabilidad al que están expuestos sus productos de software, y la tecnología altamente cambiante. Estos hallazgos coinciden con los resultados publicados por Tuner et al., (2010) y proporcionan la base para que las investigaciones desarrollen técnicas, marcos de trabajo, y/o herramientas que contribuyan a subsanar la problemática de estas empresas.

En este sentido, a través del estudio realizado en este capítulo se puede concluir también que tales investigaciones están poniendo un énfasis especial en simplificar la forma en que se realiza la gestión de los proyectos en el entorno de las PyMES mediante la definición de guías de implantación o ejecución, marcos de trabajo, bibliotecas de activos y de conocimientos. Sin embargo también es posible identificar que no existe una base teórica enfocada exclusivamente a estos dos procesos básicos (*Planificación del Proyecto* y la *Monitorización y Control del Proyecto*) que se oriente a que los pequeños equipos de trabajo puedan definir e implantar estos procesos de manera práctica y sencilla. Aunado a esto, y dadas la características de estas empresas, no se observa que las propuestas analizadas agreguen un alto grado de repetitividad de tal forma que los pequeños equipos puedan gestionar sus proyectos reutilizando el conocimiento adquirido en sus proyectos

previos. Esta repetitividad se podría alcanzar al establecer una estructura de trabajo que clasifique la información y la haga útil a través de una herramienta computarizada. Así pues, los hallazgos de este capítulo coincide con los comentarios vertidos por la investigación de (Robillard, Walker, & Zimmermann, 2010) en relación al uso de herramientas computarizadas en las actividades de la Ingeniería de Software. Es decir, el uso de una herramienta computarizada en la gestión de los proyectos de las PyMES podría asistir al personal a realizar las actividades que usualmente omite por no entenderlas o “agilizar” su trabajo.

Así pues, estos hallazgos soportan la idea inicial sobre que un metamodelo podría definir de manera simple los elementos relacionados con la gestión de los proyectos, establecer a partir de éstos diferentes niveles de abstracción para su correcta comunicación, y modelar las relaciones que permitan la interacción entre los elementos especificados para definir de forma sistemática un proceso repetible de gestión. Estos argumentos coinciden con la investigación de Henderson-Sellers (2011), quien establece que un metamodelo define las relaciones que existen entre los diferentes elementos de un proceso a través de relaciones de homomorfismo<sup>19</sup>. De esta manera, los metamodelos suelen ser llamados “modelo de modelos”, pero no porque contengan información de muchos modelos sino porque incluyen el concepto de patrón, de tal forma que los modelos definidos en él pueden ser utilizados de muchas formas en diferentes contextos (proyectos). Es decir, a través de esta idea la PyMES tendría la oportunidad de definir su propio proceso de gestión haciendo uso del metamodelo construido y obtener así su propio modelo a través de un modelo ya existente, un modelo que permite crear diferentes modelos.

Por último, teniendo este objetivo en mente, el siguiente capítulo presenta el diseño del metamodelo introducido en estas conclusiones con el objetivo de definir una estructura de trabajo que agrega una alta repetitividad en la definición de un proceso (de acuerdo a las características de cada proyecto) en el contexto de la PyMES.

---

<sup>19</sup> En matemáticas, un homomorfismo (o a veces simplemente morfismo) desde un objeto matemático a otro con la misma estructura algebraica, es una función que preserva las operaciones definidas en dichos objetos. Es decir, este concepto aplicado al diseño de metamodelos dentro de la Ingeniería de Software implica el diseño de un modelo (regularmente con UML) que define elementos y relaciones particulares de un proceso, el cual puede ser aplicado a diferentes proyectos sin alterar su definición original (Gonzalez-Perez & Henderson-Sellers, 2008).



## **3. Metamodelo de gestión para los pequeños equipos**

### **3.1. Los metamodelos en la Ingeniería de Software**

De acuerdo con (García, Piattini, Ruiz, Canfora, & Visaggio, 2006) uno de los principales objetivos de las empresas de software es tener éxito en la gestión de sus procesos, con el fin de mejorar la calidad de sus productos. Así pues, el satisfacer los requisitos de calidad implica que los procesos deben ser definidos de manera correcta para producir los resultados esperados, y que cualquier mejora introducida en la empresa debe estar estrechamente relacionada con sus objetivos de negocio. Estos aspectos pueden hacer la diferencia cuando las empresas de software intentan ser más competitivas. En este contexto, la gestión de los procesos implica cuatro responsabilidades: definir, medir, controlar y mejorar. Por ejemplo, en relación con la definición (aspecto relevante en el contexto de esta tesis) es necesario considerar las siguientes actividades:

- El modelado de los procesos. Dada la especial complejidad de los procesos de software, que se deriva de la gran diversidad de elementos que deben considerarse para su gestión, es necesario definirlos eficazmente. Así, el modelado de procesos constituye el punto de partida para analizarlos, verificarlos y mejorarlos. Sin embargo, ante la notoria diversidad de propuestas de modelos de procesos (e.g., CMMI-DEV v1.3, MoProSoft, PMBoK), es necesario contar con un metamodelo que permita definir un modelo para que pueda ser utilizado en la definición de otros más. Por ejemplo, con este objetivo OMG propuso su metamodelo SPEM y un perfil de UML (Lenguaje Unificado de Modelado) que definen un idioma de referencia que ha servido para modelar diversos procesos desde su creación.
- La evaluación del proceso. Con el fin de promover la mejora de los procesos de software, es muy importante establecer previamente un marco para su análisis con el objetivo de determinar sus puntos fuertes y débiles. Este objetivo requiere la definición de medidas relacionadas con los diferentes elementos involucrados en un proceso de software. Sin embargo, dada la diversidad de entidades involucradas en la evaluación de los procesos de software, se hace necesario el establecimiento de una terminología común para la definición, el cálculo y la explotación de las medidas requeridas para la gestión del proceso.

En este sentido, el modelado y la evaluación de los procesos de software pueden beneficiar a las empresas de la siguiente manera:

- Constituyen un factor crítico para alcanzar un alto grado de madurez en sus procesos, de acuerdo con los modelos existentes (e.g., CMMI-DEV v1.3, MoProSoft, ITIL);
- Representan un medio para reducir la incompreensión de los modelos de madurez mediante el uso de una misma notación de modelado y criterios uniformes de evaluación;

- La integración de ambas actividades, modelado y evaluación, en una herramienta computacional de software reduce los costos, en términos de licencias y tiempo para la capacitación de los participantes.

Así pues, durante estos últimos años se ha requerido ser más estricto con respecto al uso eficiente de los procesos dentro de la Ingeniería de Software, lo cual ha generado como consecuencia un incremento en el interés de la creación de fundamentos rigurosos para diversos aspectos de la Ingeniería de Software, llevando a la exploración de diversas soluciones. En este sentido, los metamodelos surgen para ofrecer un medio estricto enfocado al modelado de procesos/metodologías, y su implantación en un contexto real a través de la adaptación/definición de procesos más simples. En consecuencia, el uso de metamodelos implica el modelar una metodología de trabajo como si se tratase de cualquier otro sistema, a través de la aplicación de las mismas ideas y procedimientos que por lo general se utilizan en el modelado de aplicaciones de negocio u otros sistemas de software intensivo (Henderson-Sellers, 2011). Así, es lógico intuir que los metamodelos están estrechamente ligados con los modelos, por lo que primero es necesario establecer la definición de modelo y posteriormente la de metamodelo:

- Por ejemplo, en la guía de la Arquitectura Dirigida por Modelos (MDA, por sus siglas en inglés) de la OMG (OMG, 2003) se argumentó que un modelo era *“una descripción o especificación de un sistema y su entorno que pretende cubrir algún propósito específico”*.
- Por otro lado, Guizzardi (2005) definió formalmente que un modelo era *“una abstracción de la realidad de acuerdo a cierta conceptualización”*.
- Por último, Henderson-Sellers (2011) afirma que un modelo es *“la representación de cierta realidad”*.

Considerando las definiciones anteriores, Kühne (2006) definió a un metamodelo como *“una abstracción (real o basada en algún lenguaje) de un sistema que permite predecir o inferir qué se debe hacer para alcanzar un propósito”*. De manera similar, Gonzalez-Perez, McBride, y Henderson-Sellers (2005) definieron un metamodelo como *“el modelo de una metodoloxía, o familia de metodologías, que proporciona un mayor grado de formalización y un mejor soporte para obtener una abstracción consistente y personalizada, puesto que los conceptos y fundamentos que lo soportan están explícitamente definidos”*. En este sentido, Henderson-Sellers (2011), también considera que un metamodelo es un “modelo de modelos” y afirma que a través de un metamodelo que represente una metodología es posible generar diferentes instancias (o modelos más pequeños) que pueden ser utilizados para lograr un propósito concreto. En este contexto, es claro que debe existir una relación de uno a varios entre los elementos del metamodelo, ésta describe una relación de homomorfismo (González-Pérez & Henderson-Sellers, 2008). Así, la instanciación del modelo se ajusta a su metamodelo cuando cada elemento en él es asignado a un elemento correspondiente en el metamodelo.

Así, los metamodelos son construidos usualmente como un conjunto de (meta) clases y (meta) relaciones que permiten crear instancias a través de elementos metodológicos (e.g., este es el principio seguido por SPEM y UML). De esta forma, las clases representan los conceptos que la metodología debe abordar, como: procesos, tareas, actividades, técnicas, etc. Sin embargo, diversos estándares y autores utilizan términos diferentes para un mismo concepto y diferentes significados para los mismos términos, es por eso que se recomienda primero definir la terminología que el metamodelo utilizará para facilitar su uso (Gonzalez-Perez & Henderson-Sellers, 2007).

Por otro lado, los metamodelos se encuentran estrechamente relacionados con las ontologías, ya que ambos son utilizados a menudo para describir y analizar las relaciones entre los conceptos de la Ingeniería de Software (Söderström, Andersson, Johannesson, Perjons, & Wangler, 2002). Las ontologías expresan un significado dentro de un universo o dominio del discurso mediante la utilización de una gramática que define el uso de un vocabulario. Así, dicha gramática especifica lo que implica una declaración bien formada dentro de dicho universo a través del vocabulario definido por la ontología. Un metamodelo, por otro lado, es un modelo explícito de constructores y reglas necesarias para construir modelos específicos dentro de un dominio de interés. Un metamodelo válido es una ontología, pero no todas las ontologías se modelan explícitamente como metamodelos. En este sentido, un metamodelo puede verse desde tres perspectivas diferentes:

- Como un conjunto de bloques de construcción y reglas que se utilizan para generar modelos,
- Como el modelo de un dominio de interés, y
- Como una instancia de otro modelo.

Así pues, esta tesis pretende desarrollar un metamodelo que establece un mecanismo para definir procesos para la gestión de proyectos (dominio específico), bajo los conceptos definidos por Kühne (2006) y Henderson-Sellers (2011). Es decir, se pretende crear un “modelo de modelos” que permita definir el marco conceptual de los procesos contenidos en un modelo de referencia, de tal manera que sea posible obtener instancias de éste para que sean aplicables a cualquier proyecto independientemente de sus características.

Una vez que se ha definido el concepto de metamodelo, las siguientes secciones de este capítulo muestran la solución propuesta en esta tesis considerando la metodología definida en el Capítulo 1 (véase Figura 3.1)

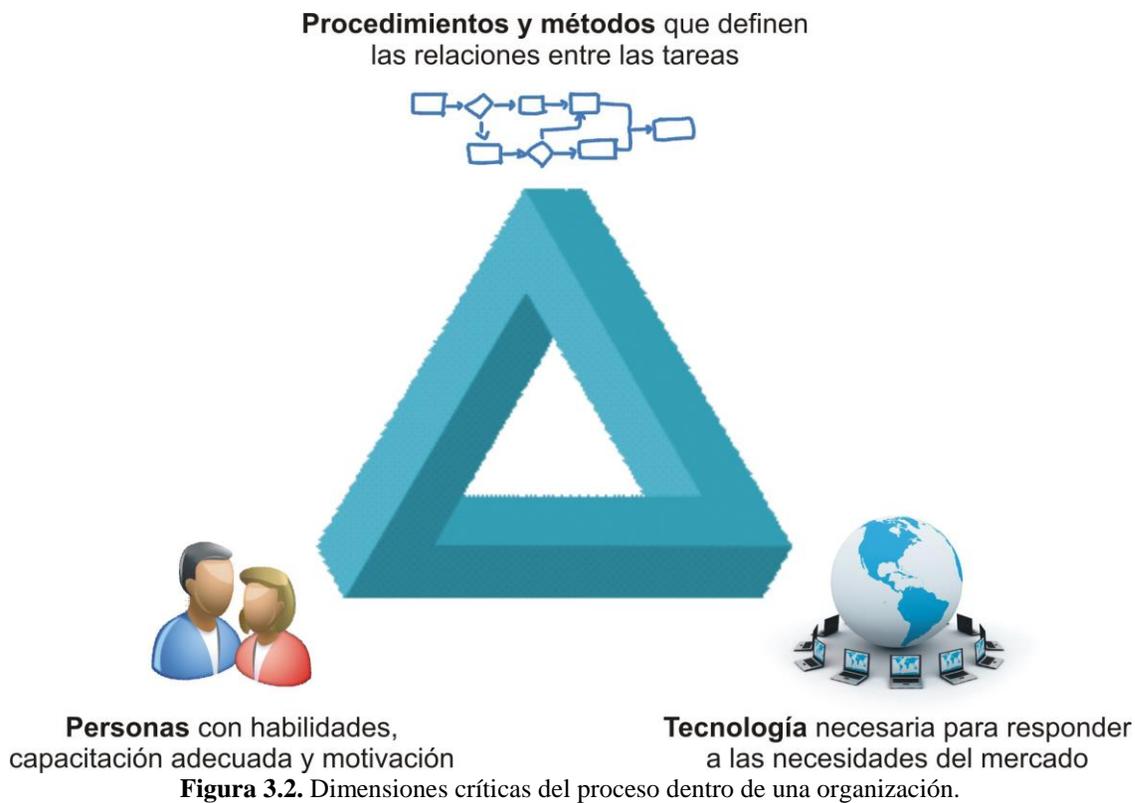


**Figura 3.1.** Actividades para desarrollar el metamodelo propuesto.

### 3.2. Determinar el ámbito del metamodelo

De acuerdo con Pressman y Maxim (2014), el proceso es el fundamento de la Ingeniería de Software. Es decir, a través de éste es posible definir un marco de trabajo para un conjunto de tareas clave, las cuales forman en el caso particular de esta tesis, la base del control sobre la gestión de los proyectos de software y establecen el contexto en el cual se aplican los métodos técnicos, se producen los resultados del trabajo, se establecen hitos, se asegura la calidad, y se gestiona adecuadamente el cambio. En este sentido, el Glosario Estándar de Terminología para la Ingeniería de Software de la IEEE definió a un proceso como “una secuencia de pasos realizados para alcanzar un fin determinado” (Radatz, Geraci, & Katki, 1990).

Así pues, para que una empresa humana como: por ejemplo, una “compañía de software” produzca una salida, es necesario relacionar tres elementos básicos: personas, procesos, y tecnología; como se ve en la figura (véase Figura 3.2). La geometría indica que los lados y ángulos del triángulo están interrelacionados y que no se puede modificar alguno de estos sin cambiar los otros de manera predecible. Algo similar ocurre en la industria de software, ya que de acuerdo a la evolución del proceso, las personas y la tecnología cambiarán de manera predecible. Es decir, los procesos proporcionan a las empresas una forma de incorporar el conocimiento sobre cómo hacer mejor las cosas, permitiéndoles así explotar sus recursos al máximo y analizar las tendencias de su actividad. En este sentido, el proceso ayuda a los miembros de una empresa a alcanzar sus objetivos de negocio, puesto que se trabaja de manera inteligente, no con mayor esfuerzo pero si de un modo consistente (CMMI, 2010).

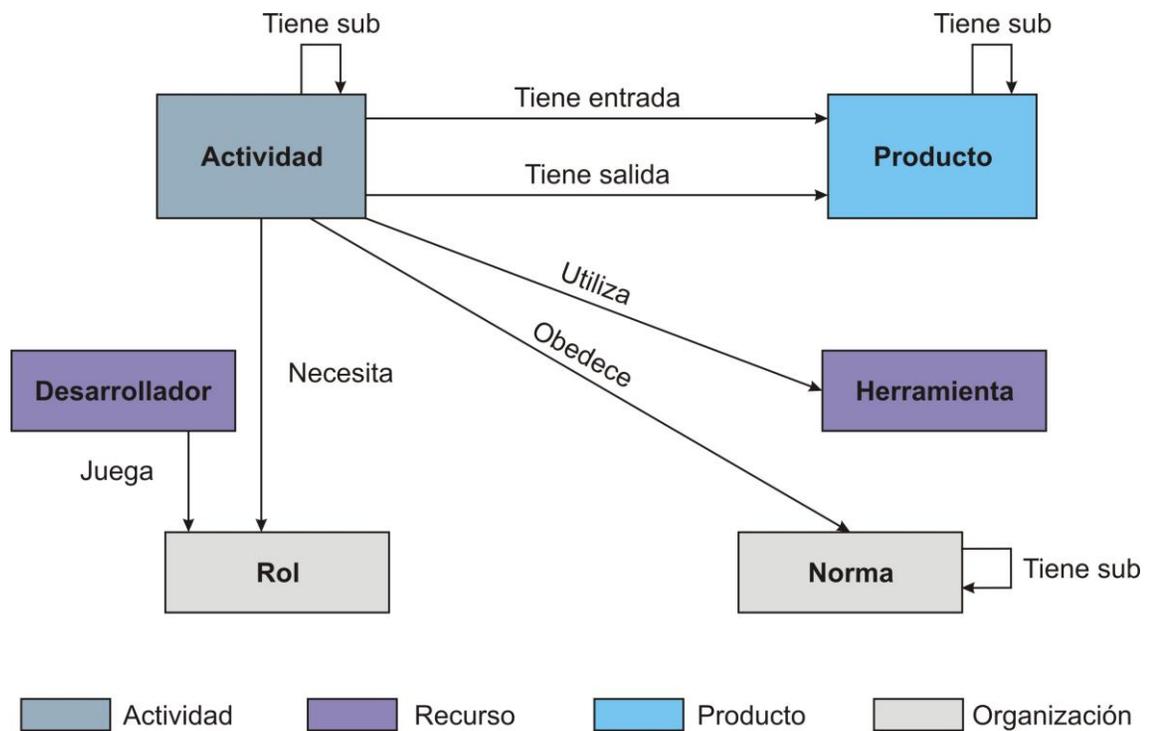


Así, el término “proceso” se refiere a un contexto que va más allá del empleado para el desarrollo de software (i.e., análisis, diseño, codificación, pruebas, liberación), es decir, se refiere a cualquier forma de trabajo utilizada por un proyecto de software o empresa, y a las actividades

identificadas para llevarla a cabo y, de esta manera, generar un producto o servicio de software. Es importante recalcar que un proceso requiere de la continua revisión, medición y control.

En este contexto, Fuggetta (2000) argumentó hace quince años que un proceso se fundamentaba en un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos para concebir, desarrollar, implementar y mantener un producto de software. Esta declaración ha sido complementada a lo largo del tiempo con el objetivo de obtener una visión clara de la importancia al definir un proceso dentro de la industria de software. Por ejemplo, los colaboradores del SEI han argumentado que un proceso reúne un conjunto de actividades, métodos y prácticas, y transformaciones que se utilizan en la producción y evolución de software (CMMI, 2010). De manera similar, Pino, Garcia, y Piattini (2007) retoman la idea de Fuggetta al establecer que la utilidad de un proceso dentro de una empresa de software se enfoca en la definición de la organización lógica de las personas, material, equipos y procedimientos, y su relación con las actividades de trabajo diseñadas para generar un producto. Es importante mencionar que para alcanzar los resultados esperados, dicho proceso debe ser definido, organizado, gestionado, y medido. Así pues, la definición del proceso es la principal característica que se pretende alcanzar con la incorporación de un metamodelo.

En este sentido, la Figura 3.3 muestran los tipos de elementos que son necesarios para definir y modelar un proceso (Pino, Ruiz, Garcia, & Piattini, 2012). La figura muestra que un proceso contiene actividades como elemento principal de su ejecución. La realización de estas tareas conduce a la generación de productos a través de entradas y salidas que son previamente definidas. Así, las actividades son realizadas por personas con diferentes roles que regularmente utilizan herramientas documentales o computarizadas que facilitan su labor. Es muy importante mencionar que las actividades de un proceso regularmente corresponden a las mejores prácticas resumidas por una norma o modelo de procesos (o de referencia) y que deben ser medidas y controladas para asegurar su correcta ejecución.



**Figura 3.3.** Tipos de elementos para modelar/representar un proceso (Pino, Ruiz, Garcia, & Piattini, 2012).

Considerando lo anterior puede argumentarse que los elementos mostrados en la Figura 3.3 representan la unidad fundamental del proceso, es decir, éste se puede definir en términos de elementos de proceso que están simbolizados de forma general por un conjunto de actividades estrechamente relacionadas. De acuerdo con el (CMMI, 2010), estos elementos pueden ser descritos a través del uso de plantillas que deban completarse, abstracciones que deban refinarse o descripciones que deban utilizarse o modificarse. En este mismo contexto, Varkoi (2010) ha establecido en base al estándar ISO/IEC 29110 (ISO/IEC, 2011) una definición simplificada de un proceso enfocado a la gestión de proyectos dentro de las pequeñas empresas de software (véase Figura 3.4). El perfil definido por Varkoi simplifica la definición del proceso considerando que a partir de un objetivo se pueden establecer objetivos que sean alcanzados a través de prácticas base, actividades y productos de trabajo.



**Figura 3.4.** Elementos básicos de un proceso para la gestión de proyectos en pequeñas empresas (Varkoi, 2010).

Sin embargo, la mayoría de las personas cree que un proceso suele manejar mucha información, pero la importancia de su definición reside en la necesidad de enfatizar que dicha documentación no siempre se utiliza de la misma manera. En este sentido, la Tabla 24 resume los hallazgos de Olson (2006) al proponer que una definición corta y usable de un proceso puede lograrse con el establecimiento de documentos básicos y la identificación de sus formas de uso.

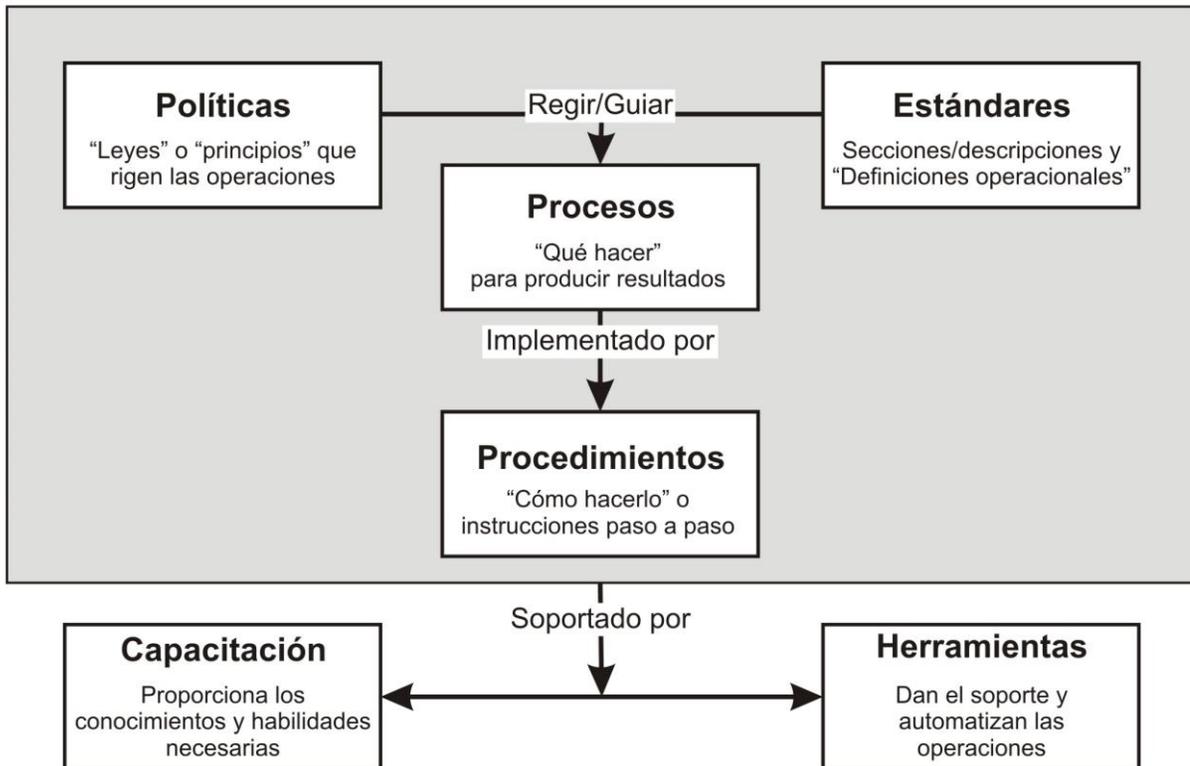
**Tabla 24.** Tipo de documentación de un proceso (Olson, 20006).

Tipo de documento	Uso principal
Política	<ul style="list-style-type: none"> <li>- Utilizado por la alta dirección para establecer el rumbo de una empresa.</li> <li>- Establece los principios que las empresas deben seguir.</li> </ul>
Estándar	<ul style="list-style-type: none"> <li>- Especifica las secciones que debe contener un documento y proporciona una descripción de lo que sucede en esas secciones.</li> <li>- Incorporan la característica de repetitividad.</li> </ul>
Proceso	<ul style="list-style-type: none"> <li>- Qué pasa con el tiempo para producir el resultado</li> </ul>

	deseado. - Debe responder a cinco preguntas: quién, qué, dónde, cuándo y por qué.
Procedimiento	- La información de cómo hacerlo o paso a paso. Incluyen ejemplos de procedimientos como listas de comprobación ( <i>checklists</i> ), formularios, y/o tablas de paso/acción. - Implementa parte de un proceso.

De manera similar, Olson establece las relaciones existentes entre los tipos de documentos de un proceso (véase Figura 3.5). En este sentido, es importante recalcar que dentro de una organización es correcto que los procesos y procedimientos tengan diferentes usuarios: principiantes (aquellos que nunca han utilizado el proceso), intermedios (aquellos que han utilizado el proceso solamente un par de ocasiones y por consiguiente necesitan orientación y lecciones aprendidas), y expertos (aquellos que han utilizado el proceso muchas veces e incluso pueden ser responsables de su ejecución). Así, la información del proceso puede ser clasificada de acuerdo a los niveles del usuario simplificando así su uso.

**Marco de trabajo para la documentación**



**Figura 3.5.** Tipos de documentación de procesos y sus relaciones (Olson, 2006).

En este sentido, con la intención de obtener una descripción general de los elementos de un proceso a continuación se definen algunos conceptos utilizados por diversos modelos de proceso y que encajan con el marco de trabajo propuesto por Olson:

- **Proceso:** es un conjunto sistemático de actividades de trabajo relacionadas entre sí, y que tienen una o más entradas que están dirigidas a producir una o más salidas hasta alcanzar un resultado final (CMMI, 2010).

- Rol: describe de forma abstracta el conjunto de habilidades y/o responsabilidades asociadas con la ejecución de una o más actividades. De acuerdo con el PMBoK, un rol es una función definida para ser realizada por un miembro del equipo del proyecto (e.g., probar, archivar, inspeccionar, codificar) (PMI, 2013).
- Actividades: representan una porción definida y planificada de trabajo ejecutado durante el curso de un proyecto (PMI, 2013).
- Entrada: es cualquier elemento interno o externo del proyecto que es requerido por un proceso para seguir avanzando (CMMI, 2010).
- Salida: es un producto, resultado o servicio generado por un proceso. Puede representar un dato inicial de otro proceso sucesor (NYCE, 2005).
- Productos de trabajo: es un resultado útil de un proceso. Dicho resultado puede incluir ficheros, documentos, productos, partes de un producto, servicios, descripciones de un proceso, y especificaciones (CMMI, 2010). Por otro lado, se dice que éstos representan al objeto de trabajo de un entorno, y corresponden a objetos de desarrollo de software típicos, tal como los documentos de requisitos, planes de pruebas, pruebas de casos, etc.
- Guía: es una recomendación que resume políticas, estándares o procedimientos acerca de la manera en que debería de realizarse una porción del trabajo (PMI, 2013).
- Metodología: un sistema de prácticas, técnicas, procedimientos y normas utilizado por quienes trabajan dentro de una disciplina (PMI, 2013).
- Practica: un tipo específico de actividad profesional o gestión que contribuye a la ejecución de un proceso y que puede utilizar una o más técnicas y herramientas (NYCE, 2005).
- Recurso: medio de cualquier clase que, en caso de necesidad, sirve para conseguir lo que se pretende dentro de un proyecto. Por ejemplo, recursos humanos especializados (en disciplinas específicas, ya sea habilidades individuales o grupales), equipos, servicios, suministros, materias primas, materiales, presupuestos o fondos económicos (CMMI, 2010).
- Herramientas: medios que automatizan la ejecución de ciertas actividades (CMMI, 2010). Representan algo tangible, como una plantilla o un software especializado, que es utilizado para apoyar el desarrollo de una actividad con el objetivo de facilitar la obtención de un producto o resultado (PMI, 2013).
- Políticas: un patrón estructurado de acciones adoptado por una organización que explica, a través de un conjunto de principios básicos, los lineamientos que rigen su conducta (NYCE, 2005).
- Proyecto: un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único (PMI, 2013).
- Medida: es aquella que proporciona una indicación cuantitativa de extensión, cantidad, dimensión, capacidad, y/o tamaño de algunos atributos de un proceso o producto de software (Pressman & Maxim, 2014).
- Activo: son planes, procesos, políticas, procedimientos y bases de conocimiento que son específicos de la organización ejecutante y que son utilizados por la misma (PMI, 2013).

De esta manera, la definición de un proceso puede verse como la acción de encapsular la experiencia obtenida por la empresa de software en forma de una receta exitosa.

Los elementos de un proceso son identificados en la Tabla 25 para cada uno de los siguientes modelos MoProSoft v 1.3, CMMI-DEV v 1.3, el estándar ISO/IEC 29110 y la guía del PMBoK.

**Tabla 25.** Mapeo de elementos de proceso entre los modelos/estándares que promueven la gestión.

Elementos de proceso	Modelos			
	MoProSoft v 1.3	CMMI-Dev v1.3	ISO/IEC 29110	PMBoK
Proceso	APE (Administración de Proyectos Específicos).	Gestión de Proyectos (procesos básicos y avanzados).	Gestión de Proyectos.	Gestión de Proyectos.
Roles	<ul style="list-style-type: none"> <li>• Responsable de la Gestión de Proyectos (RGPY).</li> <li>• Responsable de la Administración del Proyecto Especifico (RAPE).               <ul style="list-style-type: none"> <li>• Cliente (CL).</li> </ul> </li> <li>• Responsable del Subcontrato (RSC).</li> <li>• Responsable de Desarrollo y Mantenimiento de Software (RDM).</li> <li>• Equipo de Trabajo (ET).</li> </ul>	<ul style="list-style-type: none"> <li>• Alta dirección.</li> <li>• Jefe del proyecto.</li> <li>• Equipo del proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>• Cliente.</li> <li>• Gestor del Proyecto.</li> <li>• Líder técnico.</li> <li>• Equipo de trabajo.</li> </ul>	<ul style="list-style-type: none"> <li>• Director del proyecto.</li> <li>• Equipo del Proyecto.</li> </ul>
Fases	<ul style="list-style-type: none"> <li>• Planificar.</li> <li>• Realizar.</li> <li>• Evaluar y Controlar.               <ul style="list-style-type: none"> <li>• Cerrar.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Establecer estimaciones para la planificación.</li> <li>• Desarrollar un plan de proyecto.</li> <li>• Obtener el compromiso con el plan.</li> <li>• Monitorizar el proyecto usando el plan.</li> <li>• Gestionar las acciones correctivas hasta el final.</li> </ul>	<ul style="list-style-type: none"> <li>• Planificar el proyecto.</li> <li>• Ejecutar del plan de actividades.</li> <li>• Evaluar y controlar el proyecto.</li> <li>• Cerrar el proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>• Desarrollar un acta de constitución del proyecto.</li> <li>• Desarrollar un plan para la dirección del proyecto.               <ul style="list-style-type: none"> <li>• Dirigir y gestionar el trabajo del proyecto.</li> </ul> </li> <li>• Monitorear y controlar el trabajo del proyecto.               <ul style="list-style-type: none"> <li>• Realizar el control integrado de cambios.                   <ul style="list-style-type: none"> <li>• Cerrar el proyecto o fase.</li> </ul> </li> </ul> </li> </ul>
Entradas	<ul style="list-style-type: none"> <li>• Documentación de Procesos.</li> <li>• Descripción del Proyecto.</li> <li>• Requisitos del Proyecto.</li> <li>• Metas Cuantitativas para el Proyecto.</li> </ul>	<ul style="list-style-type: none"> <li>• Requisitos del proyecto.</li> <li>• Aproximación técnica.               <ul style="list-style-type: none"> <li>• Estimaciones.</li> </ul> </li> <li>• Plan del proyecto.</li> <li>• Compromisos.</li> </ul>	<ul style="list-style-type: none"> <li>• Requisitos del proyecto.</li> <li>• Enunciado de trabajo.</li> <li>• Configuración del software.</li> <li>• Solicitud de cambio.</li> </ul>	<ul style="list-style-type: none"> <li>• Requisitos del proyecto.</li> <li>• Enunciado del trabajo.</li> <li>• Caso de negocio.               <ul style="list-style-type: none"> <li>• Acuerdos.</li> <li>• Factores ambientales de la</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>• Acciones Correctivas o Preventivas.</li> </ul>			<p>empresa.</p> <ul style="list-style-type: none"> <li>• Activos de los procesos de la organización</li> </ul>
Salidas	<ul style="list-style-type: none"> <li>• Documento de Aceptación.</li> <li>• Reporte de Mediciones y Sugerencias de Mejora.</li> <li>• Lecciones Aprendidas.</li> </ul>	<ul style="list-style-type: none"> <li>• Plan del proyecto (actualizado).</li> <li>• Compromisos revisados.</li> <li>• Datos cuantitativos y desviaciones corregidas.</li> </ul>	<ul style="list-style-type: none"> <li>• Plan del proyecto.</li> <li>• Acta de aceptación.</li> <li>• Repositorio del proyecto (actualizado).</li> <li>• Acta de reunión.</li> <li>• Configuración del software (actualizada).</li> </ul>	<ul style="list-style-type: none"> <li>• Plan del proyecto.</li> <li>• Transferencia del producto, servicio o resultado final.</li> <li>• Activos de los procesos de la organización (actualizados). <ul style="list-style-type: none"> <li>• Acta de constitución del trabajo.</li> </ul> </li> <li>• Plan para la dirección del proyecto.</li> <li>• Entregables. <ul style="list-style-type: none"> <li>• Datos de desempeño del trabajo.</li> </ul> </li> <li>• Solicitud de cambio (aprobada).</li> <li>• Informes del desempeño del trabajo.</li> <li>• Registro de cambios (actualizado).</li> </ul>
Productos de trabajo	<ul style="list-style-type: none"> <li>• Plan del Proyecto.</li> <li>• Plan de Desarrollo. <ul style="list-style-type: none"> <li>• Reporte de Seguimiento.</li> <li>• Reporte de Mediciones y Sugerencias de Mejora.</li> </ul> </li> <li>• Documentación de Aceptación.</li> <li>• Lecciones Aprendidas.</li> <li>• Acciones Correctivas.</li> <li>• Minuta(s).</li> <li>• Reporte de Verificación.</li> <li>• Reporte de Validación.</li> <li>• Solicitud de Cambio.</li> </ul>	<ul style="list-style-type: none"> <li>• Descripciones de las tareas.</li> <li>• Descripción de los paquetes de trabajo.</li> <li>• Tamaño y complejidad de las tareas y de los productos de trabajo.</li> <li>• Modelos de estimación. <ul style="list-style-type: none"> <li>• Estimaciones de los atributos. <ul style="list-style-type: none"> <li>• Calendario.</li> <li>• Presupuesto.</li> </ul> </li> </ul> </li> <li>• Aproximación técnica. <ul style="list-style-type: none"> <li>• Registros del rendimiento del proyecto.</li> <li>• Registro de las desviaciones significativas.</li> </ul> </li> <li>• Informe y rendimiento de costos. <ul style="list-style-type: none"> <li>• Registro de las revisiones de los compromisos.</li> <li>• Registro de la</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Solicitud de cambio.</li> <li>• Acta de reunión. <ul style="list-style-type: none"> <li>• Acciones correctivas.</li> <li>• Resultados de verificación.</li> <li>• Resultados de validación.</li> </ul> </li> <li>• Reporte de avance. <ul style="list-style-type: none"> <li>• Respaldo del repositorio del proyecto.</li> </ul> </li> <li>• Plan del proyecto.</li> <li>• Acta de aceptación.</li> </ul>	<ul style="list-style-type: none"> <li>• Enunciado del trabajo del proyecto.</li> <li>• Caso de negocio. <ul style="list-style-type: none"> <li>• Acuerdos.</li> <li>• Acta de constitución del proyecto.</li> </ul> </li> <li>• Registro de interesados.</li> <li>• Calendario de recursos.</li> <li>• Plan para la dirección del proyecto.</li> <li>• Documento de adquisiciones.</li> <li>• Solicitud de cambio (aprobada).</li> <li>• Medidas de control de calidad.</li> <li>• Informes de desempeño.</li> </ul>

		<p>involucración de las partes interesadas.</p> <ul style="list-style-type: none"> <li>• Resultados documentados de la revisión del proyecto.</li> <li>• Listas de las cuestiones que requieren acciones correctivas.</li> <li>• Plan de acciones correctivas.</li> <li>• Resultados de las acciones correctivas.</li> </ul>		
Guías	<ul style="list-style-type: none"> <li>• No se definen.</li> </ul>	<ul style="list-style-type: none"> <li>• No se definen.</li> </ul>	<ul style="list-style-type: none"> <li>• Guía de gestión e ingeniería: grupo de perfil genérico. Perfil básico.</li> </ul>	<ul style="list-style-type: none"> <li>• No se definen.</li> </ul>
Recursos	<ul style="list-style-type: none"> <li>• Herramientas que permitan documentar, manejar y controlar el Plan del proyecto y el Plan de desarrollo.</li> </ul>	<ul style="list-style-type: none"> <li>• Instalaciones físicas.</li> <li>• Ordenadores, periféricos y software. <ul style="list-style-type: none"> <li>• Redes.</li> </ul> </li> <li>• Entornos de seguridad.</li> <li>• Personal del proyecto. <ul style="list-style-type: none"> <li>• Procesos.</li> <li>• Recursos de infraestructura en el entorno de desarrollo, en el entorno de prueba, en el entorno de producción, en el entorno operativo.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Humanos, materiales, estándares, equipo y herramientas (incluyendo la capacitación necesaria).</li> </ul>	<ul style="list-style-type: none"> <li>• Humanos, materiales, estándares, equipo y herramientas (incluyendo la capacitación necesaria).</li> </ul>
Herramientas	<ul style="list-style-type: none"> <li>• Base de Conocimiento (automatizada)..</li> </ul>	<ul style="list-style-type: none"> <li>• CPM.</li> <li>• PERT.</li> <li>• Herramientas para la identificación y análisis de los riesgos.</li> <li>• Taxonomías de riesgos.</li> <li>• Evaluaciones de riesgos.</li> <li>• Listas de comprobación.</li> <li>• Modelos de rendimiento del producto, del proyecto y del proceso. <ul style="list-style-type: none"> <li>• Modelos de costo (COCOMO II, DELPHI).</li> <li>• PBS.</li> <li>• WBS.</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Estructura de descomposición del trabajo (EDT).</li> <li>• Herramientas para el control de versiones.</li> </ul>	<ul style="list-style-type: none"> <li>• Estructura de desglose del trabajo.</li> <li>• Diccionario de la EDT/WBS <ul style="list-style-type: none"> <li>• Sistema de información para la dirección de proyectos.</li> <li>• Herramientas para el control de cambios</li> </ul> </li> <li>• Herramientas de gestión y control de la calidad.</li> </ul>
Medidas	<ul style="list-style-type: none"> <li>• M1 (I1) Comprobar que el Plan del Proyecto y el Plan de Desarrollo consideran los Objetivos establecidos en la Descripción del Proyecto y las Metas</li> </ul>	<p><i>De tamaño:</i></p> <ul style="list-style-type: none"> <li>• Tamaño (estimado) de los productos.</li> <li>• Tamaño (estimado) para los cambios a los productos.</li> <li>• Tamaño (estimado) del código generado.</li> <li>• Tamaño (estimado) de</li> </ul>	<ul style="list-style-type: none"> <li>• Alcance del trabajo.</li> <li>• Tareas y recursos necesarios.</li> <li>• Objetivo del proyecto</li> <li>• Registro del estado del</li> </ul>	<ul style="list-style-type: none"> <li>• Medición del desempeño: Gestión del Valor Ganado (EVM)</li> <li>• Valor Planificado (PV).</li> <li>• Valor Ganado (EV).</li> </ul>

	<p>Cuantitativas para el Proyecto para lograr la consistencia entre éstos.</p> <ul style="list-style-type: none"> <li>• M2 (I2) Comparar el Plan del Proyecto y Plan de Desarrollo contra los Reportes de Seguimiento y Reportes de Actividades correspondientes para conocer la desviación contra lo planificado.</li> <li>• M3 (I3) Comparar el Plan del Proyecto contra el tiempo y costo reales contenidos en el Reporte de Seguimiento para conocer la desviación contra lo estimado.</li> <li>• M4 (I4) Revisar las Minutas para comprobar la realización de las reuniones de avance del proyecto.</li> <li>• M5 (I5) Revisar las Solicitudes de Cambio para comprobar su atención adecuada.</li> </ul>	<p>la prueba completa.</p> <ul style="list-style-type: none"> <li>• Tamaño (estimado) de código entregado.</li> <li>• Unidades (estimadas) de documentación entregada.</li> <li>• Tamaño (proyectado) de los productos (re-estimación basados en los actuales).</li> <li>• Tamaño (estimaciones revisadas).</li> <li>• Tamaño (actual) del código generado.</li> <li>• Tamaño (actual) del código probado completamente.</li> <li>• Tamaño (actual) del código entregado.</li> <li>• Tamaño (actual) de los productos (importantes).</li> <li>• Tamaño (actual) de los cambios a los productos (importantes).</li> <li>• Unidades (actuales) de documentación entregada.</li> <li>• Recursos del ordenador.</li> <li>• Tamaño.</li> <li>• Tamaño (estimado) de los productos.</li> <li>• Tamaño (estimado) para los cambios a los productos.</li> <li>• Tamaño (estimado) del código generado.</li> <li>• Tamaño (estimado) de la prueba completa.</li> <li>• Tamaño (estimado) de código entregado.</li> <li>• Unidades (estimadas) de documentación entregada.</li> <li>• Tamaño (proyectado) de los productos (re-estimación basados en los actuales).</li> <li>• Tamaño (estimaciones revisadas).</li> <li>• Tamaño (actual) del código generado.</li> <li>• Tamaño (actual) del código probado completamente.</li> <li>• Tamaño (actual) del código entregado.</li> </ul>	<p>avance del proyecto.</p> <ul style="list-style-type: none"> <li>• Esfuerzo y costo estimado.</li> <li>• Calendario de tareas del proyecto.</li> <li>• Impacto técnico.</li> <li>• Riesgos del proyecto.</li> <li>• Verificación y validación de los Productos de Trabajo/ Entregables.</li> <li>• Control de versiones y de la línea base para el proyecto.</li> <li>• Duración estimada.</li> </ul>	<ul style="list-style-type: none"> <li>• Coste Real (AC).</li> <li>• Presupuesto hasta la conclusión. <ul style="list-style-type: none"> <li>• Variación de Cronograma (SV).</li> <li>• Variación del costo.</li> </ul> </li> <li>• Variación de la conclusión.</li> <li>• Estimación de la conclusión. <ul style="list-style-type: none"> <li>• Índice de desempeño del cronograma.</li> <li>• Índice de desempeño del costo.</li> </ul> </li> <li>• Estimación hasta la conclusión. <ul style="list-style-type: none"> <li>• Índice de desempeño del trabajo por completar.</li> <li>• Medición del desempeño técnico: pesos, tiempos de transacción, número de piezas defectuosas entregadas, capacidad de almacenamiento, etc. <ul style="list-style-type: none"> <li>• Trabajo terminado.</li> </ul> </li> </ul> </li> <li>• Fechas de inicio de actividades programadas. <ul style="list-style-type: none"> <li>• Fechas de finalización de actividades programadas.</li> <li>• Número de solicitudes de cambio.</li> <li>• Número de defectos.</li> </ul> </li> <li>• Costos reales.</li> <li>• Duración real.</li> </ul>
--	---	--	---	--

		<ul style="list-style-type: none"> <li>• Tamaño (actual) de los productos (importantes).</li> <li>• Tamaño (actual) de los cambios a los productos (importantes).</li> <li>• Unidades (actuales) de documentación entregada.</li> <li>• Tamaño de los (principales) productos de software.</li> <li>• Tamaño de los cambios para los productos de software.</li> <li>• Tamaño proyectado de los productos de software (estimaciones + actuales).</li> </ul> <p><i>De esfuerzo</i></p> <ul style="list-style-type: none"> <li>• Esfuerzo software (estimado) para el proyecto.</li> <li>• Distribución del esfuerzo (estimada) sobre el ciclo de vida del proyecto.</li> <li>• Esfuerzo gastado (para las tareas de planificación).</li> <li>• Esfuerzo gastado en realizar el seguimiento y control.</li> <li>• Esfuerzo (actual) para el proyecto.</li> <li>• Esfuerzo requerido (para las tareas del plan).</li> <li>• Esfuerzo (estimado).</li> <li>• Esfuerzo gastado al realizar el seguimiento y control.</li> </ul> <p><i>De personal:</i></p> <ul style="list-style-type: none"> <li>• Personal (estimado).</li> <li>• Personal (actual).</li> </ul> <p><i>De calendario:</i></p> <ul style="list-style-type: none"> <li>• Fechas planificadas para los hitos del plan del proyecto.</li> <li>• Fechas de terminación para los hitos del plan del proyecto.</li> <li>• Calendario (planificado).</li> <li>• Fechas de terminación (hitos) (planificada).</li> <li>• Fechas de las revisiones</li> </ul>		
--	--	---	--	--

		<p>(planificada).</p> <ul style="list-style-type: none"> <li>• Tiempo de actividades (planificado).</li> <li>• Fecha de los hitos impuestos (planificadas).</li> <li>• Fechas críticas de las dependencias (planificadas).</li> <li>• Otras limitaciones (planificadas).</li> <li>• Fechas planificadas para los hitos del proyecto.</li> <li>• Calendario para el trabajo completado (planificado).</li> <li>• Fecha de terminación (actividades) (planificadas).</li> <li>• Fecha de terminación (otros compromisos) (planificadas).</li> <li>• Calendario (fechas de terminación) (planificado). <ul style="list-style-type: none"> <li>○ Actividades.</li> <li>○ Hitos.</li> <li>○ Compromisos.</li> </ul> </li> <li>• Revisiones de calendario.</li> <li>• Fechas de terminación (actuales) para los hitos de la planificación.</li> <li>• Calendario (actual).</li> <li>• Calendario (fechas de terminación) (actual). <ul style="list-style-type: none"> <li>○ Actividades.</li> <li>○ Hitos.</li> <li>○ Compromisos.</li> <li>○ Estado técnico.</li> </ul> </li> <li>• Fechas de terminación (actividades) (actuales).</li> <li>• Fechas de terminación (hitos) (actuales).</li> <li>• Fechas de terminación (otros compromisos) (actuales).</li> <li>• Calendario (fechas de terminación) (actuales). <ul style="list-style-type: none"> <li>○ Actividades.</li> <li>○ Hitos.</li> <li>○ Compromisos.</li> </ul> </li> <li>• Tiempo planificado (actual) por proyecto, etapa del proceso, categoría de programa, tamaño de programa, grado de modificación</li> </ul>		
--	--	--	--	--

		<p>del programa, y demás.</p> <ul style="list-style-type: none"> <li>• Rendimiento del calendario.</li> <li>• Trabajo completado (para las tareas del plan).</li> <li>• Trabajo completado (estimado) contra tiempo y/o costes y esfuerzo estimados.</li> <li>• Trabajo completado.</li> </ul> <p><i>De costo:</i></p> <ul style="list-style-type: none"> <li>• Costos (estimados) del proyecto.</li> <li>• Distribución del costo (estimada) sobre el ciclo de vida del proyecto.</li> <li>• Costos (actuales) del proyecto.</li> <li>• Productividad.</li> <li>• Datos de productividad.</li> </ul> <p><i>De riesgo:</i></p> <ul style="list-style-type: none"> <li>• Riesgos asociados con el costo, recursos, calendario, y aspectos técnicos. <ul style="list-style-type: none"> <li>○ Prioridades.</li> <li>○ Contingencias.</li> </ul> </li> <li>• Prioridades del riesgo (inicial).</li> </ul> <p><i>De calidad:</i></p> <ul style="list-style-type: none"> <li>• Tendencias organizacionales de la calidad.</li> <li>• Metas de calidad para los procesos de la organización.</li> <li>• Metas de calidad para los procesos de cada proyecto.</li> <li>• Historial de defectos de cada programa.</li> <li>• Tendencia de calidad para cada proyecto.</li> </ul> <p><i>Otros:</i></p> <ul style="list-style-type: none"> <li>• Fondos requeridos (para las tareas del plan).</li> <li>• Recursos críticos para cada componente mayor.</li> <li>• Liberar contenidos para construcciones sucesivas (planificado).</li> <li>• Liberar contenidos para construcciones sucesivas</li> </ul>		
--	--	---	--	--

		<p>(actual).</p> <ul style="list-style-type: none"> <li>• Estimaciones.</li> <li>• Información necesaria para reconstruir y verificar las estimaciones.</li> <li>• Otros recursos gastados al realizar el seguimiento y control.</li> <li>• Actividad de cambio de la estimación de tamaño.</li> <li>• Actividad de cambio de estimación de costo.</li> <li>• Actividad de cambio de estimación de recursos críticos del ordenador.</li> <li>• Actividad de cambio de calendario.</li> <li>• Medición de los valores reales de los parámetros de planificación del proyecto, comparar los valores reales con los estimados en el plan e identificar las desviaciones significativas.</li> </ul>		
Activos	<ul style="list-style-type: none"> <li>• Plantilla para el plan del proyecto</li> <li>• Plantilla para el reporte de actividad</li> <li>• Plantilla para el reporte de mediciones y sugerencias de mejora</li> <li>• Plantilla del reporte de seguimiento</li> <li>• Plantilla para el documento de aceptación</li> <li>• Lecciones aprendidas <ul style="list-style-type: none"> <li>• Plantilla para las acciones correctivas <ul style="list-style-type: none"> <li>• Minuta</li> </ul> </li> <li>• Plantilla para el reporte de verificación</li> <li>• Plantilla para el reporte de validación</li> </ul> </li> <li>• Plantilla para el plan de adquisiciones y capacitación</li> <li>• Plantilla para la descripción del proyecto</li> </ul>	<ul style="list-style-type: none"> <li>• Plantilla para la estructura de descomposición del trabajo la WBS.</li> <li>• Plantilla para la descripción del trabajo.</li> <li>• Modelos de estimación del coste de software(COCOMO II)</li> <li>• Plantilla del plan del proyecto</li> <li>• Plantilla para el plan de gestión de la configuración</li> <li>• Plantilla para el plan de aseguramiento de la calidad.</li> <li>• Plantilla para el plan de mejora de procesos de la organización</li> <li>• Plantilla para el plan de formación de la organización. <ul style="list-style-type: none"> <li>• Plantilla para los criterios de acciones correctivas de los proyectos. <ul style="list-style-type: none"> <li>• Plantilla para la documentación de riegos</li> </ul> </li> </ul> </li> <li>• Plantilla para generar</li> </ul>	<ul style="list-style-type: none"> <li>• Plantilla para el plan del proyecto</li> <li>• Plantilla del enunciado del trabajo</li> <li>• Plantilla para la especificación de los requisitos</li> <li>• Plantilla para el registro de estado del avance del proyecto</li> <li>• Plantilla para el acta de aceptación</li> <li>• Plantilla para las solicitudes de cambio <ul style="list-style-type: none"> <li>• Plantilla de configuración del software</li> </ul> </li> <li>• Plantilla para el reporte del avance</li> <li>• Plantilla para los resultados de verificación</li> <li>• Plantilla para las acciones correctivas</li> <li>• Líneas base para el proyecto</li> </ul>	<ul style="list-style-type: none"> <li>• Procesos estándar de la organización, políticas y definiciones de procesos.</li> <li>• Plantilla para el acta de constitución del proyecto. <ul style="list-style-type: none"> <li>• Guías estandarizadas, instrucciones de trabajo, criterios para la evaluación de propuestas y criterios para la medición del desempeño</li> </ul> </li> <li>• Plantilla para el plan de dirección del proyecto, que incluye: <ul style="list-style-type: none"> <li>○ Guías y criterios para adoptar el conjunto de procesos estándar de la</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>• Plantilla para el plan de manejo de riesgos <ul style="list-style-type: none"> <li>• Plantilla para la solicitud de cambios</li> <li>• Plantilla para las acciones correctivas</li> </ul> </li> <li>• Plantilla para el plan de desarrollo <ul style="list-style-type: none"> <li>• Plantilla para el reporte de seguimiento del proyecto</li> </ul> </li> <li>• Lecciones aprendidas</li> </ul>	<p>estimaciones de tamaño del producto de trabajo</p> <ul style="list-style-type: none"> <li>• Descripción de la metodología de diseño del producto de trabajo.</li> <li>• Plantilla para llevar a cabo revisiones de gestión</li> <li>• Plantilla para el plan para la iteración con las partes interesadas</li> <li>• Plantilla para establecer y mantener un plan global del proyecto</li> <li>• Base de datos históricos</li> <li>• Plantilla para el registros de revisiones correctivas de los compromisos</li> <li>• Plantilla para el registro de la involucración de las partes interesadas</li> <li>• Plantilla de revisiones del proyecto <ul style="list-style-type: none"> <li>• Plantilla de las revisiones de los hitos</li> <li>• Listas de cuestiones para acciones correctivas</li> </ul> </li> <li>• Plantilla para el plan de acciones correctivas <ul style="list-style-type: none"> <li>• Platilla para el seguimiento de acciones correctivas.</li> </ul> </li> </ul>		<p>organización con el fin de que de que satisfagan las necesidades específicas del proyecto.</p> <ul style="list-style-type: none"> <li>◦ Guías o requisitos para el cierre del proyecto, tales como criterios de validación y aceptación del producto.</li> </ul> <ul style="list-style-type: none"> <li>• Procedimientos de control de cambios, incluidos los pasos según los cuales se modificarán los estándares, políticas, planes y procedimientos oficiales de la organización, o cualquier documento del proyecto, y la manera en que aprobará y validará cualquier cambio.</li> <li>• Archivos de proyectos anteriores (p.ej. líneas de base del alcance, de costos, del cronograma y de medición del desempeño, calendario del proyecto, diagramas de red del cronograma, )</li> <li>• Información histórica y base de conocimientos de lecciones aprendidas. <ul style="list-style-type: none"> <li>• Base de conocimiento de gestión de la configuración que contiene las versiones y</li> </ul> </li> </ul>
--	---	--	--	---

				<p>líneas base de todos los estándares, políticas y procedimientos oficiales de la organización, y cualquier otro documento del proyecto.</p> <ul style="list-style-type: none"><li>• Guías e instrucción de trabajo estandarizadas.</li><li>• Requisitos de comunicación permitidos y el tiempo de conservación de los registros, así como los registros de seguridad.</li><li>• Procedimientos para la gestión de incidentes y defectos que definen los controles para incidentes y defectos que definen los controles para incidentes y defectos, la identificación y la solución de los mismos, así como el seguimiento de los elementos de acción.</li><li>• Base de datos para la medición de procesos, que se utiliza para recopilar y tener a disposición los datos de mediciones de procesos y productos.</li><li>• Archivos de proyectos anteriores.</li><li>• Base(s) de datos sobre la gestión de incidentes y</li></ul>
--	--	--	--	--

				<p>defectos, que contiene(n) el estado histórico de los mismos, información de control, resolución de los incidentes y defectos, así como los resultados de las acciones emprendidas.</p> <ul style="list-style-type: none"><li>• Los requisitos de la comunicación de la organización<ul style="list-style-type: none"><li>• Los procedimientos de control financiero.</li><li>• Los procedimientos para el control de cambios, incluidos los relacionados con las variaciones del alcance, del cronograma, del costo y de la calidad.</li><li>• Los procedimientos del control de riesgos, que incluyen las categorías de riesgos, las definiciones de la probabilidad y el impacto y la matriz de la probabilidad y el impacto.</li></ul></li><li>• La base de datos de medición de procesos.</li><li>• La base de datos de lecciones aprendidas.<ul style="list-style-type: none"><li>• Los procedimientos y control de cambios incluyendo los pasos para</li></ul></li></ul>
--	--	--	--	---

				<p>modificar los estándares, políticas y planes oficiales de la organización, y cualquier otro documento del proyecto, y la descripción de cómo se aprobará, validará e implementará cualquier cambio.</p> <ul style="list-style-type: none"> <li>• Procedimientos para aprobar y emitir autorizaciones de cambio.</li> <li>• Documentos del proyecto. <ul style="list-style-type: none"> <li>• Base de conocimiento de gestión de la configuración, que contiene las versiones y líneas base de todos los estándares, políticas y procedimientos oficiales de la organización, y cualquier otro documento del proyecto.</li> <li>• Guías o requisitos para el cierre del proyecto o fase.</li> <li>• Información histórica y bases de conocimiento de lecciones aprendidas.</li> </ul> </li> </ul>
--	--	--	--	---

### 3.3. Definir y/o seleccionar el lenguaje de representación y creación de objetos

Como ya se indicó anteriormente, en los últimos años se ha prestado especial atención en definir eficientemente los procesos relacionados con la IS. En este sentido, las investigaciones recientes demuestran que los metamodelos y las ontologías están destacando en esta tarea. De acuerdo con Roszkowski y Roszkowska (2013), la introducción de metamodelos en el ámbito empresarial promueve el concepto de gestión total de la calidad al desarrollar productos de software;

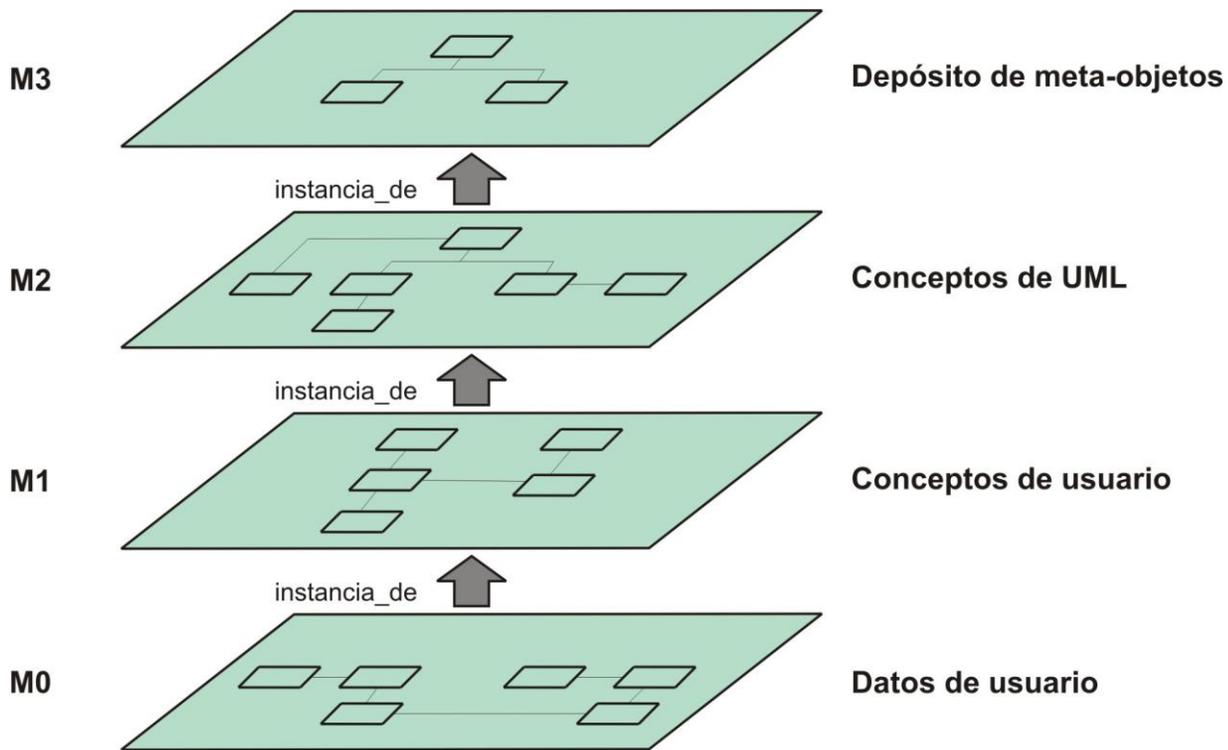
dichos metamodelos varían en su enfoque abordando la definición de procesos, la medición y organización del proceso, la evaluación del proceso, entre otras áreas. Por lo general estos metamodelos, a pesar de que se han desarrollado de forma independientemente, comparten conceptos básicos en su concepción, concretamente el lenguaje de representación.

Así, uno de los lenguajes de modelado más utilizado en el diseño de metamodelos es UML. UML, que fue diseñado para ser un lenguaje de modelado de propósito general, es un conjunto de notaciones de modelado orientado a objetos que permite la representación gráfica de cualquier sistema. De hecho, este lenguaje de modelado permite, a través del metamodelo SPEM, generar diferentes representaciones de un proceso que pueden ser aplicables a cualquier dominio y problema. Así pues, es un tanto difícil negar que UML reúne las mejores prácticas de modelado y que incorpora una notación que ha demostrado su extensa utilidad en la práctica, tanto en el entorno académico como en el industrial (Fu, Luo, & Luo, 2014). En este sentido, entre las características más notorias de UML se encuentran las siguientes:

- Provee a los usuarios de un lenguaje fácil de usar y altamente visual para el desarrollo de modelos.
- Proporciona mecanismos de especialización y extensión de conceptos elementales definidos en el núcleo de su marco conceptual.
- Brinda soporte a especificaciones independientes de lenguajes de programación y procesos específicos de desarrollo.
- Alienta a la industria para aportar nuevas herramientas al mercado.
- Proporciona conceptos de desarrollo de alto nivel como componentes, colaboraciones, entornos de trabajo, y patrones.
- Integra las buenas prácticas en los procesos de desarrollo.

Cabe mencionar que la especificación del lenguaje UML se sitúa en el Nivel M2 de la Arquitectura dirigida por Modelos (MDA, por sus siglas en inglés) (véase Figura 3.2), y su definición se basa en los siguientes principios:

1. Modularidad: se refiere a la agrupación de los constructores en paquetes y a la organización de aspectos comunes en meta-clases.
2. Estratificación: la división por capas es aplicada de dos formas en el metamodelo. La primera es la estructura de los paquetes que se estratifica para separar los constructores que forman el núcleo del metalenguaje y los constructores de alto nivel que los utilizan. La segunda es el patrón de la arquitectura de cuatro capas (mostrada en la Figura 3.6) que separa los conceptos en diferentes niveles de abstracción.
3. Extensibilidad: UML se puede extender de dos maneras: (1) mediante el uso de perfiles es posible crear nuevos lenguajes adaptando los constructores a plataformas o dominios específicos, y (2) mediante la reutilización de parte del paquete *Infrastructure Library* para aumentarlo con nuevas meta-clases y meta-relaciones, y así crear un nuevo lenguaje de modelado que, por definición, sería ya un metamodelo diferente.
4. Reusabilidad: ésta se consigue mediante librerías flexibles del metamodelo que permiten definir otros metamodelos, tales como MOF MOF (*Meta Object Facility*) o CWM (*Common Warehouse Meta-model*).



**Figura 3.6.** Arquitectura de MDA propuesta por OMG (Henderson-Sellers, 2011).

Así pues, UML es un intento general de coordinación de varios estándares para proporcionar un soporte formal y completo a la filosofía general de MDA (Mora, Ruiz, García, & Piattini, 2006). En este sentido, de acuerdo con (Chaudron, Heijstek, & Nugroho, 2012) una de las principales ventajas del lenguaje es el uso de modelos para la comprensión de un problema en un nivel abstracto, lo cual obliga al usuario a pensar tempranamente en los aspectos de desarrollo reduciendo así el costo económico.

En base a todos estos argumentos y puesto que UML es el lenguaje que más ha sido utilizado para la creación de metamodelos (e.g., Gonzalez-Perez, et al., 2005; García, et al., 2006; Da Cruz & Faria, 2010; Capilla, Zimmermann, Zdun, Avgeriou, & Küster, 2011; Do, Cook, Campbell, Scott, Robinson, Power, & Tramoundanis, 2012), se decide escogerlo como lenguaje de representación de los objetos que conforman el metamodelo propuesto, que es presentado en la siguiente sección de este capítulo.

### 3.4. Creación del metamodelo y el escenario inicial

En la actualidad, la mayoría de los “modelos” o “guías” para la gestión de los proyectos no fueron creadas específicamente para los pequeños equipos, aunque existen derivaciones o propuestas exitosas (e.g., Callegari & Bastos, 2007; Mas & Mesquida, 2013; O’Sheedy & Sankaran, 2013; Mesquida & Mas, 2014), que están influenciadas por una de las recopilaciones más utilizadas de buenas prácticas para la gestión de proyectos, la Guía PMBoK del PMI. En este contexto, dentro de la Ingeniería de Software existe una lista extensa de modelos (o marcos de trabajo) que sirven de referencia para establecer y controlar eficientemente un proceso de desarrollo de software. Muchos de estos modelos de proceso proporcionan orientación específica, actividades, y roles para abordar los problemas de gestión subyacentes a cualquier proyecto de software, sin embargo carecen de una solución más integrada.

Por lo tanto, es evidente que, por un lado, es necesario desarrollar una solución más comprensiva que integre mejor los modelos para el desarrollo del software y las mejores prácticas y conceptos de la gestión de proyectos. No obstante, por otro lado, la complejidad y el número de proyectos con los que el jefe de proyectos de una empresa pequeña debe lidiar simultáneamente se incrementan cada día. Es decir, a medida que se incrementa el número de proyectos, el jefe de proyectos debe hacer frente a un mayor número de variables. Así, la respuesta a este planteamiento parece residir en las herramientas que apoyan la toma de decisiones productivas y de gestión en el contexto de una pequeña empresa. Sin embargo, este esfuerzo requiere una visión más amplia de los modelos actuales creados para abordar ambas áreas (por supuesto que se ha analizado en el estado del arte de esta tesis que es poco probable que exista un modelo universal que ajuste cada esfuerzo realizado en el desarrollo de software a las particularidades de las pequeñas empresas, pero, como se ha señalado también, la integración de los modelos en un metamodelo puede proporcionar un medio para desarrollar herramientas o proponer la automatización de las tareas que conduzcan a resultados más exitosos).

En este sentido, es verdad que en la actualidad muchos procesos para el desarrollo de software están a disposición de las pequeñas empresas, pero por lo general éstos no abordan adecuadamente la dimensión de la gestión de los proyectos -a menudo proporcionan simplemente un conjunto de prácticas que apoyan a las actividades ahí sugeridas y a los flujos de trabajo asociados (e.g., la simplificación de ISO/IEC 29110 y PMBoK, y la integración de Scrum y PMBoK, ambas analizadas en el estado del arte del Capítulo 2 de esta tesis). Además, la mayoría de las guías de las que puede disponer una pequeña empresa (e.g., PMBoK) no fueron concebidas como un proceso en sí y, por lo tanto, no se señala cómo deben realizar las actividades dentro de un proyecto para el desarrollo correcto de un producto final.

Así, con el fin de proporcionar un proceso más accesible para la pequeña empresa, que incluya la dimensión de gestión de proyectos en un nivel más comprensible, es necesario guiar apropiadamente a sus equipos en la aplicación de los conocimientos sobre la gestión de los proyectos dentro del desarrollo de software. Por esta razón, esta tesis pretende contribuir con esta tarea a través de la provisión de un metamodelo que facilite la comprensión de los conceptos básicos proporcionados por los cuerpos de conocimiento y modelos disponibles, el análisis de los aspectos comunes entre ellos, y la creación de un mecanismo integrado que guíe a los jefes de proyectos en la realización de una mejor gestión y, en consecuencia, en la obtención de mejores productos de software.

En este sentido, en la actualidad es común que el metamodelado dentro de la Ingeniería de Software siga una de dos posibles arquitecturas. La arquitectura de OMG, mostrada en la Figura 3.6, que se basa en el metamodelado estricto (Atkinson, 1997) donde la única relación permitida entre los niveles es denominada “instancia\_de”. Es decir, en los estándares de OMG, se dice que un objeto M0 es una instancia de una clase en el Nivel M1; se dice que una clase en el Nivel M1 es una instancia de una metaclass en el Nivel M2; y así sucesivamente. Sin embargo, los especialistas han argumentado que una mejora a esta arquitectura sería considerar a las entidades del Nivel M1 como *clabjects* (es decir, que sean clases y objetos al mismo tiempo). De acuerdo con Gonzalez-Perez y Henderson-Sellers (2007), las *clabjects* son consistentes con la noción de los *powertypes*<sup>20</sup>. Así, la otra arquitectura utilizada actualmente define capas multinivel que pueden ser instanciadas a través

---

<sup>20</sup> El “patrón *powertype*” se define como un par de clases en las que una de ellas (la *powertype* o supertipo) particiona a la otra haciendo que las instancias de la primera sean subtipos de la otra (Gonzalez-Perez & Henderson-Sellers, 2006). Así, en el contexto de la definición de metodologías para el desarrollo de software, este patrón combina las ventajas importantes de otros enfoques de metamodelado y permite la integración de aspectos documentales de las metodologías. Con este enfoque, los componentes de una metodología y las entidades de los proyectos pueden ser descritas directamente por el mismo metamodelo.

de un patrón (Gonzalez-Perez & Henderson-Sellers, 2008), el cual permite generar instancias como subtipos de otras clases/tipos. Por lo tanto, considerando los conceptos base de esta última arquitectura se define en esta tesis un modelo conceptual para la creación del metamodelo, de tal forma que sea posible definir un proceso para la gestión de proyectos en base a la combinación de los elementos definidos en cada capa a través del concepto de patrón.

Este modelo conceptual (véase Figura 3.7) representa pues una integración de elementos que puede conducir a la formalización de conceptos en la pequeña empresa, y utiliza definiciones formales de un proceso para generar herramientas que apoyen la gestión de los proyectos. Como se observa, este enfoque también puede facilitar la adaptación de un proceso de acuerdo a las características de un equipo/empresa puesto que se pretende usar un modelo de proceso como parámetro de entrada para crear un proceso específico (a partir de él) para un entorno de desarrollo. El objetivo de este modelo conceptual es proporcionar un medio para facilitar el entendimiento y uso de un proceso dentro de los pequeños equipos, mediante la representación de las entidades relacionadas con la gestión en cuatro diferentes capas de abstracción que se definen de la siguiente manera:

- Nivel M0: Los datos generados por todos los proyectos representan el “mundo real” de la arquitectura. Es decir, una empresa debe ser capaz de considerar su elemento más preciado, la información, para definir un proceso que sea eficiente para una situación en particular. Así, los datos históricos obtenidos mediante el desarrollo de proyectos exitosos o fracasados, son útiles para aprender a predecir a largo plazo el comportamiento de sus procesos. Esta capacidad de predicción se dará con el uso prolongado del metamodelo a través de la integración con M1.
- Nivel M1: Esta integración se lleva a cabo mediante la inclusión de los modelos de referencia necesarios para definir un proceso, un mecanismo de evaluación de las entidades del proceso, y las plantillas requeridas para apoyar la ejecución de cualquier actividad relacionada con el proceso de gestión. Así, en el Nivel M1 se proporciona, a través de modelos concretos que definen las actividades específicas para la gestión de proyectos, los elementos que serán instanciados para definir un proceso específico que se adecue a las necesidades del proyecto de la empresa. En este sentido, la definición y evaluación de un proceso son aspectos importantes dentro de esta arquitectura, puesto que no es posible definir (o adaptar) un nuevo proceso que le sea útil a un equipo sin determinar primero cuáles son sus habilidades actuales mediante la evaluación.
- Nivel M2: La definición de un nuevo proceso, a partir de uno o varios anteriores, reside en la necesidad de adecuar una descripción a un contexto específico. El concepto de patrón que se encuentra inmerso en M2 permite que el jefe de proyectos tome de una metodología, o varias, únicamente lo que le sirve con el objetivo de desarrollar un proyecto específico.
- Nivel M3: El depósito de todos los metaelementos anteriores permite utilizar, al igual que el metamodelo SPEM de UML, instancias de los diferentes niveles para crear procesos diferentes a partir de otros que ya existían con anterioridad.

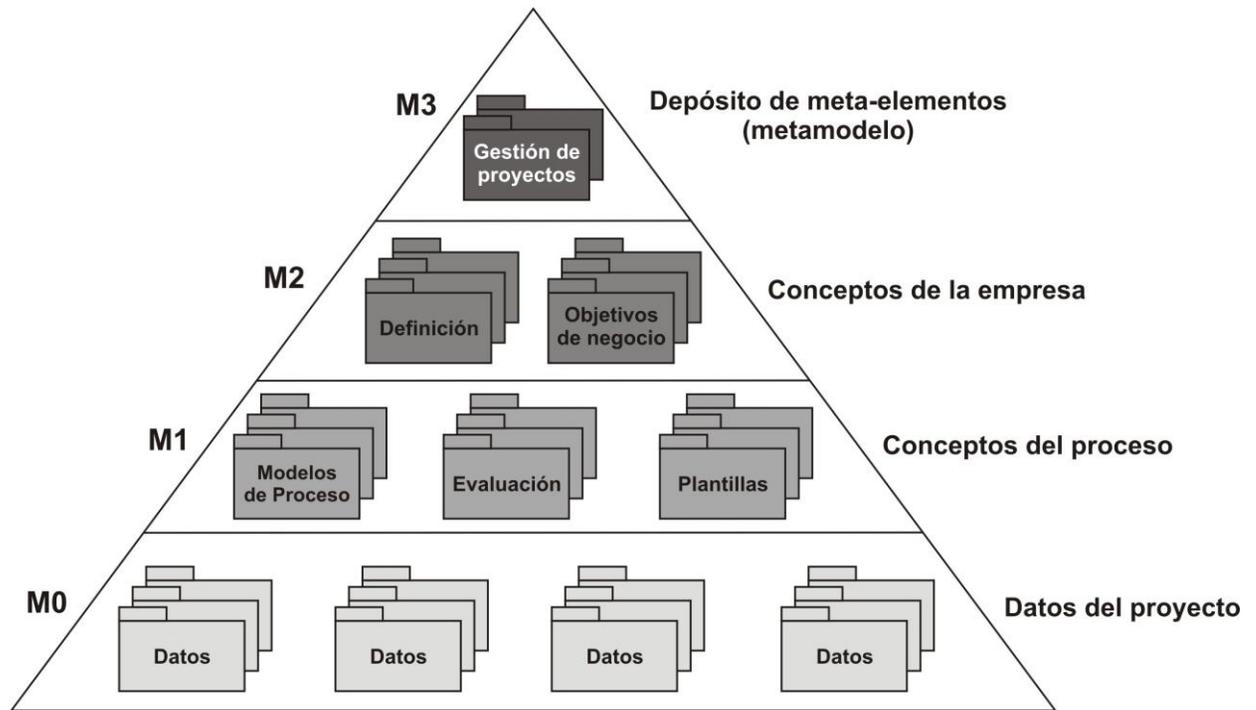


Figura 3.7. Modelo conceptual propuesto.

De acuerdo a este modelo conceptual, a través de los mecanismos establecidos en M2 un jefe de proyectos es capaz de definir un proceso para gestionar y desarrollar un proyecto considerando los objetivos de negocio de su empresa (conceptos de la empresa), y el equipo de desarrollo utiliza los elementos dentro de M1 (conceptos del proceso) y M0 (datos del proyecto) para desarrollar el trabajo establecido previamente por este jefe de proyectos. Así pues, la instanciación de estos elementos no está obligada a establecer únicamente una relación “instancia\_de”, sino que a través del concepto de patrón M3 puede tomar lo que le sirva de M1, o M2 los datos que requiera de M0, por ejemplo. Así, cuando un proceso es adaptado<sup>21</sup> para un proyecto en M2 (es decir, cuando se crea una instancia real desde M1), las actividades a realizar se rigen por una determinada duración (semanas, por ejemplo). Esto significa que cualquier actividad incluida en el proceso nuevo debe tener un valor específico para este atributo (duración), y que para facilitar la tarea de estimación podría obtenerse información útil al respecto de M0. Sin embargo, como todas las actividades deben tener una duración; será necesario que éstas sean asignadas en M3 donde se define un modelo genérico para un proyecto específico.

En este sentido, siguiendo este modelo conceptual se propone el metamodelo mostrado en la Figura 3.8 con la intención de facilitar la definición de un proceso para planificar y controlar los proyectos de software de una empresa pequeña. Cabe resaltar que este metamodelo resume aquellos elementos esenciales para que un pequeño equipo lo pueda utilizar. Es decir, a través de éste se pretende comenzar a madurar a la empresa a nivel de proyectos para que, en un futuro, después madure sus procesos.

<sup>21</sup> De acuerdo con Gonzalez-Perez y Henderson-Sellers (2006), la adaptación de un proceso a través de un metamodelo se puede ver como la aplicación de varias metodologías a un proyecto en particular. Esta adaptación implica el utilizar diferentes instancias de dicha metodología en el desarrollo de los proyectos y, finalmente, en la obtención del software final. Por lo tanto, tal adaptación debe distinguir qué elementos del proceso se deben utilizar o crear, y cómo se relacionan entre sí, para resolver un problema. Por lo general, los jefes de proyectos, junto con el resto del equipo de desarrollo, realizan esta adaptación (en el caso particular del modelo conceptual propuesto, a través de los últimos tres niveles de abstracción).

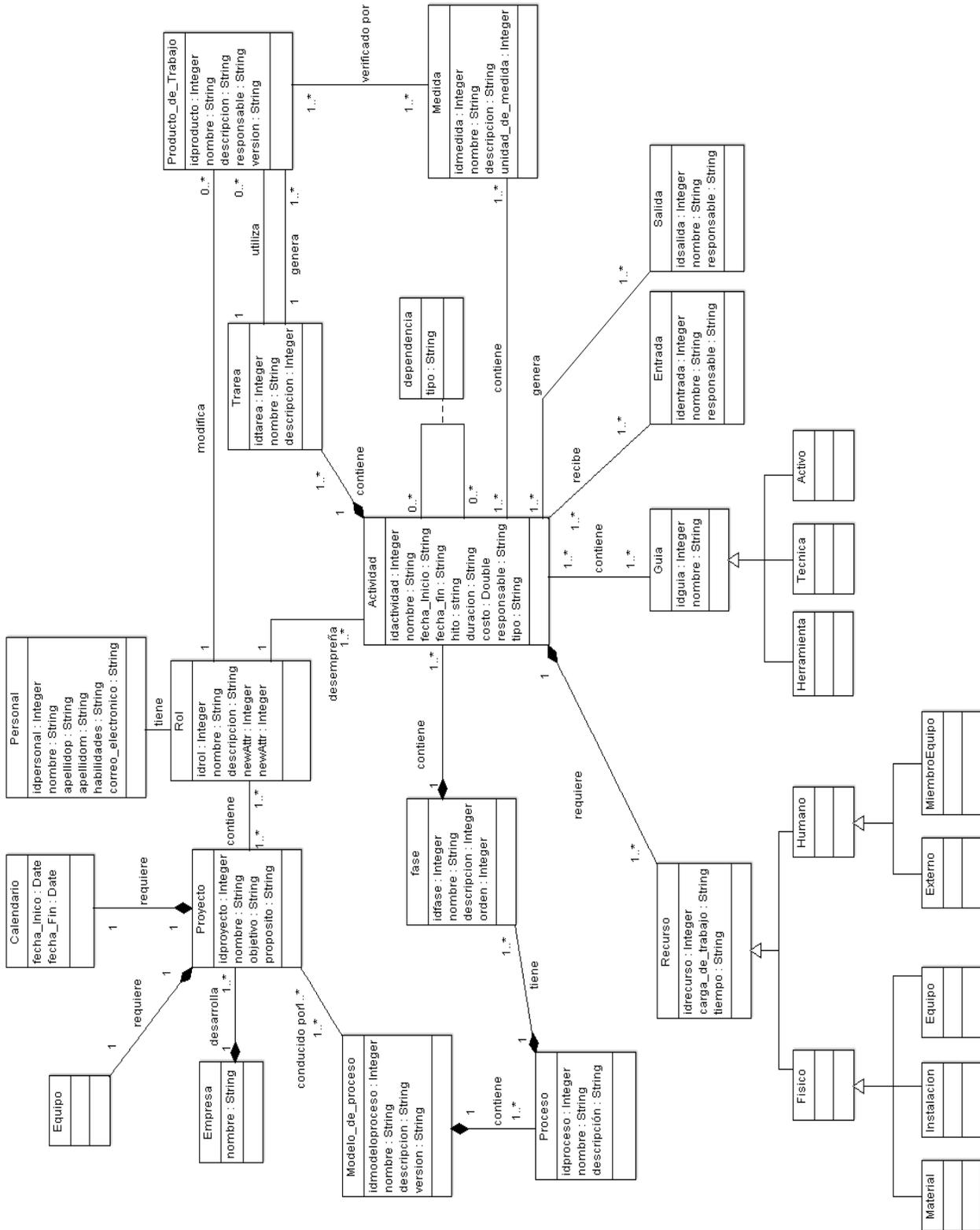


Figura 3.8. Metamodelo propuesto para la definición de los procesos de planificación y control de proyectos.

Uno de los principales conceptos de la gestión de los proyectos es el de las partes interesadas relevantes (i.e., todos aquellos personas u organizaciones que pueden afectar o verse afectados por el proyecto. Las partes interesadas relevantes participan en actividades (o tareas) que se clasifican dentro de una serie de áreas de conocimiento, y pueden ser apoyadas por herramientas y técnicas. Las actividades consumen y generan productos de trabajo (o entregables) a través del uso de los recursos apropiados. Así, más que un simple conjunto de actividades, este metamodelo considera que un proyecto se define como “un esfuerzo temporal emprendido para crear un único producto, servicio o resultado”. Sin embargo, hablar de partes interesadas relevantes en proyectos desarrollados por equipos pequeños puede confundir a sus integrantes. En este sentido, la solución propuesta en esta tesis pretende definir un número pequeño de roles de tal forma que sea posible facilitar la definición de un proceso eficiente.

En este contexto, el metamodelo se ha diseñado a través de un análisis detallado de los conceptos que se encuentran en las referencias más utilizadas para implantar la gestión y desarrollo de un proyecto en cualquier tipo de empresa. La Figura 3.8 muestra que casi la mitad del modelo (del lado derecho) se dedica a expresar las actividades en términos de entradas, salidas, productos de trabajo y medidas, puesto que se considera imperante definir, primeramente, la forma de trabajo del equipo pequeño para desarrollar exitosamente sus proyectos. En el lado superior izquierdo de la figura, se introduce el concepto de una empresa que desarrolla un conjunto de proyectos dirigidos por un actor determinado. Mientras que en la parte inferior se refleja la importancia de los recursos apropiados para lograr el objetivo del proyecto. De esta manera, el metamodelo cubre las tres dimensiones críticas definidas en la Figura 3.2. A continuación, se analiza el escenario inicial de este metamodelo.

El metamodelo propuesto respeta la noción de un **proyecto** tal cual lo hacen modelos como PMBoK o PSP. Es decir, las **empresas** desarrollan proyectos a través de **equipos** que realizan actividades establecidas, regularmente por un **modelo de procesos**, dentro de un **calendario** específico. En este sentido es importante mencionar que un modelo de procesos es un conjunto estructurado de elementos que describen las características de un proceso efectivo (o maduro) y de calidad, indicando “qué hacer”, no “cómo hacerlo” ni “quién lo hace”. Esto no quiere decir que en el contexto de las pequeñas empresas necesariamente se siga un modelo formal de procesos, como lo son CMMI-DEV v1.3 o MoProSoft, porque de hecho es bastante difícil que se haga; pero sí se suelen utilizar modelos más convencionales, por ejemplo, un modelo en cascada o incremental (i.e., ciclos de vida) los cuales también son conocidos como modelos de procesos. Así pues, estos modelos de procesos están compuestos por una colección de **fases**. Cada fase está compuesta a su vez por un conjunto de **actividades**. Es importante mencionar que dada la arquitectura de este metamodelo, no se establece por default un conjunto estricto de fases o actividades; de acuerdo al modelo conceptual mostrado anteriormente, cuando se crea una instancia para un proyecto específico, es necesario que, a través del patrón, el jefe de proyectos escoja la secuencia de fases y sus actividades de acuerdo con el producto que se va a desarrollar.

Al mismo tiempo, las actividades se pueden descomponer en acciones más concretas llamadas **tareas** que requieren de **productos de trabajo** para facilitar su ejecución. Es posible que las actividades tengan **dependencias** entre ellas, lo cual permite la definición del orden en que éstas deben realizarse dentro del proyecto. Las actividades suelen ser apoyadas por algún tipo de **guía** (**herramienta, técnica, activo**) de tal forma que sea posible verificar el procedimiento a realizar antes de su ejecución. Las actividades normalmente producirán **salidas** (e.g., entregables, documentos, artefactos), y dependerán de **entradas** para generar un resultado.

Los **roles** en el metamodelo están divididos de acuerdo al tipo de actividad que se vaya a realizar (ya sea una actividad de gestión o de desarrollo). Así pues, cada actividad debe ser realizada por uno o más roles, algo típico de los pequeños equipos. Una relación similar puede ocurrir con los

**recursos (físicos o humanos)** puesto que una actividad puede depender de cierta infraestructura o conocimiento para poder alcanzar el objetivo del proyecto. Por otro lado, cualquier actividad determinada tiene un solo responsable (tal cual lo indican las buenas normas de gestión de un proyecto), con lo que se intenta eliminar el problema común de las pequeñas empresas que modifican continuamente los roles en cada proyecto. Una actividad puede utilizar un producto de trabajo como entrada o salida, pero también puede modificar uno existente; así, este concepto ha sido ampliado para permitir las tres asociaciones distintivas: *generar*, *modificar*, y *utilizar* como insumos (i.e., una actividad puede generar un producto de trabajo, o bien modificar o utilizar alguno ya existente). Es importante mencionar también que se ha indicado la asociación “modificar” entre el producto de trabajo y el rol, puesto que, además de que ya se había definido que la tarea puede modificar un artefacto, es la persona que desempeña dicho rol la que haría el cambio, posiblemente, sin realizar alguna tarea. Se añadió esta consideración para permitir la automatización de un proceso instanciado considerando el hecho de que no se podía permitir eliminar una actividad que crea un producto de trabajo si existe otra actividad que modifica o utiliza como entrada ese mismo producto.

Cada producto de trabajo debe mostrar información sobre su versión (i.e., control de versiones) y su tipo, ya sea “externo” (cuando está sujeto a la aprobación del responsable o cliente) o “interno” (cuando se entrega total o parcialmente al responsable de la actividad). Por último, un producto de trabajo y una actividad deben utilizar **medidas** para ser verificados y determinar el éxito en su uso, respectivamente.

Una vez que el metamodelo ha sido definido, es necesario que, considerando el modelo conceptual propuesto en la Figura 3.7, se establezca el concepto de patrón de tal forma que sea posible probar su implantación en una empresa real. En este sentido, el paso final en la creación del metamodelo, su prueba, analiza este elemento faltante de la arquitectura conceptual.

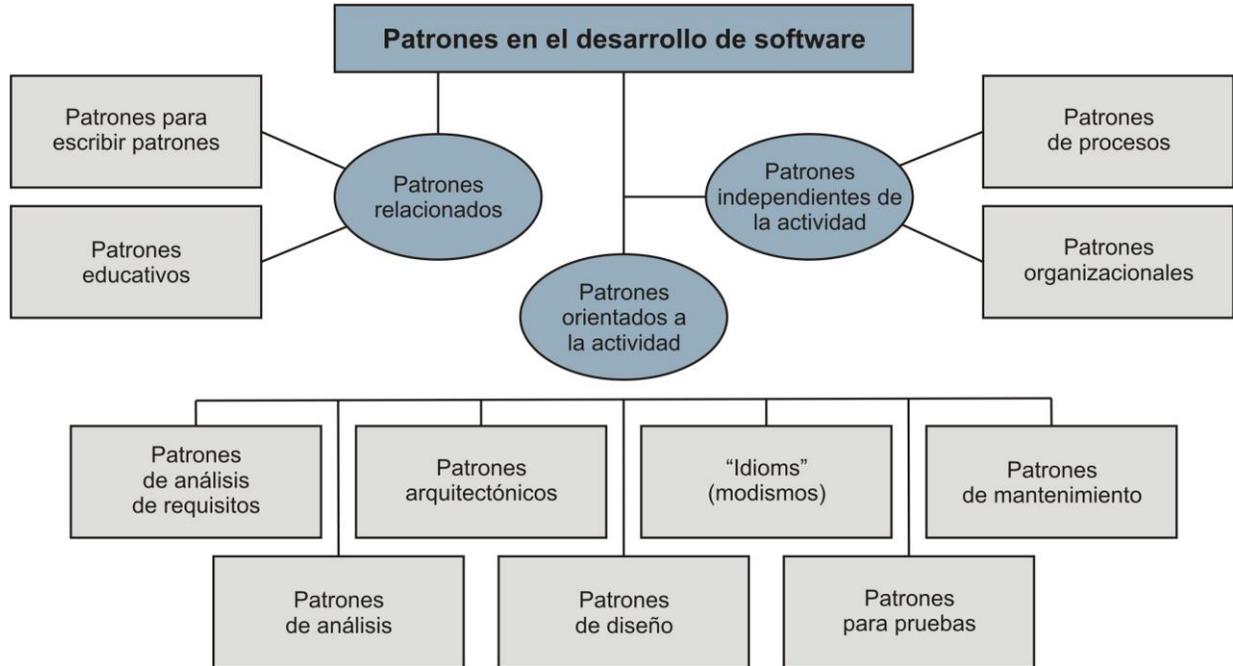
### 3.5. Probar el metamodelo

El término “patrón” se originó dentro del campo de la Arquitectura, siendo Christopher Alexander (1936) quién por primera utilizara este término de manera similar a como se utiliza dentro de la Ingeniería de Software. Así, Alexander expresó que un patrón describía, en primera instancia, un problema que ocurre repetidamente en uno entorno, para describir posteriormente el núcleo de la solución a dicho problema, de tal forma que ésta pudiera ser utilizada un millón de veces sin hacerlo, ni siquiera dos veces, de la misma forma. En la actualidad este concepto es aplicable a muchos otros campos y la Ingeniería de Software no es la excepción, puesto que se han utilizado patrones de manera significativa en el diseño y arquitectura del software, y en el desarrollo de procesos (Ampatzoglou, Frantzeskou, & Stamelos, 2012). En este sentido, la Figura 3.9 resume las áreas de aplicabilidad de los patrones dentro de la Ingeniería de Software. Del lado derecho de esta figura se observa el uso de patrones de procesos, los cuales han sido un enfoque útil para aplicar estrategias de gestión del conocimiento dentro de las organizaciones de desarrollo de software. Así, regularmente las tareas relacionadas con la Mejora del Proceso de Software son un tanto complejas y pueden apoyarse en los patrones de procesos para gestionar el conocimiento y reducir la dificultad en la implantación de un modelo de procesos (García, Martín, Urbano, & de Amescua, 2013).

Así pues, es posible agilizar la definición e implantación de un proceso a través de un patrón dados los siguientes beneficios:

- Mejora la comunicación entre los grupos de ingenieros de software.
- Facilita la gestión de los elementos involucrados en el diseño del proceso.
- Facilita el diseño colaborativo.

- Facilita el aprendizaje sobre el uso apropiado de los modelos de procesos.
- Facilita la sistematización de conocimientos y resolución de problemas.
- Facilita la adquisición de conocimientos sobre los modelos de procesos y las técnicas para la resolución de problemas.



**Figura 3.9.** El uso de patrones en las áreas de la ingeniería del software (García, Martín, Urbano, & de Amescua, 2013).

En este sentido, una vez que se ha descrito la importancia del concepto de patrón dentro del metamodelo propuesto, la Figura 3.10 resume las relaciones de los elementos que lo conforman considerando aspectos relevantes para su implementación en el contexto de la tesis. La creación de un patrón tiene como objetivo la reducción de los costos en el arranque de un proyecto desde el punto de vista del tiempo invertido en entrenamiento del proceso y desde la perspectiva de reutilizar aquellos activos que son útiles para un proyecto determinado, utilizando la experiencia previa de otros proyectos. Es decir, el patrón creado genera la definición de un proceso en términos de los activos estándar creados anteriormente (tareas, procesos, productos y medidas) para los proyectos desarrollados. De acuerdo a lo anterior, la Figura 3.10 muestra de forma general la estructura de un patrón para un proyecto específico. Esta estructura indica que un patrón puede tomar de las capas M1 y M0 lo que mejor se adecue al proyecto que se pretende desarrollar. Es decir, el jefe de proyectos debe considerar la metodología o modelo con el que pretende trabajar y cargar a su proyecto toda la información relevante que le permita llevarlo a buen fin. Es evidente que antes de tomar esta decisión, el jefe de proyectos debe determinar qué tan alejado se encuentra de lograr una cobertura respetable sobre el modelo de referencia escogido. Con este objetivo es que en el Nivel M1 se define el concepto de Evaluación, el cual será explicado un poco más adelante.

Así pues, la definición mostrada en la Figura 3.10 debe ser implantada dentro de la empresa para obtener datos sobre la efectividad del metamodelo, y para cubrir la última actividad relacionada con su creación (la prueba). Por lo tanto, es necesario determinar cómo implementar el concepto de una manera práctica y sencilla para el pequeño equipo. En este sentido, de acuerdo con (García, de Amescua, Sánchez, & Bermón, 2011) una Biblioteca de Activos de Proceso (PAL, por sus siglas en inglés) es un repositorio de documentos relacionados con un proceso de software que permite

implementar infraestructuras tecnológicas como soporte a una organización. Además, permite que la experiencia acumulada pueda ser utilizada en proyectos futuros de software. Es así pues que una PAL representa una alternativa sencilla de implementar en el entorno de los pequeños equipos dada su facilidad de uso y capacidad de crecimiento (véase Figura 3.11).

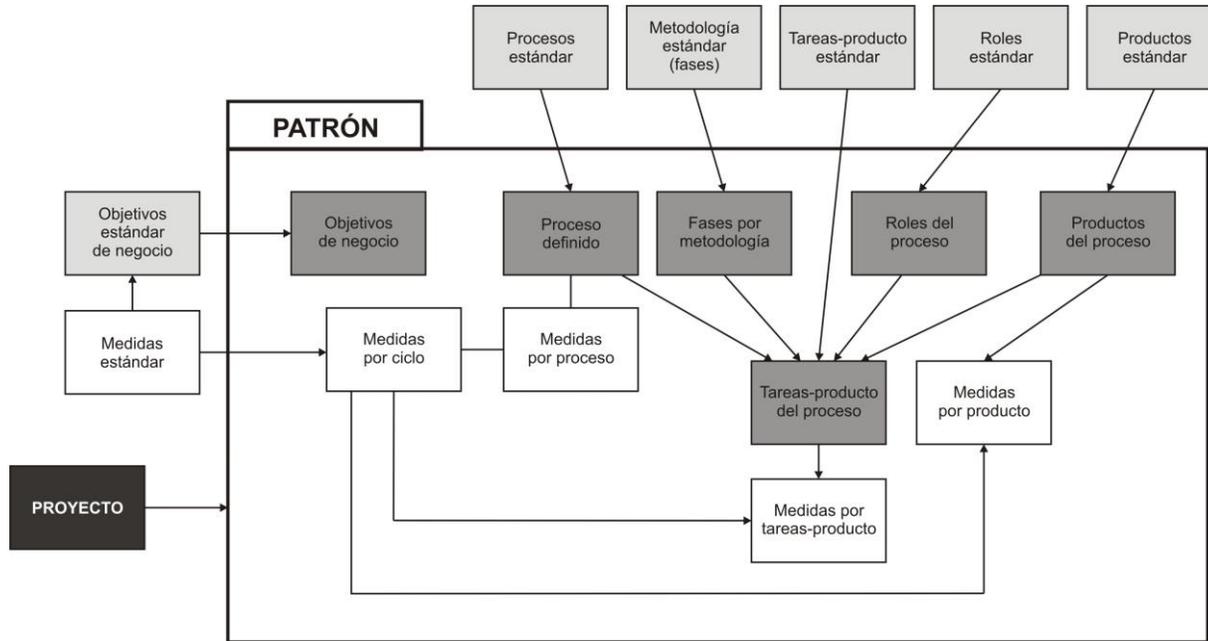


Figura 3.10. Definición del concepto patrón a implementar en el metamodelo.

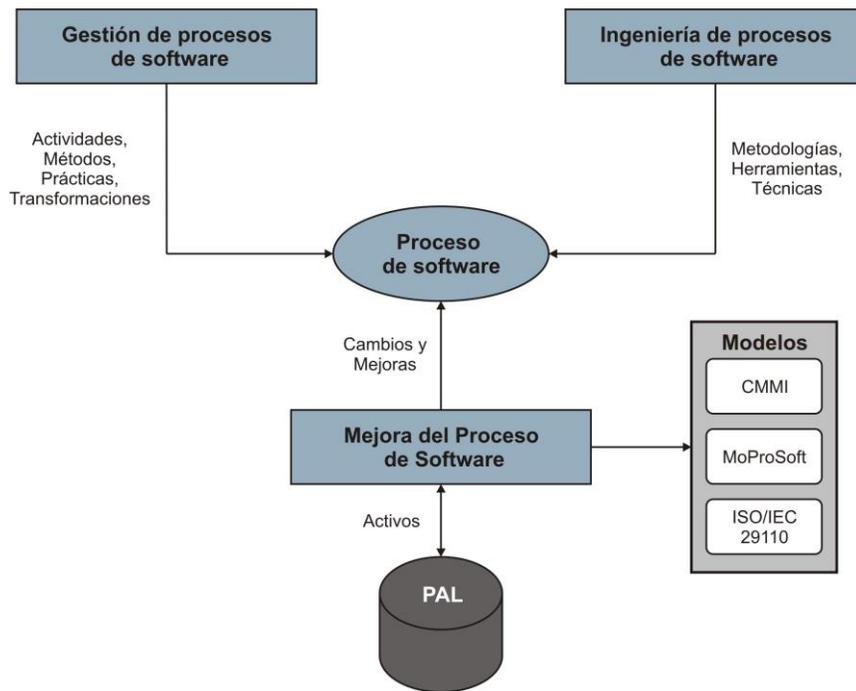


Figura 3.11. Definición conceptual de una PAL (García, de Amescua, Sánchez, & Bermón, 2011).

En este sentido, una PAL es una tecnología de apoyo al proceso de software, puesto que permite la integración de otras tecnologías de producción y gestión de conocimiento, además de brindar apoyo para la integración de las tecnologías de recuperación y distribución de la información

del proceso de la organización. Es decir, en términos prácticos la PAL es un repositorio de búsqueda indexada y organizada de los activos de un proceso, los cuáles son fácilmente accesibles para quien necesite información o guía (i.e., datos, plantillas u otros materiales de soporte) sobre el proceso a ejecutar.

Una PAL entonces proporcionará la base para la estandarización de los procesos definidos a través del metamodelo, de esta forma será posible que todos los proyectos utilicen el mismo proceso o distintas adaptaciones aprobadas por el proceso “estándar” (Calvo-Manzano, García, & Arcilla, 2008). De acuerdo con esta premisa, las PyMES pueden establecer diversos activos de proceso para sus labores cotidianas, activos que incluyen políticas, directrices, definiciones estándar de proceso, modelos de ciclo de vida, guías y plantillas, etc. Así, al desarrollar un proyecto específico podrán utilizar tales activos y adaptarlos a procesos “a medida” de la organización.

### 3.5.1. Incorporación de la evaluación

Como se mencionó anteriormente, el metamodelo define el concepto de evaluación en su capa M1 con el fin de caracterizar el estado inicial del proceso de las PyMES y facilitar la definición de un proceso acorde a sus necesidades. En este sentido, se incorpora un mecanismo de evaluación al metamodelo para permitir que los jefes de proyectos evalúen sus procesos con el fin de identificar las debilidades y fortalezas de los mismos. A través de esta evaluación, el jefe proyectos recibe una ayuda inicial sobre qué actividades debe incorporar a su nuevo proceso, el cual es definido en el nivel de abstracción M2. De esta manera, los jefes de proyectos deben responder, por ejemplo, a una serie de cuestionarios para obtener un nivel de cobertura por cada proceso evaluado, tomando como base el proceso actual y el modelo de procesos que se pretenda seguir. En consecuencia, el jefe de proyectos analiza los resultados de la evaluación dando un peso específico a cada tipo de respuesta e identificado las prácticas que están institucionalizadas dentro de las empresas, y cuáles no se realizan en absoluto (véase Tabla 26).

**Tabla 26.** Clasificación de niveles de rendimiento durante la evaluación del proceso (García, Calvo-Manzano, Cuevas, & San Feliu, 2007).

Posible respuesta	Nivel de rendimiento	Descripción
Siempre	4	La actividad está documentada y establecida en la empresa. Siempre se lleva a cabo, entre el 76% y 100% de las veces, en sus proyectos de software.
Usualmente	3	La actividad está establecida en la empresa, pero rara vez es documentada. Se realiza por lo general, entre el 51% y 75% de las veces, en sus proyectos de software.
Algunas veces	2	La actividad está débilmente establecida en la empresa. Se realiza a veces, entre el 26% y 50% de las veces, en sus proyectos de software.
Rara vez	1	La actividad se realiza con poca frecuencia en la empresa. Rara vez se lleva a cabo, entre el 1% y el 25% de las veces, en sus proyectos de software.
Nunca	0	La actividad no se lleva a cabo en la organización (0%). Ninguna persona o grupo realiza la actividad de la organización.
No lo sé	--	La persona no está segura de cómo responder a la pregunta.
No aplica	--	La pregunta no se aplica a la empresa.

En el contexto de la Mejora del Proceso de Software, suele elegirse una técnica basada en cuestionarios ya que proporciona soluciones rápidas dentro de una metodología de investigación, dado que es el investigador quien determina las preguntas a formular y la gama de respuestas que se pueden dar. Esto hace que el resultado obtenido sea más preciso y fácil de analizar desde el punto de vista del investigador (Young, Fang, & Hu, 2006; Habra, Alexandre, Desharnais, Laporte, &

Renault, 2008; Pino et al., 2010). Así, el mecanismo de evaluación utilizado se basa en dos tipos de actividades: prácticas específicas (relacionadas con los jefes de proyectos) y prácticas genéricas (relacionadas con la alta dirección de la empresa) (García, Calvo-Manzano, Cuevas, & San Feliu, 2007). A modo de ejemplo, la Figura 3.12 muestra la técnica utilizada para identificar fortalezas y debilidades de un proceso.

Administración de Proyectos Específicos	Jefes de proyecto		1					2		
	1	2	#S	#U	#A	#R	#N	Ncp	M	D
1. ¿Se identifican hitos importantes para el proyecto?	S	N	1 50%	0	0	0	1 50%	50%	2	2.00
2. ¿Se identifican suposiciones sobre el calendario del proyecto?	A	R	0	0	1 50%	1 50%	0	37%	1.5	0.50
3. ¿Se identifican las limitaciones del proyecto?	A	A	0	0	2 100%	0	0	50%	2	0
4. ¿Se identifican las dependencias de las tareas del proyecto?	U	N	0	1 50%	0	0	1 50%	37%	1.5	1.50
5. ¿Se definen y mantienen el presupuesto y calendario del proyecto?	S	A	1 50%	0	1 50%	0	0	75%	3	1
6. ¿Se establecen acciones correctivas para el proyecto?	R	A	0	0	1 50%	1 50%	0	37%	1.5	0.50
7. ¿Se identifican los riesgos del proyecto?	A	A	0	0	2 100%	0	0	50%	2	0
8. ¿Se documentan los riesgos?	R	R	0	0	0	2 100%	0	25%	1	0
9. ¿Se evalúan los riesgos cuando es necesario?	R	R	0	0	0	2 100%	0	25%	1	0
10. Do you establish requirements and procedures to ensure privacy and security of the project's data?	U	R	0	1 50%	0	1 50%	0	50%	2	1
11. ¿Se establecen mecanismos para archivar los datos y acceder a los datos archivados?	U	U	0	2 100%	0	0	0	75%	3	0
12. ¿Se determinan los datos del proyecto que serán identificados, recogidos y distribuidos?	U	U	0	2 100%	0	0	0	75%	3	0
13. ¿Se determinan los requisitos de instalaciones y equipo?	R	R	0	0	0	2 100%	0	25%	1	0
14. ¿Se determinan los requisitos de personal?	R	R	0	0	0	2 100%	0	25%	1	0
15. ¿Se identifican el conocimiento y las habilidades necesarias para desarrollar el proyecto?	U	U	0	2 100%	0	0	0	75%	3	0
<b>TOTALES</b>			2 6.7%	10 33.3%	6 20%	12 40%	0	47%		

Figura 3.12. Ejemplo de análisis de respuestas realizado por los jefes de proyectos.

La técnica utiliza un promedio ponderado para calcular el nivel de cobertura para cada pregunta (número 1 en la Figura 3.12). Por lo tanto, el análisis se traslada de un punto de vista cualitativo (gama de respuestas Siempre (S), Usualmente (U), Algunas veces (A), Rara vez (R), y Nunca (N)) a una visión cuantitativa (nivel de cobertura por pregunta) (número 2 en la Figura 3.12).

Además, la desviación estándar también es considerada mediante el cálculo del nivel de cobertura para cada proceso (número 3 en la Figura 3.12).

El porcentaje de nivel de cobertura para cada pregunta (Ncp) se calcula considerando las respuestas de todos los jefes de proyecto de la siguiente manera:

$$Ncp_{ij} = Cr_{iS} * 1 + Cr_{iU} * 0.75 + Cr_{iA} * 0.50 + Cr_{iR} * 0.25 + Cr_{iN} * 0 \quad (1)$$

donde  $Cr$  es la cobertura de la respuesta,  $i$  es el número de pregunta,  $j$  es el número de jefes de proyectos, y los números representan los porcentajes definidos en la Tabla 26 para cada tipo de respuesta. La razón de esta ponderación es porque se considera que las afirmaciones más fuertes deben tener más importancia que las débiles, dado que, por ejemplo, algo que siempre se hace no puede influir de la misma manera que algo que se hace raramente. Sin esta ponderación todas las preguntas tendrían siempre una cobertura del 100%, lo cual no aclararía nada.

Además, para cada pregunta, la media matemática y la desviación estándar son calculadas de la siguiente manera:

$$M_i = \frac{\#A * 4 + \#U * 3 + \#S * 2 + \#R * 1 + \#N * 0}{jp} \quad (2)$$

$$Desv_i = \sqrt{\frac{[\#jp * (\#A * 4^2 + \#U * 3^2 + \#S * 2^2 + \#R * 1^2) - \#jp^2 * M_i^2]}{jp^2}} \quad (3)$$

donde  $i$  es el número de pregunta,  $jp$  es el número de jefes de proyectos que participan en la evaluación de los procesos y las letras con # son los niveles de desempeño definidos en la Tabla 26. Los valores de 1, 0.75, 0.5, 0.25 y 0 también podrían ser utilizados y el valor máximo de la media matemática en lugar de 4 sería 1. Sin embargo, es recomendable utilizar valores más altos de ponderación (4, 3, 2, 1, y 0) mediante la aplicación de un factor, por lo tanto las diferencias existentes son más visibles. Es importante tener en cuenta que existe una correspondencia biunívoca o el mismo significado entre el porcentaje de cobertura de la pregunta y su media matemática (la única diferencia es que los factores de escala son diferentes), de tal forma que un alto porcentaje de cobertura también reflejará una media matemática alta y en la misma proporción que el porcentaje.

Por último, el nivel de cobertura total (NCT) del proceso se calcula de la siguiente manera:

$$NCT(p) = \frac{\sum_{i=1}^n Cp_i}{n} \quad (4)$$

donde  $p$  es el proceso evaluado,  $i$  es el número de pregunta, y  $n$  es el total de preguntas del cuestionario. De esta manera, los jefes de proyectos determinan el nivel de cobertura de su empresa y el promedio de cobertura para identificar las fortalezas y debilidades de los procesos seleccionados. En este sentido, para analizar las fortalezas y debilidades (o problemas), los jefes de proyectos tienen que usar las respuestas obtenidas en las evaluaciones. Aquellas preguntas con una cobertura inferior a cierto porcentaje (número 4 en la Figura 3.12) implican que la actividad del modelo de referencia correspondiente (número 5 en la Figura 3.12) no está bien implementada en la empresa y tiene que ser considerada como una cuestión a mejorar (o una debilidad) en el proceso. Este porcentaje de cobertura varía de acuerdo al modelo de procesos que se pretenda adoptar, por ejemplo para obtener un nivel de madurez del CMMI, el órgano certificador solicita una cobertura mínima del 75%, en caso de MoProSoft este valor varía a 85%. Por otro lado, las preguntas con

cobertura igual o mayor al porcentaje marcado por el órgano certificador deben ser analizadas por los jefes de proyectos a través de la desviación estándar de la siguiente manera:

- Si la desviación estándar es inferior a 0.8, la actividad del modelo de referencia relacionada con la pregunta representa una fortaleza del proceso evaluado (número 6 en la Figura 3.12).
- Si la desviación estándar es igual o superior a 0.8 (número 7 en la Figura 3.12), la actividad del modelo de referencia relacionada con esa pregunta debe ser profundamente explorada por los jefes de proyectos a través de entrevistas para aclarar si se trata de una fortaleza o de un aspecto a mejorar. Este caso significa que ha habido fuertes discrepancias en las respuestas dadas por la empresa al completar los cuestionarios (las respuestas de los jefes de proyectos son muy divergentes y, por lo tanto, deben ser exploradas a mayor profundidad).

Así, la Figura 3.12 resume el proceso que los jefes de proyectos tienen que realizar para tomar una decisión y definir un proceso que tome en cuenta todas las deficiencias detectadas. Adicionalmente, los resultados obtenidos en las empresas pueden mostrarse con gráficos de barras, que resumen el porcentaje de prácticas efectivas que es cubierto por los procesos evaluados, y gráficas de Kiviat, que muestran el nivel de madurez/capacidad de todos los procesos evaluados en la empresa.

### 3.5.2. Implementación del concepto de patrón a través de la PAL

Una vez definidos los elementos que conforman la arquitectura es necesario implementar el Nivel M3 del modelo conceptual, el repositorio de meta-elementos. Es evidente que no es posible entregar únicamente el metamodelo a la PyMES porque en lugar de facilitarle el trabajo se le estaría complicando la labor de gestionar. En este sentido, se introduce el concepto de patrón a través de un modelo de datos que considera, como punto de partida, el metamodelo presentado en la Figura 3.8.

En el contexto de los negocios, un modelo de datos está representado por una estructura abstracta que documenta y organiza la información necesaria para definir la comunicación entre el personal del departamento técnico y el resto de empleados. En el campo de la Ingeniería de Software esta definición difiere en cuanto a su enfoque, el cual se centra en el planeamiento del desarrollo de aplicaciones y la decisión de cómo se almacenarán los datos y cómo se accederá a ellos. Concretamente un modelo de datos se utiliza para describir las entidades fundamentales en el diseño y las relaciones entre ellas (Pressman & Maxim, 2014). Es decir, regularmente un modelo de datos es la técnica normal que se utiliza para describir las estructuras de las bases de datos.

Así pues, la técnica de modelado de datos más utilizada es la de Entidad-Relación-Atributo (ERA) (Johnson, Rosebrugh, & Wood, 2002), que muestra las entidades de datos, sus atributos asociados y las relaciones entre estas entidades. Sin embargo, como todos los modelos gráficos, los modelos ERA carecen de detalle y deben complementarse con descripciones más detalladas de las entidades, las relaciones y los atributos que se incluyen en el modelo. Estas descripciones detalladas se pueden recolectar en un repositorio o diccionario de datos. De forma general, los diccionarios de datos son útiles cuando se desarrollan modelos del sistema y se utilizan para administrar toda la información proveniente de todos los modelos. De acuerdo con (Somerville, 2015), un diccionario de datos es, simplemente, una lista alfabética de los nombres que se incluyen en los diferentes modelos de un sistema. Además del nombre, el diccionario debe incluir una descripción asociada de la entidad nombrada y, si el nombre representa un objeto compuesto, tendrá que existir una descripción de la composición. En este sentido, la implementación de la PAL requiere de una transformación del metamodelo a su correspondiente modelo de datos, para posteriormente incorporar el concepto de patrón. Cabe destacar que para esto es necesario hacer una propuesta de transformación puesto que no existen reglas universales que sean aplicadas a este

proceso. Por lo tanto, es necesario transformar, en primera instancia, cada una de las clases en entidades, es decir, una clase es “mapeada” a un tipo entidad en el modelo ERA y los atributos de la clase son mapeados a atributos de las entidades resultantes. Para el caso de las clases con herencia (como las clases **recurso** y **guía**) es necesario crear una tabla para cada subclase con los atributos comunes que figuran en ellas. Además, cada tabla posee una clave primaria y para las subclases **material**, **instalación**, **equipo**, **externo** y **MiembroEquipo** se creó un atributo en sus clases correspondientes llamado *tipo*. Por otro lado, la clase **guía** es mapeada a una entidad con sus atributos correspondientes y se crea un atributo adicional llamado *tipo* para indicar si se trata de una **herramienta**, **técnica** o **activo**. La transformación de las relaciones “uno a muchos” en la conversión del diagrama de clases a ERA requiere agregar una columna común a cada una de las entidades relacionadas. Dicha columna suele ser la llave primaria de una de las tablas, quedando así la llave primaria del lado “uno” y la tabla del lado “muchos”. La llave agregada del lado “muchos” no forma parte de la llave primaria, solamente es una columna común para ambas tablas. En el caso de que la asociación sea “uno a uno”, ésta se trata como si fuera un caso particular de la relación “uno a muchos” y la clave se propaga en las dos direcciones. Así, es necesario analizar la situación para determinar en dónde se colocará la clave, con la restricción de evitar valores nulos para el campo de la llave foránea. Ahora bien en el caso de las asociaciones “muchos a muchos”, es necesario crear una entidad adicional a las relaciones de las clases asociadas, la cual contendrá las claves primarias de ambas tablas que formarán la clave primaria de la nueva entidad. Al hacer este análisis se revisa si es conveniente que se agreguen atributos. En el caso de la asociación reflexiva que existe en **actividad**, es necesario crear una entidad para la clase **dependencia** con los atributos de la clave primaria de **actividad** y otro con el nombre dado al vínculo en la asociación reflexiva, y la cardinalidad de la relación se detalla colocando patas de gallo en las líneas que conectan a las entidades en el lado muchos de la relación (véase Figura 3.13).

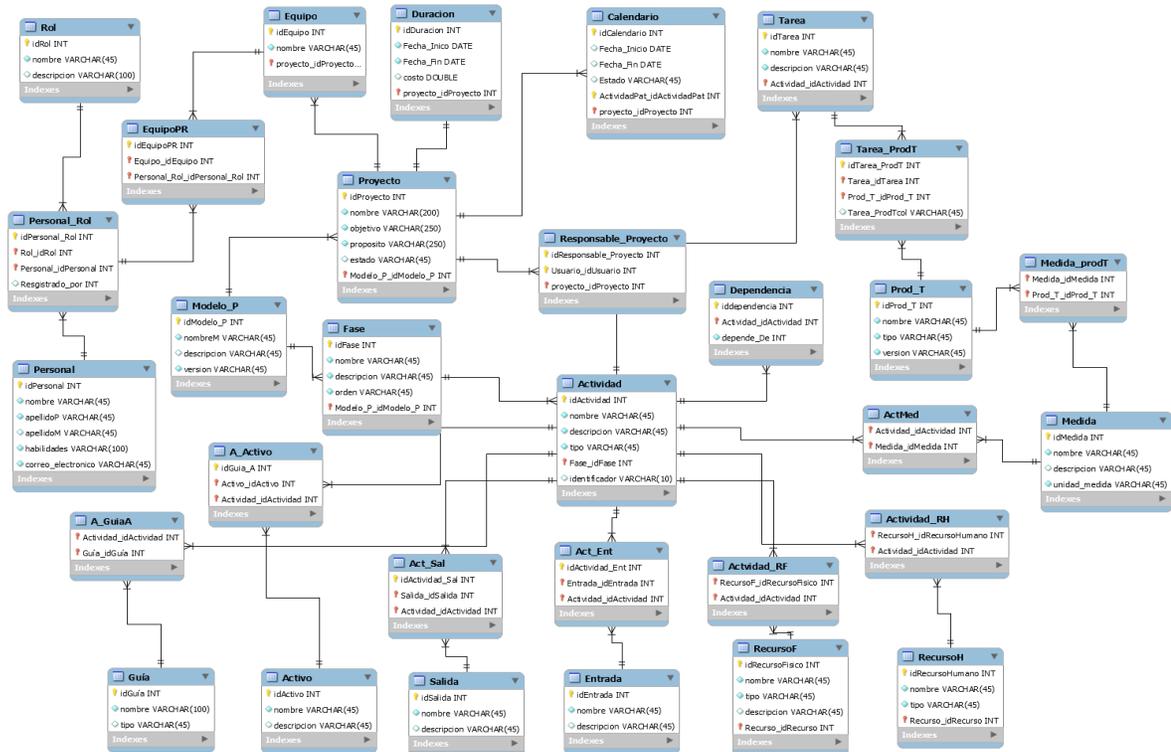


Figura 3.13. Modelo de datos obtenido a partir del metamodelo.

Cabe resaltar que es necesario hacer una revisión del modelo y ajustarlo para que la información sea coherente de tal manera que sea posible llevar, mediante el refinamiento, el metamodelo hacia la implementación. Considerando esto y tomando como referencia el modelo de datos de la Figura 3.13, la Figura 3.14 muestra al modelo de datos propuesto para implantar el concepto “patrón” del metamodelo en una herramienta computacional orientada a las PyMES. Así, se requirió de la agregación de tablas “patrón” (e.g., modelo\_p\_pat, proceso\_pat, fases\_pat, actividades\_pat, etc.,) con el objetivo de que éstas almacenaran la información de las componentes estándar ya definidas y que el jefe de proyectos utilizara. Sin embargo, esto crea redundancia en la base de datos dado el siguiente problema: las actividades, entradas, salidas, activos, guías están contenidas en tablas “estándar” que se relacionan puesto que pertenecen al mismo modelo de procesos, lo cual genera redundancia con las tablas “patron” que almacenarían a su vez información similar a la de las tablas “estándar”, y que también están relacionadas entre sí y con el mismo modelo de procesos. Además de que se incrementa considerablemente el número de tablas del modelo. En este sentido, la resolución de este problema requirió la normalización de la base de datos mediante la agregación de tablas intermedias con relaciones de uno a muchos (por ejemplo, la tabla intermedia modelo\_proceso para las tablas proceso y modelo\_p). De manera similar, se agregaron la tabla intermedia “estándar” entre modelo\_proceso y fase, y la tabla “patrón” para que cada nuevo proyecto sea asignado a un patrón que será almacenado en la tabla de proyecto\_pat para relacionar así la tabla “patrón” con la tabla “estándar”. De esta manera se pueden crear las relaciones de los elementos de un proceso estándar y también se puede generar un “patrón” sin sacrificar las relaciones de integridad entre las tablas del modelo de datos.

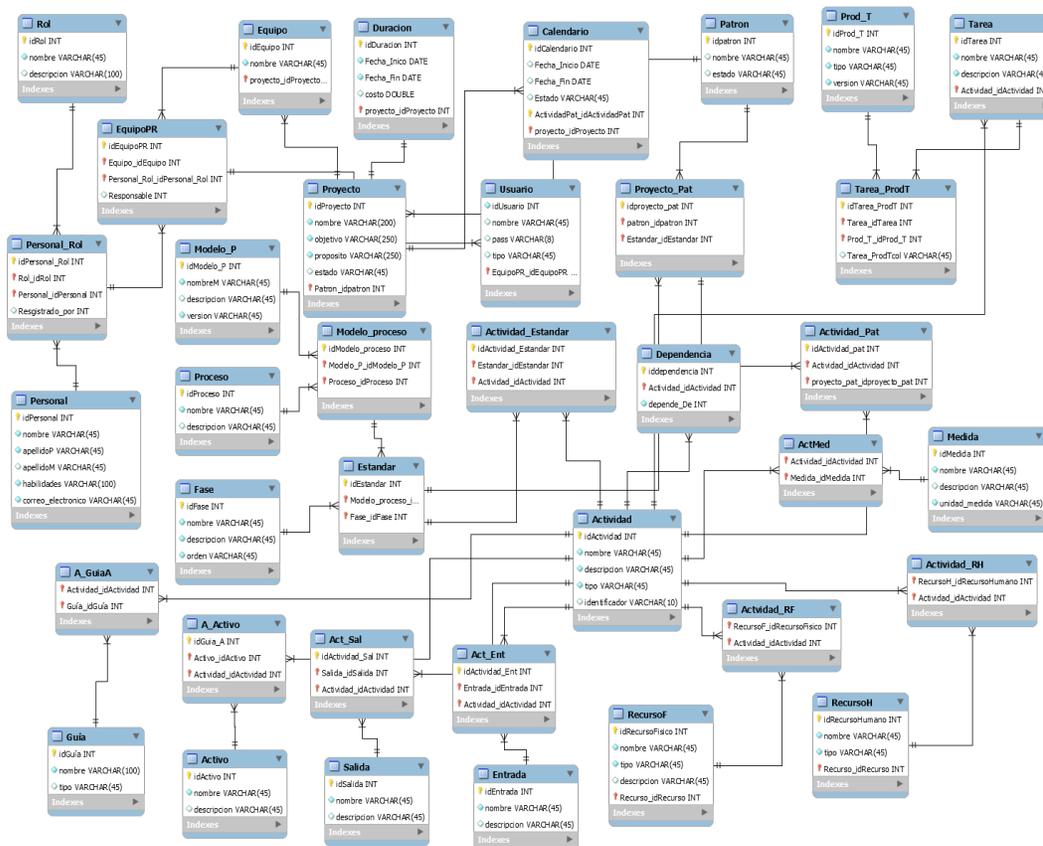


Figura 3.14. Modelo de datos para la implementación del concepto de patrón.

Finalmente, una vez que se han definido tanto el metamodelo (en su forma conceptual) como el modelo de datos que conduce a la construcción de la PAL que permita probar dicho metamodelo en un entorno real, a continuación se procede a establecer un caso de estudio que permita demostrar la hipótesis establecida al inicio de esta tesis.



## **4. Resultados obtenidos**

### **4.1. Contexto de la validación**

Desde hace poco menos de tres décadas se ha aplicado el proceso experimental para estudiar una diversidad de áreas relacionadas con la tecnología del software. En este sentido, la experimentación ha probado ser efectiva al mejorar el entendimiento de la Ingeniería de Software y promover el conocimiento sobre los procesos y productos de software. De acuerdo con Juristo y Moreno (2013), la experimentación dentro de la Ingeniería de Software pretende demostrar con hechos las suposiciones, hipótesis, especulaciones y creencias que abundan alrededor de la construcción y uso del software. Así, la experimentación dentro de la Ingeniería de Software permite, en palabras de Pfleeger (2005), aumentar la comprensión de lo que hace bueno al software y cómo hacerlo bien. Es decir, hacer del desarrollo de software una actividad científicamente predecible, gracias al conocimiento de las relaciones entre los procesos de producción y los productos que se obtienen con éstos (Hannay, Sjoberg, & Dyba, 2007).

En este sentido, las estrategias empíricas experimentales dentro de la Ingeniería de Software incluyen a las encuestas, casos de estudio y experimentos (Wohlin, Runeson, Höst, Ohlsson, Regnell, & Wesslén, 2012). Por ejemplo, a través de un caso de estudio el investigador puede recopilar información detallada sobre un proyecto durante un período de tiempo. Además, durante el desarrollo de un caso de estudio es posible aplicar una variedad de procedimientos para recopilar los datos. De acuerdo con Runeson y Höst (2009), un ejemplo de éstos puede ser utilizar un proyecto piloto para evaluar los efectos de un cambio con respecto a una línea base. Así pues, los casos de estudio son adecuados para la evaluación industrial de métodos y herramientas de la Ingeniería de Software porque pueden evitar problemas de escalado. En este contexto, una ventaja de los casos de estudio es que son más fáciles de planificar, pero su principal desventaja es que los resultados que se obtienen son difíciles de generalizar y más difíciles de interpretar, es decir, es posible mostrar los efectos en una situación típica, pero tales resultados no permiten la generalización a cualquier situación (Yin, 2013).

Considerando estos argumentos se ha escogido un caso de estudio como la estrategia empírica experimental para aceptar, o no, la hipótesis establecida en el Capítulo 1 de esta tesis. Por lo tanto, de acuerdo con las recomendaciones de Wohlin et al. (2012) existen tres diferentes estrategias para desarrollar un caso de estudio:

1. Comparar los resultados obtenidos entre una nueva propuesta y un proyecto anterior (línea base).
2. Desarrollar dos proyectos en paralelo (i.e., proyectos hermanos), eligiendo uno de ellos como línea base, y

3. Aplicar la nueva propuesta sobre algunos componentes seleccionados y comparar los resultados que se obtengan con los de los componentes en los cuales no se aplicó.

Para este capítulo de experimentación se presenta la aplicación de la primera estrategia: “Comparar los resultados obtenidos entre una nueva propuesta y una línea base”, para la cual se considerarán como línea base los resultados obtenidos de la aplicación de los cuestionarios de evaluación diseñados para cubrir el Nivel M1 del modelo conceptual. En este sentido, es importante mencionar que dichos cuestionarios serán definidos posteriormente. De manera similar, se utilizarán los datos históricos de proyectos anteriores de la PyMES participante en el estudio. Por lo tanto, como paso inicial para el establecimiento de la línea base, los jefes de proyecto de la empresa deben responder a los cuestionarios antes de utilizar la herramienta computacional, la cual fue creada con el objetivo de facilitar la definición de un proceso de gestión en la empresa. Posteriormente, esta información será comparada con una segunda evaluación que se realizará después de haber utilizado el enfoque presentado en esta tesis.

## 4.2. Definición del proyecto línea base del caso de estudio

El caso de estudio se enfoca en la aplicación de la solución propuesta en esta tesis en una PyMES mexicana desarrolladora de software. Esta empresa, que en adelante será denominada PES1 por razones de confidencialidad, fue legalmente constituida en el 2007 en el Estado de Oaxaca. La Tabla 27 resume las principales características de esta PyMES.

**Tabla 27.** Número de empleados y actividad principal de la empresa incluida en el caso de estudio.

Empresa	Número de empleados	Actividad principal
PES1	20	Desarrollo de software a la medida y diseño de páginas Web

La estructura administrativa de la PyMES, dado su tamaño, se compone principalmente de tres departamentos:

- Dirección administrativa. Se encarga de la dirección de la empresa a nivel administrativo. Este departamento es controlado por un director relacionado con el área de Administración de Empresas, quien establece los lineamientos organizacionales a seguir.
- Dirección técnica. Se encarga de la dirección de la empresa a nivel técnico. A través de ésta, la empresa desarrolla los proyectos de software solicitados por los clientes. La dirección técnica gestiona y controla todos los aspectos técnicos que el personal requiere para realizar y concluir exitosamente todos los proyectos.
- Departamento contable. Se encarga de realizar todas las gestiones de índole contable con los clientes y con los mismos trabajadores. Este departamento es controlado por la misma dirección administrativa.

En relación a la infraestructura de la PyMES, la Tabla 28 resume la infraestructura con la que cuentan la PyMES.

**Tabla 28.** Infraestructura de la empresa incluida en el caso de estudio.

Infraestructura	Cantidad
Servidor de gama media dedicado a datos y aplicaciones	1
Computadoras personales	21
Computadoras portátiles	4

Impresora láser	2
Impresora de inyección de tinta	2
Impresora matricial	2
Página web de la empresa	1
Islas de trabajo	6
Cañón (proyectores de video)	2
Licencias de software contable	1
Licencias de software (lenguajes de programación)	3
Licencias de software (ofimática)	1
Software propio (enfocado a gestión de documentos y personal)	1
Producto actualmente en el mercado	1

En relación a su experiencia previa, la PyMES afirmó lo siguiente:

- La empresa se ha enfocado durante poco más de 7 años al desarrollo de software a la medida con el objetivo de proporcionar soluciones tecnológicas (en su mayoría de tipo contable) al mercado estatal. Sin embargo, es demasiado frecuente que los problemas en la gestión de sus proyectos les impidan tomar y concluir exitosamente una mayor cantidad de los mismos.
- La empresa tiene conocimiento limitado sobre los modelos de procesos y sobre algunas prácticas relacionadas con la gestión de los proyectos. Sin embargo éstas no están institucionalizadas en la organización y regularmente se realizan a través de héroes que laboran como parte del personal de software.
- Además de intentar seguir los procesos definidos por MoProSoft, la empresa ha intentado incorporar actividades recomendadas por las metodologías ágiles sin éxito alguno. Los principales argumentos vertidos en relación a este resultado consideran que hace falta experiencia para entender y aplicar dichas actividades en el trabajo cotidiano.

Con la intención de recoger información que permitiera aceptar, o no, la hipótesis planteada al inicio de esta tesis, se escogió, de entre los datos históricos de la empresa, repetir el desarrollo de un proyecto denominado “PV/2013 - Control de Sucursales” por ser el que estaba más y mejor documentado y que por ende se utilizó en este caso de estudio como línea base. Es decir, siguiendo la estrategia definida por Wohlin et al. (2012), se pretende comparar los resultados obtenidos a través de la implantación del metamodelo propuesto con los resultados obtenidos para ese proyecto, los cuales representan la línea base. En este sentido, la Tabla 29 resume en tres categorías las principales funcionalidades que fueron cubiertas por dicho proyecto.

**Tabla 29.** Resumen de funcionalidades cubiertas por el proyecto escogido.

<b>Funcionalidades del proyecto PV/2013 – Control de sucursales</b>		
<b>Categoría 1: Control de ventas, reembolsos y gestión de efectivo</b>	<b>Categoría 2: Gestión de datos maestros</b>	<b>Categoría 3: Generación de reportes y gráficos</b>
Permitir la búsqueda de productos	Gestionar productos, categorías de productos y precios	Gestionar múltiples parámetros de selección para cada reporte
Permitir la gestión de impuestos	Calcular impuestos	Generar vista de reportes y gráficas
Integrar lector de código de barras (ventas)	Gestionar almacén	Exportar gráficas a formatos HTML, PDF, Excel, XML y CSV
Integrar cálculo para productos según peso	Gestionar usuarios y roles	Generar reportes de incidencias
Imprimir recibos por compra	Gestionar recursos	Generar reportes de seguridad (acceso a los datos por rol)

Generar resumen de reportes sobre operaciones diarias, venta de productos, y facturación	Gestionar almacenes	Generar reporte diario de acciones en el sistema (por rol)
Establecer seguridad basada en roles	Gestionar las características de cada producto	Generar reporte diario, semanal y mensual de facturas emitidas
Realizar ventas por cliente y corporativos	Integrar lector de código de barras (inventario)	Generar reporte de acciones sensibles a las operaciones de venta
Generar gráficos de productividad por sucursal	Generar reportes por stock de productos, almacenes, e inventario en general	Generar reportes semanales y mensuales de productos más vendidos
	Generar recibos por compra, cambio y/o devolución	Generar reporte de stock

De igual manera se definen las variables que se deben tomar en cuenta en este caso de estudio: el tiempo destinado a la definición de un proceso, tiempo destinado a la implantación de un proceso definido, la desviación de calendario, y la desviación del esfuerzo; dichas variables fueron definidas de acuerdo a sus políticas de trabajo y al proyecto línea base (datos históricos) tal como se muestra en la Tabla 30.

**Tabla 3030.** Variable y valores objetivos del proyecto línea base.

Objetivos	Valor objetivo
Reducir tiempo para definir un proceso	< a 4.5 semanas
Reducir tiempo para implantar un proceso	< a 17.5 semanas
Reducir la desviación del calendario	< a 17%
Reducir el esfuerzo del proyecto	< a 23%

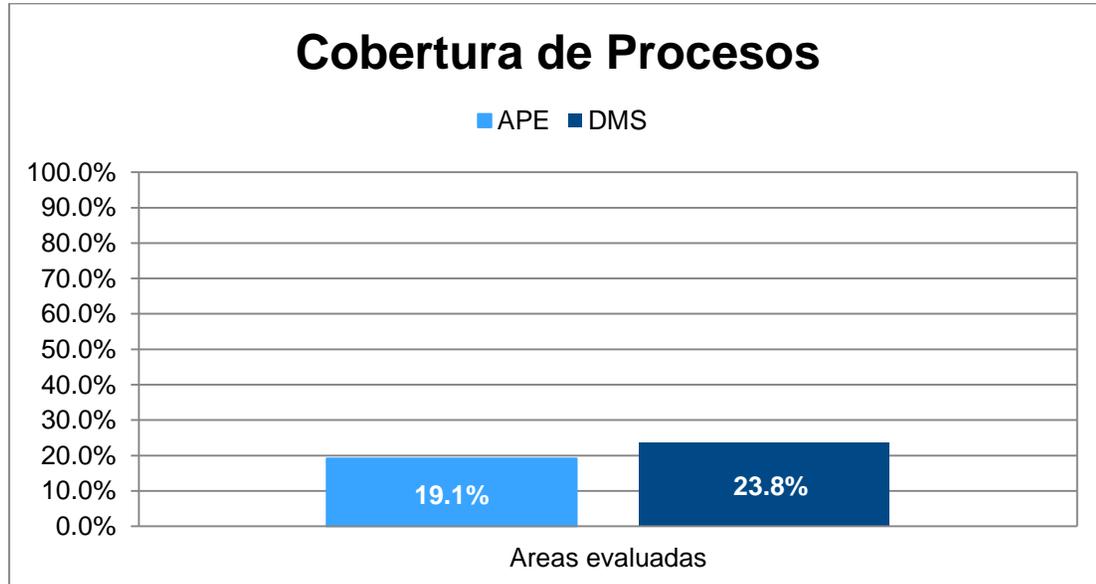
Una vez que las características del caso de estudio y del proyecto a realizar han sido definidas, las siguientes secciones de este capítulo pretenden mostrar los resultados de la cobertura inicial requerida por el Nivel M1 del metamodelo (i.e., y que son obtenidos a través de la aplicación de los cuestionarios de evaluación), los resultados de la implantación del metamodelo para la definición de un proceso de gestión, los resultados de la nueva línea base (obtenidos a través de la aplicación del cuestionario de evaluación después de implantar y utilizar el proceso definido a través del metamodelo), la comparativa entre los resultados y los valores de la línea base, y el análisis detallado sobre los resultados de la mejora.

#### 4.2.1. Identificación de la cobertura de los procesos

El caso de estudio se inició definiendo el alcance relacionado con los procesos que serían incluidos en la experimentación. En este sentido y de acuerdo a los objetivos de esta tesis, se acordó con la PyMES incluir los procesos básicos para la gestión de los proyectos (presentados en el Capítulo 2 de esta tesis), y que se pueden resumir como: Planificación del Proyecto, y Monitorización y Control del Proyecto. Dado el interés de la PyMES por adoptar las actividades recomendadas por MoProSoft, se decidió seguir la definición del proceso de “Administración de Proyectos Específicos” (APE) y “Desarrollo y Mantenimiento de Software” (DMS) puesto que incluyen a los dos procesos anteriores. Dentro de la experimentación el director de tesis y la tesista participaron como el equipo de consultoría que proporcionó la guía en el proceso de evaluación e implantación del metamodelo en el entorno de la empresa participante. Así pues, al inicio se formó un equipo de evaluación compuesto por el equipo consultor y un jefe de proyectos de la PyMES, con el objetivo de evaluar el desempeño del personal en relación a las áreas de proceso anteriores. Este jefe de proyectos tiene una amplia experiencia en la gestión y conoce la cultura de la empresa y la forma en que los proyectos son realizados dentro de la misma.

El personal (técnico y directivo) respondió a los cuestionarios presentados en el Anexo B: Cuestionarios de Evaluación. Estos cuestionarios se basan generalmente en las actividades de un modelo de procesos (o referencia), MoProSoft en el caso particular de este caso de estudio. Es importante mencionar que estos cuestionarios fueron diseñados exclusivamente para este caso de estudio puesto que, como se mencionó anteriormente, la PyMES decidió intentar definir un proceso de gestión siguiendo las recomendaciones de los Niveles 1 y 2 del modelo. Así, la aplicación de los cuestionarios proporcionó información útil relacionada con el estado actual de los procesos e indicó aquellas prácticas que requerían de atención inmediata. Siguiendo el proceso descrito en la sección 3.5.1 fue posible identificar a personas con conocimientos básicos de estimación y formulación de planes quienes fueron invitados a participar en la definición del nuevo proceso. Las respuestas a los cuestionarios reflejan el nivel de realización de las prácticas de MoProSoft para los procesos mencionados; estos resultados, junto con la realización de entrevistas y revisión de documentación objetiva de los procesos, permitieron determinar qué actividades habían sido cubiertas por el personal y cuáles habían sido extendidas en toda la empresa como un proceso institucionalizado.

Así pues, la Figura 4.1 muestra el nivel de cobertura obtenido por la PyMES para los procesos de APE y DMS.



**Figura 4.1.** Resultados de la evaluación sobre los procesos de PES1.

Esta gráfica de nivel de cobertura resume el porcentaje de actividades que se realizaba por cada proceso dentro de la PyMES. La imagen muestra, por ejemplo, que en esta evaluación inicial se obtuvo un 19.1% de cobertura promedio para APE. Los resultados obtenidos en el caso de DMS evidenciaron que la cobertura estaba en 23.8%. Al final de la evaluación, la información obtenida fue cotejada con la documentación de cada empresa y se realizó un mapeo de los resultados con entrevistas como parte de la evidencia objetiva. Entre otras cosas, se determinó que estas calificaciones bajas pudieron deberse a:

- La mayoría de los problemas con las fases iniciales de ambos procesos se relacionaron con el personal de logística quien establece el contacto inicial con el cliente.
- Los procedimientos de la PyMES no estaban documentados (i.e., con plantillas, guías, etc.), y en algunos casos la documentación existente era ambigua. De hecho, existían intentos de formular planes pero éstos no eran controlados.

- Los proyectos no eran delimitados y se terminaba por rebasar el tiempo de entrega (calendario) y el costo planificado.
- Existieron problemas de comunicación entre los programadores para realizar la integración del producto.
- Se carecía de capacitación específica en todos los roles (e.g., cursos de capacitación).
- No existían fichas de personal que permitieran la distribución eficiente de personal de acuerdo al tipo de proyecto.
- No existía una dirección técnica que controlara el desarrollo de los proyectos.

Así pues, los resultados obtenidos evidenciaron que los procesos evaluados se encontraban pobremente implementados. En este sentido, la Tabla 31 muestra los valores definidos por el Organismo de Normalización y Certificación (NYCE) para alcanzar una certificación. En base a estos resultados y considerando que no es recomendable incluir más de tres áreas de proceso en una iniciativa de mejora, se decidió enfocar los esfuerzos de mejora en definir, a través del metamodelo, procesos adecuados para la empresa.

**Tabla 311.** Escala normativa de calificación definida por el Organismo Certificador.

Calificación	Descripción	Porcentaje de cobertura
N	No alcanzado	0% al 15%
P	Parcialmente alcanzado	> 15% al 50%
A	Ampliamente alcanzado	> 50% al 85%
C	Completamente alcanzado	> 85%

#### 4.2.2. Implantación del metamodelo en el proyecto piloto

Después de aplicar la evaluación definida en el Nivel M1 del modelo conceptual, la implantación del Nivel M2 se inicia con la definición del nuevo proceso que será utilizado en el proyecto piloto a desarrollar dentro de la PyMES. Para esto, los jefes de proyecto fueron asesorados en la definición del nuevo proceso considerando los activos previamente definidos en la PAL. En este sentido, dado el interés de la empresa por seguir las prácticas de MoProSoft fue necesario diseñar todos los artefactos necesarios (i.e., activos del proceso en forma de plantillas) para ejecutar las actividades del modelo de procesos en su Nivel 2 (particularmente para las áreas de APE y DMS) y fortalecer el proceso actual de gestión de la empresa. Dichos activos se resumen en el Anexo C: Activos para los procesos APE y DMS.

Cabe recordar que la definición del proceso de la empresa se realiza a partir de los elementos estándar que se encuentran almacenados en la PAL. En el caso particular de este caso de estudio se crearon todos los artefactos utilizados por MoProSoft y debería hacerse lo mismo para cada modelo que cualquier empresa intente implantar. De esta forma, las fases, actividades, productos de trabajo, medidas y demás elementos estándar seleccionadas por los jefes de proyectos desde la PAL fueron incorporadas al patrón **Patrón PV/2013** (véase Figura 4.2), creado específicamente para el desarrollo del proyecto piloto (véase Figura 4.3). Finalmente, la Figura 4.4 muestra el diagrama de objetos que representa el funcionamiento de la herramienta para este caso particular.

Bienvenido jp

### Asignar nombre a patrón del proyecto “PV/2013 - Control de Sucursales”

Nombre del patrón  
patron PV/2013

AGREGAR

O selecciona uno de la lista

Seleccionar patron

**Figura 4.2.** Pantalla de la creación del patrón Patrón PV/2013 para realizar el proyecto piloto.

De esta forma, el caso de estudio se centró en realizar nuevamente un proyecto (denominado línea base) que era el mejor documentado en la base de datos históricos de la PyMES. Esta decisión se sustentó en las recomendaciones de Juristo y Moreno (2013), en el sentido de realizar el mismo proyecto a partir de cero con el objetivo de incrementar la probabilidad de deducir correctamente los resultados arrojados por el caso de estudio. Así pues, esta estrategia aumenta la validez y confiabilidad de los resultados observados en el caso de estudio (Juristo & Gómez, 2012).

Bienvenido jp

Modelos Fases Actividades Entradas y Salidas Recursos Tareas Productos de trabajo Fin

### Generar patrón ► patron PV/2013

Selecciona un modelo  
MoProSoft V 1.3

REGRESAR CONTINUAR

**Figura 4.3.** Pantalla de la asociación del patrón creado con el proyecto piloto.

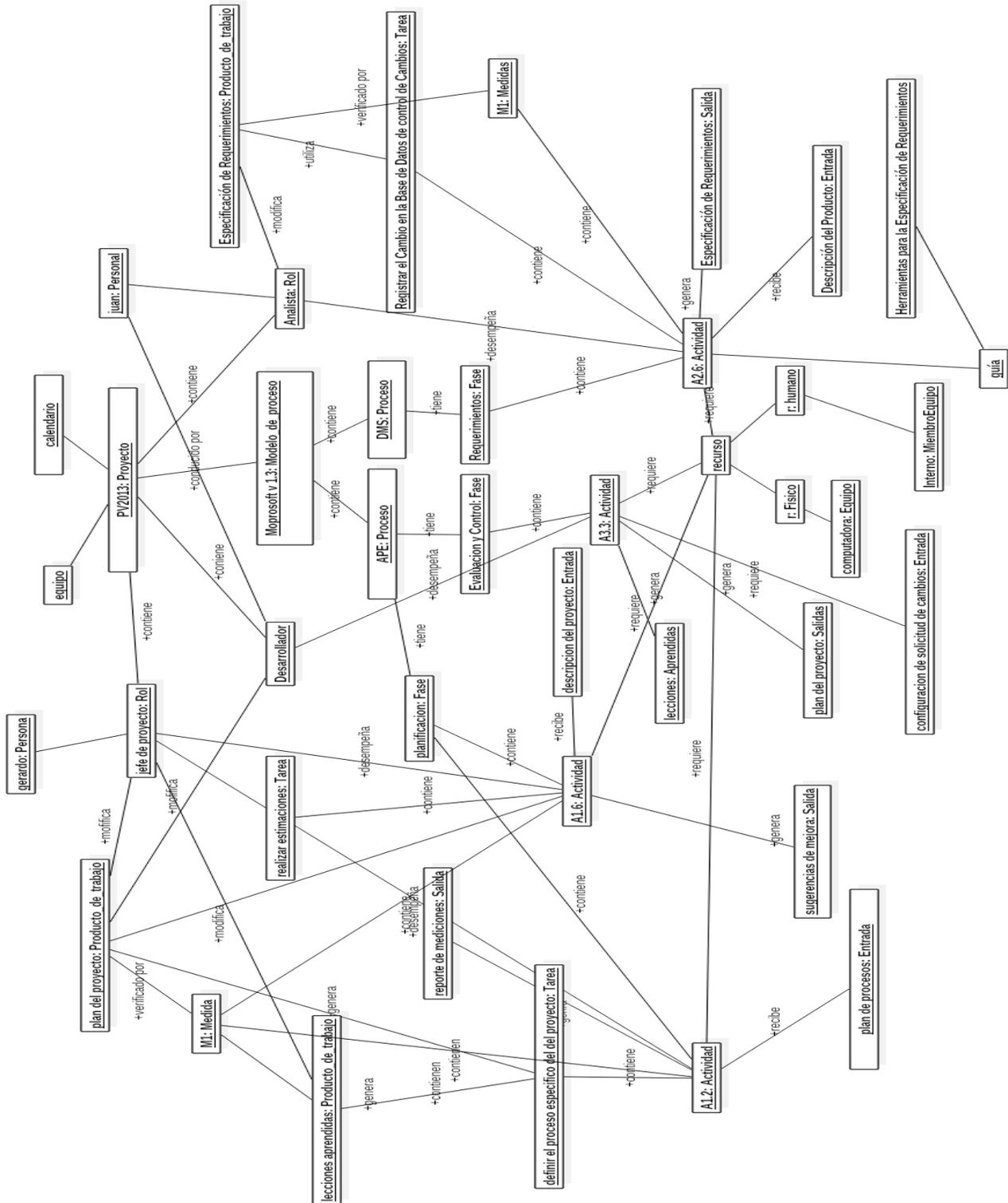
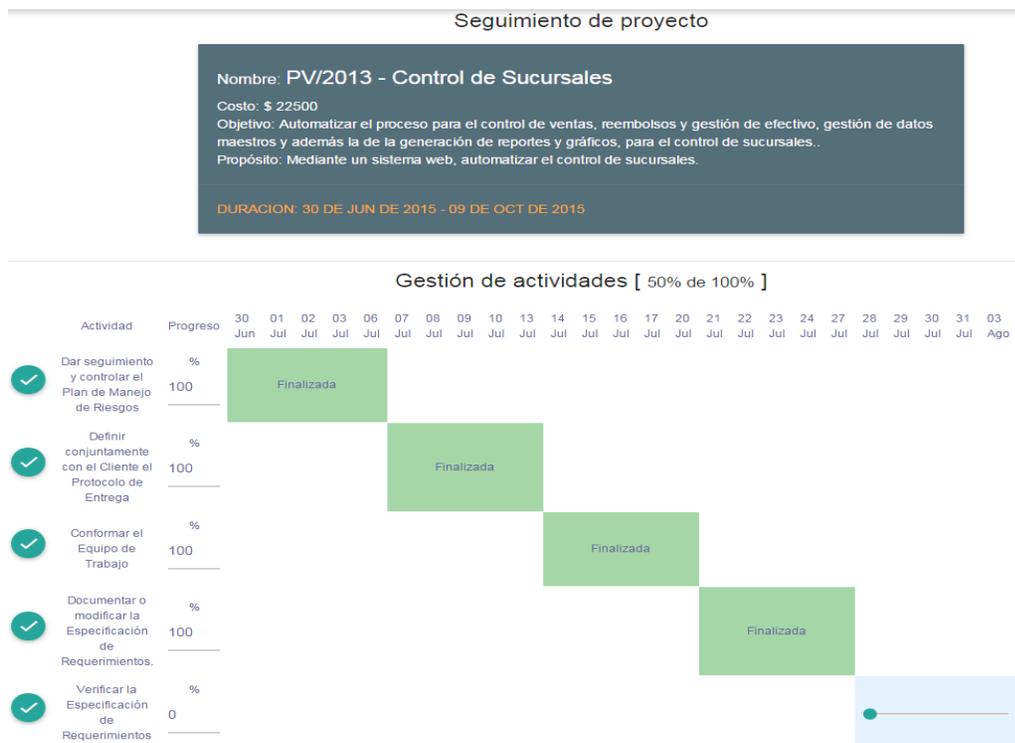


Figura 4.4. Diagrama de objetos del metamodelo.

Una vez que el proyecto piloto fue definido, el patrón **Patrón PV/2013**, que fue creado en base a los activos que han sido precargados a la PAL y que corresponden con los elementos del proceso identificados en el Capítulo 3 de esta tesis, fue asociado a dicho proyecto con la intención

de gestionarlo. Dada la estructura de la PAL, es posible cargar los activos de cualquier otro modelo de procesos sin que la funcionalidad de la herramienta se vea afectada. Esto permite que en algún momento una PyMES combine, en un solo proceso, actividades, productos, medidas y demás elementos no solamente de MoProSoft sino de cuantos modelos desee subir. De esta forma, los procesos definidos por el metamodelo son adaptados a la PyMES de acuerdo a la PAL que contiene las tareas, los procesos, los productos y las medidas a utilizar en cualquier proyecto. En paralelo con el uso de las guías de adaptación y plantillas ofrecidas por el metamodelo, se propuso la realización de reuniones semanales para evaluar el calendario, los objetivos y los riesgos del proyecto. Para esto, el equipo de desarrollo participó desde la planificación inicial y trabajó de acuerdo a los objetivos establecidos por los integrantes del mismo. Debe entenderse que los jefes de proyecto formularon los planes, monitorizaron y controlaron, y desarrollaron el proyecto desde la opción de Control de Proyectos de la PAL (véase Figura 4.5); esto quiere decir que todos los documentos utilizados para desarrollar tales funciones son plantillas que nuestro metamodelo proporciona (véase Anexo B).

Una vez definido el nuevo proceso, se procedió a dar una breve charla informativa y se inició el proyecto con una reunión de lanzamiento. Para esto, la alta dirección de la PyMES estableció objetivos de acuerdo a sus políticas de trabajo y datos históricos que se tenían sobre el proyecto, éstos son presentados en la Tabla 32. A pesar de que el objetivo de esta tesis únicamente pretendía abordar el tiempo que toma un equipo pequeño, dentro de una PyMES, para definir e implantar un proceso de gestión, se respetó la decisión de la empresa para verificar si a través de la implantación del metamodelo era posible reducir las variaciones en el tiempo planificado y el esfuerzo dedicado a la gestión del proyecto. Esta comparación se debería hacer considerando los datos históricos de la empresa para dicho proyecto, los cuales serían utilizados más adelante para analizarlos con los obtenidos a través del uso del metamodelo propuesto.



**Figura 4.5.** Pantalla de la PAL para controlar el progreso del proyecto piloto.

**Tabla 32.** Objetivos a valorar con el proyecto piloto.

Objetivo	Valores históricos
Reducir el tiempo destinado a la definición de un proceso	Entre 4 y 5 semanas
Reducir el tiempo destinado a la implantación de un proceso definido	Entre 15 y 20 semanas (solamente se incluye la modificación del proceso existente para usarlo inmediatamente en los proyectos, esto quiere decir que se realiza a prueba y error)
Reducir la desviación de calendario	17% (considerando el tiempo planeado contra el real para las actividades del proyecto)
Reducir la desviación del esfuerzo	23% (considerando el tiempo planeado contra el real)

#### 4.2.3. Comparación final con la línea base inicial

El propósito de esta comparación, posterior a la implantación del metamodelo a través de la PAL, es comprobar la utilidad del nuevo proceso definido para el proyecto piloto. Tal y como se mostró en la Figura 4.1 de este capítulo, la línea base fue representada por una “pobre” capacidad de acuerdo a las actividades recomendadas por el Nivel 2 de MoProSoft para APE y DMS. Durante los cuatro meses que duró el desarrollo del proyecto piloto se asesoró a la PyMES en el uso del proceso definido y de los activos del proceso a través de la PAL. Al finalizar el proyecto y recoger la información del mismo se realizó una segunda evaluación en la PyMES. Los resultados mostrados en la Tabla 33 evidenciaron que la PyMES incrementó la cobertura de APE y DMS alrededor de un 32.7%, en relación a los valores obtenidos en la primera evaluación. Es decir, la PES1 presentó una cobertura inicial de 19.1% en el proceso de APE y 23.8% en DMS antes de la implantación del metamodelo propuesto. No obstante, una nueva evaluación después de su implantación a través de la PAL mostró, por ejemplo, un aumento en la cobertura de APE del 34.0% (en la primera iteración), el cual es calculado mediante la siguiente fórmula:

$$\text{Incremento} = \frac{(\text{Cobertura después} - \text{Cobertura inicial}) * 100}{\text{Cobertura inicial}} \quad (5)$$

El sesgo en la recolección de estos valores fue controlado mediante la verificación de la documentación generada y el desarrollo de entrevistas con el personal de la empresa.

**Tabla 33.** Incremento en la cobertura de APE y DMS.

PyMES	Proceso	Cobertura		Incremento
		Antes del metamodelo	Después del metamodelo	
PES1	APE	19.1%	25.6%	34.0%
	DMS	23.8%	31.3%	31.5%
<b>Incremento global</b>				<b>32.7%</b>

En este sentido, es importante mencionar que durante la primera iteración se notó un mayor incremento en el proceso de DMS puesto que éste está más relacionado con las actividades que ya se realizaban en la empresa para desarrollar sus productos. Mientras que las actividades de APE (relacionadas con la planificación y control de los proyectos) fueron introducidas sin contar prácticamente con alguna base previa.

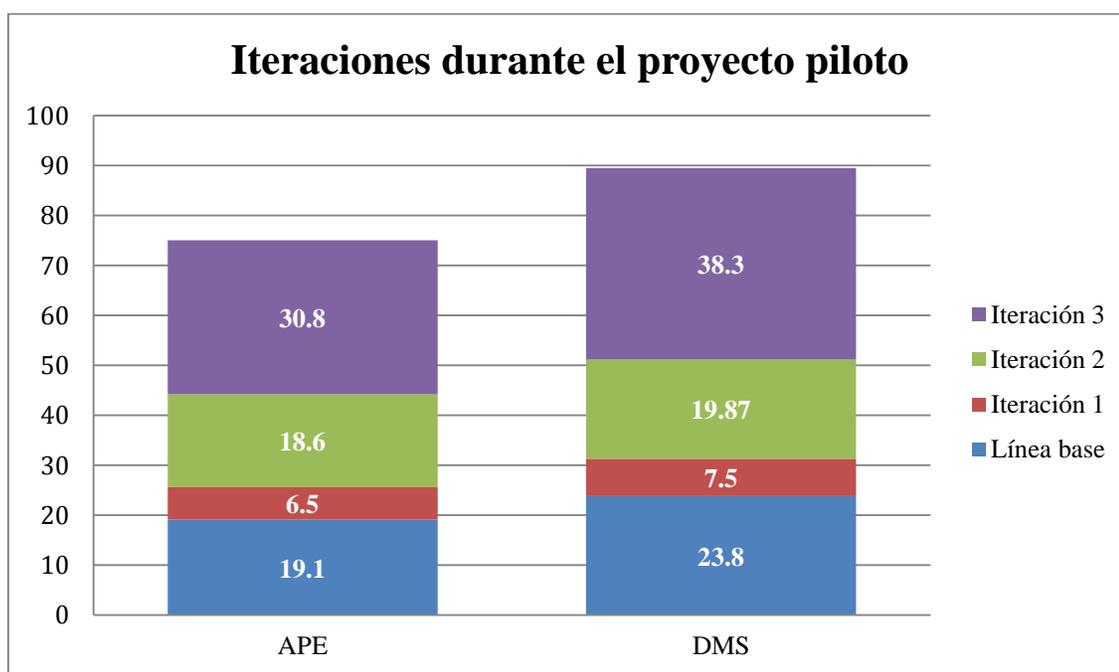
Así pues, después de recoger toda la información relacionada con la experimentación, es importante recalcar también que se determinó que el compromiso, la iniciativa y la responsabilidad de la PyMES fueron factores clave que contribuyeron en gran medida a alcanzar los resultados

resumidos en este capítulo. Esta experimentación también nos permitió obtener una combinación de experiencias valiosas, “más conocimientos sobre cómo hacer las cosas en pequeños entornos de trabajo” y observar el comportamiento del personal durante el proceso de mejora. Finalmente, los resultados mostrados en la Figura 4.5 muestran que después de cuatro iteraciones (desarrolladas a lo largo de los cuatro meses) se logró alcanzar una calificación de “Ampliamente alcanzado”, en el caso de APE, y “Completamente alcanzado”, en el caso de APE, ambos valores de cobertura establecidos por NYCE (véase Tabla 31).

#### 4.2.4. Análisis detallado sobre los resultados del proyecto piloto

De manera adicional a la evaluación sobre el nivel de cobertura en los procesos, las debilidades encontradas en APE y DMS permitieron identificar actividades concretas de la gestión de proyectos que no se estaban realizando correctamente y que fueron mejoradas con la implantación del metamodelo propuesto en esta tesis. Dado lo anterior, se decidió junto con la PyMES recoger información relacionada con los objetivos mostrados en la Tabla 32 para valorar los cambios con la realización del proyecto piloto. En este sentido, el proceso inició con la recogida de datos históricos de proyectos anteriores, con el objetivo de establecer una segunda base comparativa que permitiera determinar la eficiencia (o no) del metamodelo creado.

Así, la Tabla 34 muestra que la PyMES logró terminar el proyecto dentro del calendario planificado. Por otro lado, el esfuerzo obtenido fue negativo puesto que fue posible terminar el proyecto antes que el tiempo estimado por el jefe de proyectos.



**Figura 4.6.** Niveles de cobertura obtenidos al cierre del caso de estudio.

**Tabla 34.** Análisis de resultados sobre los objetivos establecidos por PES1 para el proyecto piloto.

Objetivos	Valor objetivo	Valor con metamodelo	Desviación
Reducir tiempo para definir un proceso	< a 4.5 semanas	3.5 semanas	22.3%
Reducir tiempo para implantar un proceso	< a 17.5 semanas	9 semanas	48.5%
Reducir la desviación del calendario	< a 17%	0.5%	97.0%
Reducir el esfuerzo del proyecto	< a 23%	1.75%	92.3%

Utilizando esta tabla como referencia comparativa es posible observar los beneficios de la implantación del metamodelo mediante la comparación del valor objetivo establecido por la PyMES contra el valor obtenido con la implantación del metamodelo. La desviación calculada permite analizar qué tanto se alejó el valor obtenido del objetivo buscado, por ejemplo el valor de 97.0% mostrado como desviación en la Tabla 34 considera el valor objetivo que debería ser menor del 17%, esto quiere decir que el 100% está representado por el 16.9% (o umbral establecido), por lo que el cálculo se realiza como  $100\% * 0.5$  (que es el valor obtenido a través de la implantación del metamodelo)/16.9, obteniendo un 3.0%. Por lo tanto, la diferencia es del  $100\% - 3.0\% = 97.0\%$ .

Para la incorporación de la gestión efectiva de un proyecto, el metamodelo requirió más tiempo del considerado puesto que no se contempló en un inicio el tiempo que llevaría capacitar al personal en el uso del modelo de procesos escogido para la experimentación (MoProSoft) y el tiempo que se requería para entrenar al personal en el uso de la PAL. Sin embargo, a pesar de que se requirió de este tiempo, los valores de la Tabla 34 muestran una reducción importante en cuanto al tiempo que la empresa tomaba para hacer cambios a sus procesos e implantarlos dentro de sus proyectos. Cabe mencionar que el tiempo destinado a la capacitación no fue tomado en cuenta en este caso de estudio para no sesgar los resultados, ya que éste solamente afectaría a los primeros proyectos piloto realizados en la empresa y posteriormente sería nulo. En relación a la desviación del calendario, el metamodelo incorporó actividades para la planificación del proyecto que permitieron la formulación de planes para mejorar la visión sobre el avance del proyecto piloto, así como actividades para el control del plan de proyecto a través del método del Valor Ganado. Cabe mencionar que estas actividades permitieron que los jefes de proyecto detectaran problemas a tiempo y los corrigieran sin generar algún retraso.

Por último, para apoyar a la PyMES en la adopción del nuevo proceso, fue necesario programar reuniones semanales con los jefes de proyecto para evaluar el calendario y los objetivos. El equipo de desarrollo participó desde la formulación del plan inicial y trabajó de acuerdo a los objetivos establecidos por todos los integrantes. El uso de la PAL permitió que se estableciera un entorno de trabajo más participativo e integrado que es apoyado por el uso de plantillas estándar.

#### 4.2.5. Conclusiones del caso de estudio

Por último, esta sección pretende reunir la evidencia que permita aceptar o rechazar las hipótesis planteadas en el Capítulo 1 de este trabajo de tesis.

- Hipótesis: *“Mediante el uso de un metamodelo un pequeño equipo puede reducir el tiempo para definir e implantar los procesos de planificación y control de proyectos dentro de una PyMES”*.

Esta hipótesis es aceptada considerando los siguientes argumentos. El metamodelo y su herramienta de soporte (la PAL) no requieren de una gran cantidad de recursos ni de una estructura organizativa específica para poder utilizarse en los proyectos de una PyMES. La arquitectura de la PAL permite almacenar material ilustrativo para lograr el entendimiento de las prácticas del modelo que se pretenda adaptar, además de guías de uso (para actividades) que facilitan aún más su uso en el entorno de las PyMES. Otro aspecto relevante es el factor de la inversión económica, dado que sin el requerimiento de una inversión importante destinada a la definición de los procesos dentro de las PyMES, se comprobó que la implantación del metamodelo puede ayudar a este tipo de empresas en la mejora de sus actividades cotidianas. En este sentido, la hipótesis es aceptada puesto que a través de la creación e implantación de un metamodelo es posible reducir el tiempo destinado a la definición e implantación de procesos que guíen de forma correcta a una PyMES en el desarrollo de sus proyectos de software (véase Capítulo 3 de la tesis). De forma adicional, durante la experimentación se generaron las plantillas necesarias para mejorar el desarrollo y la gestión de los

proyectos a través de la PAL (véase Anexo C), y permiten el establecimiento de un conjunto de prácticas y artefactos que ayudan a los jefes de proyectos a controlar el trabajo de su equipo (véase Sección 4.2.2 del Capítulo 4). Además, el metamodelo propuesto es apoyado por una PAL (véase Sección 3.5 del Capítulo 3) que es fácil de usar y entender, y además tiene la particularidad de adaptarse a cualquier tipo de proyecto, pero que además representó una ventaja importante para la PyMES puesto que no solamente le indicó qué hacer si no le recomendó cómo hacerlo. Por último, la Tabla 33 mostró una mejora significativa de la PyMES en relación a los resultados obtenidos para establecer la línea base.

Para finalizar, es importante mencionar que la hipótesis, establecida en el contexto de la tesis, se acepta únicamente para PyMES con características similares a las que tiene la participante en el caso de estudio. De ninguna manera es posible que el metamodelo aquí propuesto produzca los mismos resultados en cualquier otra empresa debido a que esto requiere desarrollar más casos de estudio en una muestra más significativa, ya que no es posible controlar factores externos como: el nivel de conocimiento de los equipos, motivación, compromiso, etc. Sin embargo, de acuerdo con Juristo y Moreno (2013), en el contexto de la experimentación en Ingeniería de Software se afirma que los resultados positivos sobre una efectividad parcial, mostrados en un trabajo de investigación o reporte, evidencian que es posible establecer un caso de estudio con un mayor número de participantes y proyectos.



## 5. Conclusiones

La gestión es una de las áreas fundamentales para el desarrollo de cualquier proyecto dentro de cualquier sector, y el caso de su aplicación a la Ingeniería de Software no es la excepción. En este contexto, la importancia de la gestión radica principalmente en dos procesos básicos (la Planificación de los Proyectos, y el Seguimiento y Control de los Proyectos) que contribuyen notablemente al éxito en el desarrollo de un proyecto. Con el desarrollo de esta tesis fue posible observar que, de acuerdo con la literatura revisada, es evidente que en el contexto de las PyMES existen diversos estudios que simplifican la definición de ambos procesos mediante guías de implantación y ejecución, marcos de trabajo, mapeos entre estándares, y demás; pero no existe mucha información teórica enfocada a definir de manera sistemática mecanismos que simplifiquen su definición y ejecución dentro de las empresas, particularmente en aquellas PyMES que dependen en gran medida de pequeños equipos para la realización de sus proyectos.

En este sentido, cabe resaltar que aunque existen algunos modelos de procesos enfocados a fortalecer las actividades del desarrollo de software dentro de las PyMES, éstos no proporcionan una ayuda adecuada para implantar, de forma eficiente, la gestión de los proyectos dado que asumen que los equipos que los intentan seguir ya poseen el suficiente conocimiento técnico para realizar todas las prácticas ahí resumidas, algo que dista mucho de la realidad. Es así que, tomando en cuenta este escenario, esta tesis consideró el objetivo de desarrollar una propuesta para que las PyMES pudieran implantar, de forma más sencilla, los procesos básicos relacionados con la gestión de proyectos de software en sus pequeños equipos. Más importante aún es el hecho de que dicha propuesta debía ser capaz de adaptarse a las necesidades particulares de las PyMES y de permitir la correcta ejecución de las actividades de gestión aprovechando todo el conocimiento (e.g., lecciones aprendidas) generado por los proyectos exitosos o fracasados.

Así pues, esta tesis propuso la creación de un metamodelo con el objetivo de brindar el soporte correcto a los pequeños equipos creados dentro de las PyMES en la definición e implantación de la gestión de los proyectos. En este sentido, la investigación inició con una búsqueda exhaustiva de literatura que permitió identificar una clara tendencia que aboga por la creación de metamodelos dentro de la Ingeniería de Software y el ámbito empresarial, para promover el concepto de gestión de la calidad al desarrollar productos de software (i.e., en áreas como la medición del software, la ingeniería de requisitos, la validación y verificación del software, y la mejora del proceso). Sin embargo, son escasos los trabajos que exploran el área de la gestión de los proyectos de software. Por lo tanto, el metamodelo propuesto permite la definición de los elementos que un proceso utiliza, sus relaciones y restricciones, de tal forma que facilita la generación de un esquema estandarizado de instancias o modelos más específicos que puede ajustarse a las necesidades de cada proyecto. Así, la arquitectura del metamodelo incluye el concepto de “patrón” para convertir a elementos estándar en elementos personalizados a cada proyecto.

El proceso de creación del metamodelo requirió de una metodología adecuada al contexto de la Ingeniería de Software que se resume de la siguiente manera:

- Para la *definición del ámbito del modelo* fue necesario identificar aquellos elementos requeridos para definir y modelar un proceso, así como identificar los elementos básicos que conforman a un proceso de gestión de proyectos enfocado a las PyMES.
- La *identificación del lenguaje de representación y creación de objetos* requirió una revisión de la literatura existente sobre la creación de metamodelos dentro de la Ingeniería de Software. De acuerdo con esta revisión, se identificó que el lenguaje más utilizado para el modelado de metamodelos es UML debido a las características propias de este lenguaje. Adicionalmente, la selección de este lenguaje está fuertemente justificada por el antecedente del metamodelo SPEM.
- En relación a la *creación del metamodelo y del escenario inicial* fue necesario crear un modelo conceptual que integrara todos los elementos que conducen a la formalización e institucionalización de los conceptos manejados por las PyMES. Así, una de las aportaciones principales que brinda este modelo conceptual es la facilitación del entendimiento y uso del proceso dentro de los pequeños equipos de desarrollo, a través de la representación de las entidades relacionadas con la gestión de los proyectos en cuatro capas de abstracción. En el Nivel M0 se encuentran todos los datos relacionados con los proyectos, en el Nivel M1 se encuentran todos los conceptos relacionados con los procesos (modelos de procesos, plantillas, y mecanismos de evaluación), en el Nivel M2 se encuentran todos los conceptos de la empresa, y finalmente en el Nivel M3 está el depósito de meta-elementos. Cabe mencionar que esta arquitectura por niveles se ajusta a las necesidades de los pequeños equipos dentro de las PyMES con el propósito de facilitar la definición de un proceso para planificar y controlar un proyecto. Dicha arquitectura permite la creación de instancias de un proceso a través del concepto “patrón” (Gonzalez-Perez & Henderson-Sellers, 2008), las cuales son creadas de acuerdo a las necesidades de cada proyecto.
- Finalmente, para *probar el metamodelo* fue necesario crear una PAL que permitiera validar el concepto “patrón” con todos los elementos de un proceso, previamente almacenados en una base de datos. De esta manera se realizó la implementación de los Niveles M2 y M3 del modelo conceptual. Cabe mencionar que la PAL construida es un repositorio de información que almacena todos los activos de proceso (en forma de plantillas) que deben ser manejados por los jefes de proyectos para desarrollar y gestionar un proyecto. De esta forma es que el metamodelo puede llevarse a la práctica para facilitar la implementación de un proceso adecuado a los pequeños equipos a través de instancias aprobadas de un mismo proceso estándar. Es importante resaltar que fue necesario crear todas las plantillas definidas por el Nivel M1 del metamodelo puesto que no existen, hasta el día de hoy, activos similares que puedan descargarse de forma gratuita y que faciliten la ejecución de las actividades de, en el caso de la experimentación, MoProSoft.

Por otro lado, la tesis requirió plantear el uso de la experimentación para probar el metamodelo construido, por lo que fue necesario diseñar un caso de estudio que se enfocara en la implementación de la solución propuesta en una PyMES mexicana de 20 empleados. De esta manera se planteó la posibilidad de recoger datos que permitieran aceptar o rechazar la hipótesis planteada al inicio de la tesis. En este sentido, fue necesario recoger datos históricos de la empresa y desarrollar un proyecto anterior denominado “PV/2013 – Control de Sucursales”, debido a que éste se encontraba mejor documentado. Así, fue posible comparar los resultados obtenidos a través de la

implantación del metamodelo propuesto con los obtenidos por la empresa anteriormente sin utilizar dicho metamodelo. El caso de estudio consistió de las siguientes fases:

- Durante el *establecimiento de la línea base inicial*, primeramente se hizo la definición del alcance de los dos procesos básicos para la gestión de proyectos. La PyMES tenía un interés particular por adoptar las actividades recomendadas por MoProSoft, en este sentido se decidió seguir la definición de los procesos de “Administración de Proyectos Específicos” y “Desarrollo y Mantenimiento de Software”, debido a que estos procesos están relacionados directamente con la gestión de un proyecto a nivel técnico. Posteriormente se procedió a la aplicación de cuestionarios al personal (tanto técnico como directivo). Estos cuestionarios están basados en las actividades recomendadas por MoProSoft para sus Niveles 1 y 2. Los resultados obtenidos proporcionaron información importante sobre el estado actual de la PyMES y permitieron identificar las actividades que requerían atención de manera inmediata.
- Para la *implantación del metamodelo* en un proyecto piloto se recurrió al Nivel M2 del modelo conceptual: la definición de un nuevo proceso. En este sentido, fue necesario asesorar a los jefes de proyectos en la definición del nuevo proceso considerando los activos definidos previamente en la PAL (véase Capítulo 3). Así, el desarrollo del proyecto piloto se basó en el uso de un patrón denominado “ProyectoTesis”. Dada la estructura de la PAL, es posible subir las guías, plantillas, activos, etc., de cualquier otro modelo de procesos sin que la funcionalidad de la herramienta se vea afectada. Esto permitirá que, en algún momento, una PyMES combine, en un solo proceso, actividades, productos, medidas y demás elementos no solamente de MoProSoft sino de cuantos modelos desee utilizar.
- En la *comparación final contra la línea base inicial* se planteó el objetivo de comprobar la utilidad del nuevo proceso definido a través del Nivel M2 del modelo conceptual con ayuda de la PAL. En este sentido, fue posible observar una mejora significativa en la forma de trabajo de la PyMES, en relación a los procesos de APE y DMS. Sin embargo, se mencionó en la experimentación que estos resultados no hubieran sido posible sin el compromiso, la iniciativa, y la responsabilidad del personal de la PyMES. Algo que es importante mencionar es que la experimentación no contabilizó el tiempo extra requerido para la capacitación en el uso de la PAL, ni tampoco el tiempo destinado a la capacitación en los procesos de MoProSoft. Por lo que valdría la pena analizar, en proyectos futuros, estas variables y relacionarlas con el esfuerzo y tiempo requerido para la implantación del metamodelo.

Así pues, el metamodelo propuesto en esta tesis respeta la noción de un proyecto tal cual lo hacen modelos como PMBoK o PSP, y se enfoca a una actividad estratégica para el éxito de los proyectos en las PyMES: la gestión de proyectos en equipos que cuentan con poca o nula experiencia en tales actividades. Sin embargo, tal y como se mencionó anteriormente, una empresa no puede mejorar si no determina primero cuál es su situación actual. En este sentido, el metamodelo también incorporó un mecanismo de evaluación a través del Nivel M1 del modelo conceptual. Así, antes de comenzar a utilizar la PAL los jefes de proyectos deben conducir una evaluación para determinar su situación actual a través de la identificación de fortalezas y debilidades en su forma de trabajo, y a partir de los resultados obtenidos obtener un punto de referencia para identificar aquellas actividades que debe incorporar a su proceso de acuerdo al Nivel M2. En base a lo anterior, se considera que la investigación realizada en esta tesis coincide con los resultados mostrados por García et al. (2006), en el sentido de que uno de los principales objetivos de las empresas desarrolladoras de software es tener éxito en la gestión de sus proyectos con el fin de mejorar la calidad de sus productos, lo cual está estrechamente relacionado con la definición de un proceso de calidad que se ajuste a las necesidades de cada proyecto.

En caso de que una empresa diferente quisiera implantar el metamodelo creado, sería bastante recomendable consultar al director de tesis o a la tesista antes de intentar cualquier cambio. En caso contrario se recomienda, en primer lugar, determinar a través del consenso de la alta dirección, dirección técnica y jefes de proyectos el modelo de procesos con el que se desea trabajar (puesto que esta tesis se enfocó a MoProSoft). Una vez que esto se haya definido, será necesario identificar los procesos relacionados con la gestión de proyectos que se pretende abordar (dado que el caso de estudio presentado en esta tesis se centró en los procesos de APE y DMS en sus Niveles 1 y 2). Así pues, si la empresa en cuestión pretende enfocarse en procesos diferentes de un modelo de referencia diferente al utilizado en esta tesis, será necesario que algún personal especializado elabore las plantillas, productos de trabajo, guías, manuales, y cuestionarios de evaluación que cubran todas y cada una de las actividades de los procesos escogidos. En este sentido, es evidente que esta tesis pretende demostrar que una forma de gestión diferente es posible si se utiliza el concepto de metamodelo siguiendo las pautas establecidas a lo largo de la tesis; pero obviamente se requiere de la asesoría especializada para adoptar este enfoque a las necesidades de cada empresa. Posteriormente, es necesario definir los indicadores de mejora que se desean medir para iniciar con la evaluación de los jefes de proyectos y personal directivo involucrado en la gestión. Una vez terminada la evaluación se procede a determinar la cobertura actual de los procesos tal y como se explicó en el capítulo anterior de la tesis, así como a identificar las oportunidades de mejora. La PAL, que ya tiene previamente todos los activos desarrollados en esta tesis, proporciona toda la información necesaria para iniciar el trabajo de gestión (o el proceso estándar), a partir de la cual el jefe de proyectos tomará decisiones y definirá un patrón (determinado por las actividades que desea mejorar y que fueron identificadas en la evaluación) que se adapte a las necesidades de su proyecto. De esta manera es posible definir e implantar un proceso que se ajusta a las necesidades de un proyecto en particular y de la empresa. Al finalizar el proyecto es necesario repetir la evaluación para medir la cobertura de los procesos y determinar si se cumplió con los objetivos establecidos al inicio.

De manera adicional, y como conclusiones personales sobre esta tesis, considero que su elaboración fue todo un reto, puesto que no contaba con alguna experiencia previa de tesista en el sentido de investigar y escribir un documento. Por lo tanto, realizar todo este trabajo fue algo muy interesante que me ayudó a sentar las bases para la elaboración de un documento que identifica un problema, plantea una hipótesis para su solución, y comprueba si dicha solución funciona o no. Así pues, después de esta experiencia considero también que la metodología utilizada para la creación del metamodelo puede funcionar correctamente para abordar cualquier otro problema dentro del contexto de la Ingeniería de Software, puesto que plantea antes que nada el comenzar con la recopilación de las bases teóricas que ayuden al investigador a situarse en el contexto del problema que se intenta resolver, y después abordar una posible solución. Finalmente, pienso que uno de los aspectos más importantes de esta tesis fue la identificación de los problemas actuales en el desarrollo de software, concretamente aquellos relacionados con la gestión de los proyectos. Esta información puede servir para ayudar al desarrollo de otras tesis similares que pretendan centrarse en ayudar a las PyMES mexicanas, y sobre todo a los pequeños equipos que se forman en éstas, a gestionar mejor sus proyectos con el objetivo de lograr una mejor posición competitiva y, por ende, alcanzar mayores beneficios.

## 6. Anexo A.- Acrónimos

APE	Administración de Proyectos Específicos
CL	Cliente
CMMI	Modelo de Madurez y Capacidad Integrado
CN	Casos de Negocio
COCOMO	Modelo Constructivo de Costos
CP	Calidad Percibida
CPM	Método de la Ruta Crítica
CVS	Concurrent Versions System
CWM	Common Warehouse Meta-model
DMS	Desarrollo y Mantenimiento de Software
DRA	Desarrollo Rápido de Aplicaciones
EDT	Estructura de Descomposición del Trabajo
ERA	Entidad-Relación-Atributo
ERP	Planificación de Recursos Empresariales
ET	Equipo de Trabajo
ETS	Escuela Superior de Tecnología
EV	Valor Ganado
EvalProSoft	Método de Evaluación de Procesos para la Industria de Software
EVM	Medición del desempeño: Gestión del Valor Ganado
FQPMSE	Framework for Quantitative Project Management in Small Enterprises
HIM	Herramienta Integral para MoProSoft
IDC	International Data Corporation
IEEE	Instituto de Ingenieros en Electricidad y Electrónica
IMPA	Asociación Internacional para la Gestión de Proyectos
INEGI	Instituto Nacional de Estadística y Geografía
IPM	Gestión Integrada del Proyecto
IS	Ingeniería de Software
ISO	Organización Internacional para la Estandarización
ITIL	Information Technology Infrastructure Library
MDA	Arquitectura Dirigida por Modelos
MDM	Manejador de Documentos de MoProSoft
MOF	Meta Object Facility
MoProSoft	Modelo de Procesos para la Industria de Software
MiPyMEs	Micro, Pequeñas y Medianas Empresas
NYCE	Organismo de Normalización y Certificación
OMG	Object Management Group

---

PA	Planificación del Alcance
PAL	Biblioteca de Activos
PALSS	Process Asset Library for Small Settings
PERT	Program Evaluation and Review Technique
PI	Paquetes de Implementación
PLB	Plan de Línea Base
PMBok	Project Management Body of Knowledge
PMC	Monitorización y Control del Proyecto
PND	Plan Nacional de Desarrollo
PP	Planificación del Proyecto
PR	Planificación de Riesgos
PROMEPE	Project Management based on Effective Practices
PROSOFT	Programa para el Desarrollo de la Industria de Software
PSP	Personal Software Process
PV	Valor Planificado
PyMES	Pequeñas y Medianas Empresas de Software
QPM	Gestión Cuantitativa del Proyecto
RAPE	Responsable de la Administración del Proyecto Específico
RDM	Responsable de Desarrollo y Mantenimiento de Software
REQM	Gestión de Requisitos
RGPY	Responsable de la Gestión de Proyectos
RIA	Aplicaciones Ricas en Internet
RSC	Responsable del Subcontrato
RSKM	Gestión de Riesgos
SAM	Gestión de Acuerdos con Proveedores
SC	Satisfacción del Cliente
SE	Secretaría de Economía
SEI	Instituto de Ingeniería de Software
SI	Sistemas de Información
SNIITI	Sistema Nacional de Indicadores de la Industria de Tecnologías de Información
SPA	Modelo de Evaluación del Proceso de Software
SPEM	Software & Systems Process Engineering Metamodel Specification
SPI	Mejora del Proceso de Software
SV	Variación del Cronograma
SVN	Subversión
SW	Software
TI	Tecnologías de Información
TSP	Team Software Process
UML	Lenguaje Unificado de Modelado
WBS	Estructura Desglosada de Trabajo
XP	Programación Extrema

## **7. Anexo B.- Cuestionarios de evaluación para la experimentación**



## Evaluación del Proceso de Administración de Proyectos Específicos

### PLANIFICACIÓN (A.1)

I. La planificación se refiere al conjunto de actividades cuya finalidad es obtener y mantener el *Plan del Proyecto* y el *Plan de Desarrollo* que regirán al proyecto específico, con base en la *Descripción del Proyecto*.

1. ¿Existe en la organización un *Responsable de la Gestión de Proyectos*? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*Dependiendo del tipo de producto, la persona que lidera el proyecto puede cambiar de un proyecto a otro.*

Comentarios:

2. ¿Existe en la organización un documento que defina los objetivos y el alcance de un proyecto? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*Este documento, la Descripción del Proyecto, debe contener la Descripción del propósito y del producto, el Alcance del Proyecto, los Objetivos del mismo, los Entregables esperados, los Supuestos y premisas, la Necesidad de negocio, entre otros.*

Comentarios:

3. ¿El Responsable de la Gestión de Proyectos revisa la *Descripción del Proyecto*? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*La Descripción del Proyecto es revisada con la finalidad de que pueda definirse una estrategia que cubra los objetivos y entregables establecidos para el proyecto.*

Comentarios:

4. ¿Existe un *Protocolo de Entrega* para cada proyecto aceptado por la organización? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*El Protocolo de Entrega debe establecer los términos en que cada entregable del proyecto será recibido en tiempo y forma por el Cliente.*

Comentarios:

5. ¿Este *Protocolo de Entrega* es definido por el *Responsable de Administración de Proyectos Específicos* y por el *Cliente*? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*El Protocolo de Entrega debe ser establecido y aceptado por todos los involucrados en esta actividad del proyecto.*

Comentarios:

6. ¿Existe en la organización un proceso que identifique los *Ciclos y Actividades* específicas a realizar para producir los *Entregables* de un proyecto? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*Este proceso resume las actividades a realizar para cumplir con el Protocolo de Entrega. La determinación de las fases del proyecto incluye, en la mayoría de los casos, la selección y refinamiento de un modelo de desarrollo de software para establecer las interdependencias y la secuencia apropiada de las actividades del proyecto.*

Comentarios:

7. ¿Se consensa cada una de las actividades con aquellos que las llevarán a cabo? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*Para esta respuesta debe reflexionarse ¿El personal es considerado para seleccionar aquellas actividades que es capaz de realizar?*

Comentarios:



*La opción de entrenamiento interno o contratación del conocimiento y las habilidades necesarias se determinan por la disponibilidad de experiencia en el entrenamiento, del calendario del proyecto y de los objetivos de negocio.*

Comentarios:

16. ¿Existe en la organización un *Plan de Capacitación para la Adquisición de Materiales* en caso de que el proyecto lo requiera?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

17. ¿Existe en la organización un *Plan de Capacitación para el uso del Equipo* en caso de que el proyecto lo requiera?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

18. ¿Existe en la organización un *Plan de Capacitación para el uso de Herramientas* en caso de que el proyecto lo requiera?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

19. ¿Se utiliza toda la información de los planes anteriores para elaborar el *Plan de Adquisiciones y Capacitación* de la organización?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Plan de Adquisiciones y Capacitación concentra las solicitudes de adquisición de recursos. Regularmente se incluye necesidades de capacitación al personal, requisitos a proveedores, necesidades de infraestructura y herramientas, así como requerimientos de capacitación en general.*

Comentarios:

20. ¿Se cuenta con fichas de personal que permitan identificar roles y habilidades para formar un *Equipo de Trabajo* por cada proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Las fichas de personal deberán concentrar toda la información relacionada con las habilidades, certificaciones, educación, y entrenamiento de cada miembro del Equipo de Trabajo para la asignación correcta de roles.*

Comentarios:

21. ¿Existen roles definidos por la organización que establezcan tareas y responsabilidades dentro del *Equipo de Trabajo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

22. ¿Se establecen responsables dentro del *Equipo de Trabajo* para cada actividad establecida en la *Descripción del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

23. ¿Se establece un mecanismo que asegure que las actividades de la *Descripción del Proyecto* sean coordinadas y realizadas?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Los hitos son establecidos a menudo para asegurar la terminación de ciertos entregables en el proyecto. Los hitos pueden estar basados en eventos o en el calendario.*

Comentarios:



Comentarios:

31. ¿Se documenta una *Solicitud de Cambios* sobre los requisitos del proyecto?

*La Solicitud de Cambios se relaciona con el proceso de control de cambios mediante el cual se propone, se evalúa, se aprueba o se rechaza, se programa y se le da seguimiento a la modificación del software. El fundamento básico es un proceso de control de cambios, un estado de los componentes, presentación de informes, y un proceso de evaluación. El control de cambios en el software es un proceso de decisión utilizado en el control de los cambios realizados al mismo. Algunos cambios propuestos son aceptados e implementados durante este proceso, otros son rechazadas o postpuestos, y no se aplican.*

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

32. ¿Se determina el origen de este cambio?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

33. ¿Existe en la organización un responsable de gestionar el cambio del requisito desde su inicio hasta su cierre?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

34. ¿Se actualiza el *Plan del Proyecto* con cada petición de cambio?

*El Plan del Proyecto es un documento formal usado como guía para la ejecución y control del proyecto. En este punto, este documento está formado por Ciclos y Actividades, Tiempo Estimado, Plan de Adquisiciones y Capacitación, Equipo de Trabajo, Costo Estimado, Calendario, Plan de Manejo de Riesgos, y Protocolo de Entrega.*

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

35. ¿Se definen hitos para la actualización del *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

36. ¿Se establece el control de versiones ante la gestión de los cambios requeridos?

*Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Una versión, revisión o edición de un producto, es el estado en el que se encuentra dicho producto en un momento dado de su desarrollo o modificación. Aunque un sistema de control de versiones puede realizarse de forma manual, es muy aconsejable disponer de herramientas que faciliten esta gestión dando lugar a los llamados Sistemas de Control de Versiones o SVC (System Version Control).*

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

37. ¿Existe en la organización un *Plan de Desarrollo* independiente del *Plan del Proyecto*?

*El Plan de Desarrollo define una estrategia de alto nivel para el desarrollo de los productos requeridos por el Cliente que contiene: Descripción del Producto y Entregables, Proceso Específico, Equipo de Trabajo, Calendario.*

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:



Comentarios:

**REALIZACIÓN (A.2)**

**II. La realización consiste en llevar a cabo las actividades establecidas en el *Plan del Proyecto*.**

1. ¿Participa el *Responsable de Desarrollo y Mantenimiento de Software* en el desarrollo del *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

2. ¿Existe en la organización un *Plan de Comunicación e Implantación* definido para distribuir la información necesaria al *Equipo de Trabajo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La información de todos los proyectos debe estar almacenada de forma comprensible (i.e. salida electrónica o computarizada de una base de datos) o representada en su forma original y debe ser compartida para que el Equipo de Trabajo pueda consultarla.*

Comentarios:

3. ¿Se distribuye la información del proyecto al *Equipo de Trabajo* de tal forma que cada miembro reciba la información que necesita?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*No todos los participantes en el proyecto tendrán la necesidad o la autorización necesaria para acceder a los datos del proyecto. El Plan de Comunicación e Implantación debe establecer procedimientos para identificar quién tiene acceso a qué datos y cuándo accede a ellos.*

Comentarios:

4. ¿Se da seguimiento al *Plan de Adquisiciones y Capacitación* según sea apropiado?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

5. Una vez que la *Asignación de Recursos* es aceptada o rechazada, ¿se documenta la distribución de los recursos a los miembros del *Equipo de Trabajo* para que éstos realicen las actividades?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La definición de los recursos (habilidades/conocimiento, maquinaria/equipo, materiales, métodos, herramientas) y sus cantidades necesarias para realizar las actividades del proyecto se basa en la estimación inicial y proporciona información adicional que se puede aplicar para extender los Ciclos y Actividades usados en la administración del proyecto.*

Comentarios:

6. ¿Se planifican, revisan y auditan las actividades de los subcontratistas, asegurando la calidad de los productos o servicios contratados y el cumplimiento con los estándares y especificaciones acordadas?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

7. ¿Se generan *Reportes de Actividades*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Un Reporte de Actividades es un registro periódico de actividades, fechas de inicio y fin, responsables y mediciones, tales como: Tiempo de producción, de Corrección, de Verificación y de Validación, Defectos encontrados en verificación, Defectos*



Comentarios:

16. ¿Se realizan reuniones de revisión con el <i>Equipo de Trabajo</i> ?	Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
	<input type="checkbox"/>						

Comentarios:

17. ¿Se documentan en <i>Minutas</i> los puntos tratados y los acuerdos tomados entre el <i>Equipo de Trabajo</i> ?	Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
	<input type="checkbox"/>						

*Una Minuta es un resumen escrito de las reuniones, la cual describe quiénes asistieron, qué asuntos se trataron y los acuerdos o compromisos a que se llegó, resaltando quién y cuándo tendrá lista alguna solución.*

Comentarios:

18. ¿Se realizan reuniones de revisión con el <i>Cliente</i> ?	Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
	<input type="checkbox"/>						

Comentarios:

19. ¿Se documentan en <i>Minutas</i> los puntos tratados y los acuerdos tomados con el <i>Cliente</i> ?	Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
	<input type="checkbox"/>						

Comentarios:

**EVALUACIÓN Y CONTROL (A.3)**

III. La evaluación y el control se aseguran que los *Objetivos* del proyecto sean cumplidos a través de la supervisión y evaluación del progreso del mismo. En caso de que sean detectadas desviaciones en el *Calendario* original, se establecen *Acciones Correctivas* según sea apropiado. Como resultado de estas actividades el *Plan del Proyecto* y el *Plan de Desarrollo* serán actualizados.

1. ¿Se realiza una evaluación sobre el avance del <i>Plan del Proyecto</i> ?	Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
	<input type="checkbox"/>						

*Los parámetros de la planificación del proyecto constituyen indicadores típicos del progreso y rendimiento del mismo e incluyen atributos de productos de trabajo tales como actividades, costos, esfuerzo y tiempo. El seguimiento normalmente implica medir los valores reales de los parámetros de la planificación del proyecto, comparar estos valores con los estimados en el Plan del Proyecto e identificar las desviaciones significativas.*

Comentarios:

2. ¿Se evalúa el cumplimiento actual del alcance del proyecto?	Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
	<input type="checkbox"/>						

*Esta evaluación mide periódicamente la completitud real de las actividades e hitos, y se compara contra el Plan del Proyecto.*

Comentarios:

3. ¿Se documenta este nivel de cumplimiento?	Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
	<input type="checkbox"/>						

*El documentar los valores reales de los parámetros de la planificación del proyecto incluye el registrar la información contextual asociada para ayudar a comprender la comparación e identificar desviaciones importantes que deben ser tratadas.*

Comentarios:

4. ¿Se evalúa el cumplimiento actual del *Costo* estimado en el *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La evaluación del costo empleado en el proyecto normalmente incluye: Comparar periódicamente los costos reales incurridos, Comparar los costos reales con las estimaciones y Costo estimado en el Plan del Proyecto, Identificar las desviaciones significativas del Costo estimado en el Plan del Proyecto.*

Comentarios:

5. ¿Se documenta este nivel de cumplimiento?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La documentación incluye a las desviaciones significativas encontradas.*

Comentarios:

6. ¿Se evalúa el cumplimiento actual del *Calendario* establecido en el *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La evaluación del progreso incluye: Comparar el grado de avance real de las actividades e hitos frente al Calendario del Plan del Proyecto, Identificar las desviaciones significativas de las estimaciones del Calendario del Plan del Proyecto.*

Comentarios:

7. ¿Se documenta este nivel de cumplimiento?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La documentación incluye a las desviaciones significativas encontradas.*

Comentarios:

8. ¿Se evalúa el cumplimiento actual del *Equipo de Trabajo* establecido en el *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La evaluación del esfuerzo realizado por el Equipo de Trabajo normalmente incluye: Comparar periódicamente el esfuerzo del personal asignado al Equipo de Trabajo, Comparar el esfuerzo con las estimaciones del Plan del Proyecto, Identificar las desviaciones significativas del esfuerzo estimado para el Equipo de Trabajo en el Plan del Proyecto.*

Comentarios:

9. ¿Se documenta este nivel de cumplimiento?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La documentación incluye a las desviaciones significativas encontradas.*

Comentarios:

10. ¿Se evalúa el cumplimiento actual sobre los *Ciclos y Actividades* establecidos en el *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Los directivos, el Equipo de Trabajo, los Clientes, los Usuarios finales, los subcontratistas y otras partes interesadas se incluyen en la evaluación según sea apropiado. La evaluación del progreso sobre los Ciclos y Actividades se realizan con regularidad (p.ej., semanal, mensual o trimestralmente).*

Comentarios:

11. ¿Se documenta este nivel de cumplimiento?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La documentación incluye a las desviaciones significativas encontradas.*

Comentarios:

12. ¿Se establecen *Acciones Correctivas* para resolver los problemas encontrados en las evaluaciones anteriores sobre el *Plan del Proyecto*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Las Acciones Correctivas son establecidas y gestionadas hasta su cierre para corregir una desviación o problema con respecto al cumplimiento del Plan del Proyecto y el Plan de Desarrollo. Algunos ejemplos de Acciones Correctivas son: Modificar el Plan de Desarrollo, Modificar los requisitos, Modificar las estimaciones y los planes, Renegociar los compromisos, Añadir recursos, Cambiar los procesos, Modificar los riesgos del proyecto.*

Comentarios:

13. ¿Se realiza el seguimiento del *Plan de Manejo de Riesgos* para identificar, controlar o actualizar los riesgos durante el desarrollo del proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Este plan contiene la identificación y evaluación de riesgos, así como los planes de contención y de contingencia correspondientes. A medida que avanzan los proyectos (especialmente aquellos de larga duración), surgen nuevos riesgos por lo que es importante identificarlos y evaluarlos. Por ejemplo, el software, el equipo y las herramientas en uso pueden volverse obsoletos; o los miembros del Equipo de Trabajo pueden perder gradualmente las habilidades en áreas de particular importancia a largo plazo para el proyecto y la organización.*

Comentarios:

14. ¿Se genera el *Reporte de Seguimiento* del proyecto considerando los *Reportes de Actividades* generados?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Reporte de Seguimiento contiene el registro del avance de las actividades realizadas incluyendo las realizadas en el Plan de Manejo de Riesgos. El avance se registra por ciclo, incluyendo fecha de inicio y fin. Además, este reporte contiene el registro periódico de las mediciones como: Costo real del proyecto, Esfuerzo realizado, Cambios implementados y clasificados por tipo, Tiempo real invertido, Defectos encontrados, Tamaño de los productos y Trabajo duplicado.*

Comentarios:

**CIERRE (A.4)**

IV. El cierre consiste en entregar los productos de acuerdo al *Protocolo de Entrega* y dar por concluido el ciclo o proyecto. Como resultado final se genera el *Documento de Aceptación* del *Cliente* que da por terminado el desarrollo del proyecto.

1. ¿Se establece y se cumple con un *Protocolo de Entrega* en cada proyecto realizado por la organización?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Protocolo de Entrega resume los términos establecidos por la organización y el Cliente para entregar y recibir, respectivamente, el producto terminado.*

Comentarios:

2. ¿Se documenta el seguimiento y evaluación de dicho *Protocolo de Entrega*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Esta evaluación mide la completitud real de los elementos establecidos en el Protocolo de Entrega, y se compara contra lo establecido en el Plan del Proyecto.*

Comentarios:

3. Una vez terminado el proyecto, ¿se efectúa el cierre de las actividades subcontratadas con terceros?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Es importante asegurarse de tratar todas las cuestiones relativas a la propiedad intelectual del producto (o componente) adquirido antes de que el proyecto lo*

acepte.

Comentarios:

4. ¿Se obtiene el *Documento de Aceptación del Cliente* para dar por finalizado el proyecto?

Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*Este documento establece la aceptación del Cliente de los entregables establecidos en el proyecto y da por terminada la responsabilidad de la organización en todo lo relacionado con el desarrollo del proyecto.*

Comentarios:

## Evaluación del Proceso de Desarrollo y Mantenimiento de Software

### REALIZACIÓN DE LA FASE DE INICIO (A.1)

I. Durante la fase de inicio los miembros del *Equipo de Trabajo* revisan el *Plan de Desarrollo* para lograr un entendimiento común del proyecto y para obtener el compromiso de su realización.

1. ¿Se revisa con los miembros del *Equipo de Trabajo* el *Plan de Desarrollo* actual para entender el proyecto y comprometerse con su desarrollo?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Plan de Desarrollo, generado por el proceso de Administración de Proyectos Específicos, puede omitirse en el caso de que los roles de Responsable de Administración de Proyectos Específicos y de Responsable de Desarrollo y Mantenimiento sean desempeñados por una misma persona. Si éste fuera el caso, el documento que registró el proceso de Desarrollo y Mantenimiento de Software será el Plan del Proyecto.*

Comentarios:

2. ¿Se elabora un Reporte de Actividades registrando las actividades realizadas en la revisión anterior, las fechas de inicio y fin, y el responsable por cada actividad?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Un Reporte de Actividades es un registro periódico de actividades, fechas de inicio y fin, responsables y mediciones, tales como: Tiempo de producción, de Corrección, de Verificación y de Validación, Defectos encontrados en verificación, Defectos encontrados en validación, Defectos encontrados en pruebas, Tamaño de Productos.*

Comentarios:

### REALIZACIÓN DE LA FASE DE REQUISITOS (A.2)

II. Durante la fase de requisitos se desarrolla el conjunto de actividades cuya finalidad es obtener la documentación de la *Especificación de Requisitos* y el *Plan de Pruebas de Sistema* para conseguir un entendimiento común entre el *Cliente* y el proyecto.

1. ¿Son distribuidas las tareas correspondientes a esta fase entre los analistas de acuerdo al *Plan de Desarrollo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Los analistas poseen el conocimiento y experiencia para la obtención, especificación y análisis de los requisitos.*

Comentarios:

2. ¿Estos analistas identifican y consultan fuentes de información (clientes, usuarios, sistemas anteriores, documentos existentes, etc.) para obtener los nuevos requisitos?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

3. ¿Son analizados los requisitos identificados en la actividad anterior para delimitar su alcance y factibilidad, considerando las restricciones del ambiente de negocio del *Cliente* o del proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Aquellos requisitos que no sean factibles de desarrollo serán descartados y no formarán parte de la Especificación de Requisitos.*

Comentarios:

4. ¿Se elabora o modifica un prototipo de interfaz de usuario que permita facilitar el entendimiento sobre el proyecto entre el *Cliente* y la organización?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Esta interfaz de usuario contiene elementos de diseño visual que le proporcionan al usuario una idea de las interfaces que mostrará el sistema, con el fin de obtener realimentación sobre los requisitos del mismo. Es posible generar unas cuantas imágenes de pantalla o bien un esqueleto ejecutable de interfaces de usuario.*

Comentarios:

5. ¿Los analistas generan o actualizan la *Especificación de Requisitos*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La Especificación de Requisitos regularmente se compone de una introducción y una descripción de requisitos. Los requisitos son descritos como: Funcionales, de Interfaz de usuario, de Confiabilidad, de Eficiencia, de Mantenimiento, de Portabilidad, de Restricciones de diseño y construcción, Legales y reglamentarios, etc.*

Comentarios:

6. ¿Se verifica la *Especificación de Requisitos* antes de ser aprobada?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La verificación consiste en revisar la claridad de la redacción de la Especificación de Requisitos y su consistencia con la Descripción del Producto y con el estándar de documentación requerido en el Proceso Específico. Adicionalmente se revisa que los requisitos sean completos y no ambiguos o contradictorios.*

Comentarios:

7. ¿Se corrigen los defectos encontrados en la *Especificación de Requisitos* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.*

Comentarios:

8. ¿Se valida la *Especificación de Requisitos* antes de ser aprobada?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La validación es la actividad que confirma que la Especificación de Requisitos resultante cumple con las necesidades y expectativas acordadas, incluyendo las pruebas sobre la interfaz de usuario. Los defectos encontrados se documentan en un Reporte de Validación.*

Comentarios:

9. ¿Se corrigen los defectos encontrados en la *Especificación de Requisitos* con base en el *Reporte de Validación* y se obtiene la aprobación de las correcciones?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Reporte de Validación contiene el registro de los participantes en la validación, la fecha y lugar en que se realiza, y la duración.*

Comentarios:

10. ¿Se elabora o modifica el *Plan de Pruebas de Sistema* que será utilizado en la entrega del software al *Cliente*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Plan de Pruebas de Sistema identifica las pruebas requeridas para el cumplimiento de los requisitos especificados. Una práctica común para obtener el Plan de Pruebas de Sistema es hacer que el Cliente participe en la definición de los casos de prueba.*

Comentarios:

11. ¿Se verifica el *Plan de Pruebas de Sistema* antes de ser aprobado? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*La verificación consiste en revisar la consistencia del Plan de Pruebas de Sistema en relación con la Especificación de Requisitos y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.*

Comentarios:

12. ¿Se corrigen los defectos encontrados en el *Plan de Pruebas de Sistema* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.*

Comentarios:

13. ¿Se documenta la versión preliminar del *Manual de Usuario* o bien se modifica uno ya existente? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*Esta versión preliminar del Manual de Usuario servirá también para que el Cliente y la organización logren el entendimiento del proyecto y los objetivos establecidos para éste.*

Comentarios:

14. ¿Se verifica el *Manual de Usuario* antes de ser aprobado? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*La verificación consiste en revisar la consistencia del Manual de Usuario en relación con la Especificación de Requisitos y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.*

Comentarios:

15. ¿Se corrigen los defectos encontrados en el *Manual de Usuario* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.*

Comentarios:

16. Una vez que ya han sido verificados los documentos, ¿se incorporan la *Especificación de Requisitos*, los *Planes de Pruebas de Sistema*, y el *Manual de Usuario* como líneas base a la *Configuración de Software*? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

*La Configuración de Software consiste de todos los productos de software generados con el Desarrollo y Mantenimiento de Software: Especificación de Requisitos, Análisis y Diseño, Software, Registro de Rastreo, Plan de Pruebas de Sistema, Reporte de Pruebas dl Sistema, Plan de Pruebas de Integración, Reporte de Pruebas de Integración, Manual de Usuario, Manual de Operación, Manual de Mantenimiento. La Configuración de Software suele contener referencias a todos los documentos anteriores de forma que facilite su ubicación para consulta.*

Comentarios:

17. ¿Se elabora el *Reporte de Actividades* registrando las actividades realizadas, fechas de inicio y fin, y responsable por actividad? Siempre Usualmente Algunas Veces Rara Vez Nunca No Sabe No Aplica

Comentarios:

**REALIZACIÓN DE LA FASE DE ANÁLISIS Y DISEÑO (A.3)**

III. Durante la fase de construcción se desarrolla el conjunto de actividades para analizar los requisitos especificados para producir una descripción de la estructura de los componentes de software, la cual servirá de base para la construcción. Como resultado de esta fase se obtendrá la documentación del *Análisis y Diseño* y *Plan de Pruebas de Integración*.

1. ¿Son distribuidas las tareas correspondientes a esta fase entre los analistas o diseñadores de acuerdo al *Plan de Desarrollo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Los analistas poseen el conocimiento y la experiencia para la obtención, especificación y análisis de los requisitos; mientras que el diseñador posee el conocimiento y la experiencia en el diseño de la estructura de los componentes de software. En empresas pequeñas, ambos roles suelen ser desempeñados por una misma persona.*

Comentarios:

2. ¿Se analiza la *Especificación de Requisitos* para generar la descripción de la estructura interna del sistema y su descomposición en subsistemas, y éstos a su vez en componentes, definiendo las interfaces entre ellos?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

3. ¿Se describe el detalle de la apariencia y el comportamiento de la interfaz con base en la *Especificación de Requisitos* de tal forma que sea posible prever los recursos para su implementación?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

4. ¿Se describe el detalle de los componentes de tal forma que sea evidente su construcción?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

5. ¿Se genera o actualiza el *Análisis y Diseño* del proyecto?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

6. ¿Se verifica el *Análisis y Diseño* antes de ser aprobado?

*La verificación consiste en revisar la claridad de la documentación del Análisis y Diseño, su factibilidad y la consistencia con la Especificación de Requisitos y con el estándar de documentación requerido en el Proceso Específico.*

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

7. ¿Se corrigen los defectos encontrados en el *Análisis y Diseño* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.*

Comentarios:



16. Una vez que ya han sido verificados los documentos, ¿se incorporan el *Análisis y Diseño*, el *Registro de Rastreo*, y el *Plan de Pruebas de Integración* como líneas base a la *Configuración de Software*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La Configuración de Software consiste de todos los productos de software generados con el Desarrollo y Mantenimiento de Software: Especificación de Requisitos, Análisis y Diseño, Software, Registro de Rastreo, Plan de Pruebas de Sistema, Reporte de Pruebas dl Sistema, Plan de Pruebas de Integración, Reporte de Pruebas de Integración, Manual de Usuario, Manual de Operación, Manual de Mantenimiento. La Configuración de Software suele contener referencias a todos los documentos anteriores de forma que facilite su ubicación para consulta.*

Comentarios:

17. ¿Se elabora el *Reporte de Actividades* registrando las actividades realizadas, fechas de inicio y fin, y responsable por actividad?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

### **REALIZACIÓN DE LA FASE DE CONSTRUCCIÓN (A.4)**

IV. Durante la fase de construcción se desarrolla el conjunto de actividades para producir el (los) *Componente(s)* de software que correspondan al *Análisis y Diseño*, así como la realización de pruebas unitarias. Como resultado de esta fase se obtendrán el (los) *Componente(s)* de software ya probados.

1. ¿Son distribuidas las tareas correspondientes a esta fase entre los programadores de acuerdo al *Plan de Desarrollo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Los programadores poseen el conocimiento y/o la experiencia en la programación, integración y realización de pruebas unitarias.*

Comentarios:

2. ¿Se implementa o se modifica el (los) *Componente(s)* de software con base al *Análisis y Diseño* obtenido en la fase anterior?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Un Componente es un conjunto de unidades de código relacionadas con un fin.*

Comentarios:

3. ¿Son definidas y aplicadas por los programadores las pruebas unitarias que verifiquen el funcionamiento de cada *Componente* de acuerdo con el *Análisis y Diseño*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

4. ¿Son corregidos los defectos encontrados hasta lograr pruebas unitarias exitosas (es decir, sin defectos)?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

5. ¿Se actualiza el *Registro de Rastreo* incorporando el (los) *Componente(s)* construidos o modificados?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

6. ¿Se verifica el *Registro de Rastreo* antes de ser aprobado?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La verificación consiste en asegurar que el Registro de Rastreo contenga las relaciones adecuadas entre los elementos de Análisis y Diseño y el (los) Componente(s). Los defectos encontrados se documentan en un Reporte de*

## Verificación.

## Comentarios:

7. ¿Se corrigen los defectos encontrados en el *Registro de Rastreo* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.*

## Comentarios:

8. Una vez que ya han sido verificados los documentos, ¿se incorporan el (los) *Componente(s)* y el *Registro de Rastreo* como líneas base a la *Configuración de Software*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La Configuración de Software consiste de todos los productos de software generados con el Desarrollo y Mantenimiento de Software: Especificación de Requisitos, Análisis y Diseño, Software, Registro de Rastreo, Plan de Pruebas de Sistema, Reporte de Pruebas dl Sistema, Plan de Pruebas de Integración, Reporte de Pruebas de Integración, Manual de Usuario, Manual de Operación, Manual de Mantenimiento. La Configuración de Software suele contener referencias a todos los documentos anteriores de forma que facilite su ubicación para consulta.*

## Comentarios:

9. ¿Se elabora el *Reporte de Actividades* registrando las actividades realizadas, fechas de inicio y fin, y responsable por actividad?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

## Comentarios:

**REALIZACIÓN DE LA FASE DE INTEGRACIÓN Y PRUEBAS (A.5)**

V. Durante la fase de integración y pruebas se realiza el conjunto de actividades para integrar y probar el (los) *Componente(s)* de software, en base a los *Planes de Pruebas de Integración y de Sistema*, con la finalidad de obtener el *Software* que satisfaga los requisitos especificados. Se genera también las versiones finales del *Manual de Usuario*, *Manual de Operación* y *Manual de Mantenimiento*. Como resultado final se obtiene el producto de *Software* ya probado y documentado.

1. ¿Son distribuidas las tareas correspondientes a esta fase entre los programadores o responsables de pruebas de acuerdo al *Plan de Desarrollo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*Los responsables de pruebas tienen conocimiento y experiencia en la planificación y realización de pruebas de integración y de sistema.*

## Comentarios:

2. ¿Son integrados el (los) *Componente(s)* en subsistemas o en el sistema de *Software* y se aplican las pruebas siguiendo el *Plan de Pruebas de integración*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

## Comentarios:

3. ¿Son documentados los resultados de las pruebas en un *Reporte de Pruebas de Integración*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Reporte de Pruebas de Integración registra a los participantes, la fecha y lugar en que se realiza, la duración y los defectos encontrados.*

## Comentarios:

4. ¿Son corregidos los defectos encontrados, con base en el Reporte de Pruebas de Integración, hasta lograr una prueba de integración exitosa (es decir, sin defectos)?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

5. ¿Se actualiza el *Registro de Rastreo*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

6. ¿Se documenta el *Manual de Operación* o bien se modifica uno ya existente?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Manual de Operación es un documento electrónico o impreso que contiene la información indispensable para la instalación y administración del software, así como el entorno de operación (sistema operativo, base de datos, servidores, etc.). Este manual deberá ser redactado en términos comprensibles para el personal responsable de la operación del Software.*

Comentarios:

7. ¿Se verifica el *Manual de Operación* antes de ser aprobado?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La verificación consiste en asegurar la consistencia del Manual de Operación con el Software y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.*

Comentarios:

8. ¿Se corrigen los defectos encontrados en el *Manual de Operación* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.*

Comentarios:

9. ¿Se realizan pruebas del sistema siguiendo el *Plan de Pruebas de Sistema* definido en la fase de requisitos?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Plan de Pruebas de Sistema identifica las pruebas requeridas para el cumplimiento de los requisitos especificados al inicio del ciclo/proyecto. Una práctica común para obtener el Plan de Pruebas de Sistema es hacer que el Cliente esté presente.*

Comentarios:

10. ¿Son documentados los resultados de las pruebas en un *Reporte de Pruebas de Sistema*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Reporte de Pruebas de Sistema registra a los participantes, la fecha y lugar en que se realizan las pruebas, la duración y los defectos encontrados.*

Comentarios:

11. ¿Se corrigen los defectos encontrados en las pruebas del sistema con base en el *Reporte de Pruebas de Sistema* y se obtiene la aprobación de las correcciones?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

12. ¿Se documenta el *Manual de Usuario* o bien se modifica uno ya existente?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Manual de Usuario es un documento electrónico o impreso que describe la forma de uso del Software con base a la interfaz del usuario. Este manual deberá ser redactado en términos comprensibles a los usuarios.*

Comentarios:

13. ¿Se verifica el *Manual de Usuario* antes de ser aprobado?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La verificación consiste en asegurar la consistencia del Manual de Usuario con el Software y con el estándar de documentación requerido en el Proceso Específico. Los defectos encontrados se documentan en un Reporte de Verificación.*

Comentarios:

14. ¿Se corrigen los defectos encontrados en el *Manual de Usuario* con base en el *Reporte de Verificación* y se obtiene la aprobación de las correcciones?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Reporte de Verificación registra a los participantes, la fecha y lugar en que se realiza la verificación, la duración y los defectos encontrados.*

Comentarios:

15. Una vez que ya han sido verificados los documentos, ¿se incorporan el *Software*, el *Reporte de Pruebas de Integración*, el *Registro de Rastreo*, el *Manual de Operación* y el *Manual de Usuario* como líneas base a la *Configuración de Software*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*La Configuración de Software consiste de todos los productos de software generados con el Desarrollo y Mantenimiento de Software: Especificación de Requisitos, Análisis y Diseño, Software, Registro de Rastreo, Plan de Pruebas de Sistema, Reporte de Pruebas dl Sistema, Plan de Pruebas de Integración, Reporte de Pruebas de Integración, Manual de Usuario, Manual de Operación, Manual de Mantenimiento. La Configuración de Software suele contener referencias a todos los documentos anteriores de forma que facilite su ubicación para consulta.*

Comentarios:

16. ¿Se elabora el *Reporte de Actividades* registrando las actividades realizadas, fechas de inicio y fin, y responsable por actividad?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

Comentarios:

**REALIZACIÓN DE LA FASE DE CIERRE (A.6)**

VI. Durante la fase de cierre se realiza la integración final de la *Configuración de Software* generada en las fases para su entrega. Si fuera el caso, se identifican y documentan las *Lecciones Aprendidas* y se genera el *Reporte de Mediciones y Sugerencias de Mejora*.

1. ¿Se genera o actualiza el *Manual de Mantenimiento* del *Software*?

Siempre	Usualmente	Algunas Veces	Rara Vez	Nunca	No Sabe	No Aplica
<input type="checkbox"/>						

*El Manual de Mantenimiento es un documento electrónico o impreso que describe la Configuración de Software y el entorno usado para el desarrollo y realización de pruebas (compiladores, herramientas de análisis y diseño, construcción y pruebas). Este manual deberá ser redactado en términos comprensibles para el personal de mantenimiento.*

Comentarios:



## **8. Anexo C.- Activos para los procesos de DMS y APE**



## Plantilla para la Minuta de Lanzamiento: DMS Actividad A.1.1

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
<b>Objetivo de la reunión</b>	El objetivo de la reunión es obtener la minuta de revisión después de haber analizado con los miembros del <i>Equipo de Trabajo</i> el <i>Plan de Desarrollo</i> actual para lograr un entendimiento común y obtener su compromiso con el proyecto.				
<b>Participantes de la reunión</b>	RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos. RDM como responsable del proceso de Desarrollo y Mantenimiento de Software. ET como participantes del proceso de Desarrollo y Mantenimiento de Software.				
<b>Seguimiento de compromisos pendientes</b>	[Inserte aquí el texto]				
<b>Puntos tratados</b>	[Inserte aquí el texto]				
<b>Acuerdos obtenidos</b>	[Inserte aquí el texto]				
<b>Asuntos pendientes por resolver</b>	[Inserte aquí el texto]				
<b>Aprobación</b>	Los presentes aseguran haber entendido el proyecto y se comprometen a realizarlo con profesionalismo y responsabilidad de acuerdo al Plan de Desarrollo revisado.  [Inserte aquí el texto]				

## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha</b>	Fecha en la que se realiza la reunión.
<b>Hora de inicio</b>	Hora exacta en la que comienza la reunión.
<b>Hora de fin</b>	Hora exacta en la que termina la reunión.
<b>Objetivo de la reunión</b>	El objetivo de la reunión para el caso de la revisión del Plan de Desarrollo y el consecuente lanzamiento del proyecto ya se establece en el documento. Si fuera necesario, este objetivo puede modificarse para efectos de un mejor entendimiento.
<b>Participantes de la reunión</b>	Los roles de los participantes en la reunión y que han sido definidos como Equipo de Trabajo en el Plan de Desarrollo. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
<b>Seguimiento de compromisos pendientes</b>	Es responsabilidad del RDM documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
<b>Puntos tratados</b>	El RDM debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo: <ol style="list-style-type: none"> <li>1. Pase de lista.</li> <li>2. Revisión del Plan de Desarrollo.</li> <li>3. Ruegos y preguntas.</li> <li>4. Obtención de compromiso del ET.</li> <li>5. Firma de minuta.</li> <li>6. Cierre de la reunión.</li> </ol>
<b>Acuerdos obtenidos</b>	El RDM debe documentar con una lista los acuerdos finales o compromisos, estableciendo que los responsables y fechas que fueron establecidos por el RAPE en el Plan de Desarrollo son aceptados.
<b>Asuntos pendientes por resolver</b>	El RDM debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY o al RAPE) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
<b>Aprobación</b>	El RDM debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.

### Plantilla para Reporte de Actividades: DMS Actividad A.1.3

Reporte de actividades No. [Inserte aquí el texto]			
Datos generales			
Nombre del proyecto	[Inserte aquí el texto]		
Referencia del proyecto	[Inserte aquí el texto]		
Nombre del RDM	[Inserte aquí el texto]		
Fecha del reporte	[Fecha]	Período comprendido	[Fecha] al [Fecha]

Registro de actividades							
Actividad del Plan de Desarrollo	Fecha de inicio		Fecha de fin		Esfuerzo (horas)		Responsable
	Planeado	Real	Planeado	Real	Planeado	Real	
[Inserte aquí el texto]	[Fecha]	[Fecha]	[Fecha]	[Fecha]	[Fecha]	[Fecha]	[Inserte aquí el texto]
<b>Tiempo total de producción</b>					[Rev]	[Rev]	

Defectos encontrados			
Productos verificados	Registro de tiempos (esfuerzo en horas)		Número de defectos
	En verificación	En corrección	
Minuta de Lanzamiento	[Rev]	[Rev]	[Inserte aquí el texto]
Necesidades del Sistema	[Rev]	[Rev]	[Inserte aquí el texto]
Especificación de Requerimientos	[Rev]	[Rev]	[Inserte aquí el texto]
Plan de Pruebas de Sistema	[Rev]	[Rev]	[Inserte aquí el texto]
Manual de Usuario	[Rev]	[Rev]	[Inserte aquí el texto]
Registro de Rastreo	[Rev]	[Rev]	[Inserte aquí el texto]
Plan de Pruebas de Integración	[Rev]	[Rev]	[Inserte aquí el texto]
Manual de Operación	[Rev]	[Rev]	[Inserte aquí el texto]
Manual de Usuario	[Rev]	[Rev]	[Inserte aquí el texto]
Manual de Mantenimiento	[Rev]	[Rev]	[Inserte aquí el texto]



## Instrucciones de llenado de la plantilla

Campo	Descripción
<i>Datos generales</i>	
<b>Nombre del proyecto</b>	Nombre que le fue asignado al proyecto en el Plan de Desarrollo.
<b>Referencia del proyecto</b>	Referencia única que se le da a cada proyecto para referirse a él de forma abreviada. Estas referencias suelen tener identificadores que ayudan al personal a determinar la naturaleza del proyecto sin consultar el nombre o descripción completa del mismo. El estilo de las referencias debe ser definido por la empresa y debe ser respetado por todo el Equipo de Trabajo.
<b>Nombre del RDM</b>	Nombre del Responsable del Desarrollo y Mantenimiento de Software que fue asignado al proyecto en cuestión.
<b>Fecha de reporte</b>	Fecha en la que se está entregando el reporte de avance. Es obvio que cada modificación de avance requerirá la modificación del reporte existente y por ende la fecha será actualizada.
<b>Periodo comprendido</b>	Periodo de tiempo que se está reportando. Por ejemplo, la Fecha del reporte puede ser 08/06/2013 y se están reportando las actividades que se realizaron entre el 01/04/2013 y el 07/06/2013.
<i>Registro de actividades</i>	
<b>Actividad del plan de desarrollo</b>	Las actividades listadas en esta columna deben corresponder con las actividades que hayan sido registradas en el Plan de Desarrollo.
<b>Fecha de inicio – Planeado</b>	Fecha planeada por el RAPE para iniciar la actividad correspondiente.
<b>Fecha de inicio – Real</b>	Fecha real en que inició la actividad correspondiente.
<b>Fecha de fin – Planeado</b>	Fecha planeada por el RAPE para finalizar la actividad correspondiente.
<b>Fecha de fin – Real</b>	Fecha real en que finalizó la actividad correspondiente.
<b>Esfuerzo (horas) – Planeado</b>	Horas planeadas por el RAPE para realizar la actividad correspondiente.
<b>Esfuerzo (horas) – Real</b>	Horas reales para realizar la actividad correspondiente.
<b>Responsable</b>	El responsable de cada actividad tomada del Plan de Desarrollo y que acordó comprometerse a realizarla en tiempo y forma a través de la Minuta de Lanzamiento.
<b>Tiempo total de producción</b>	Sumatoria de todos los campos registrados para cada actividad en Esfuerzo (horas) – Planeado y Esfuerzo (horas) – Real, respectivamente.
<i>Defectos encontrados</i>	
<b>Registro de tiempos (esfuerzo en horas) – En verificación</b>	Tiempo total (en horas) que el responsable de la verificación tardó en revisar el producto en cuestión. Estos valores serán registrados por el RDM asignado al proyecto quien deberá revisar los Reportes de Actividades para obtener el valor exacto.
<b>Registro de tiempos (esfuerzo en horas) – En corrección</b>	Tiempo total (en horas) en que los defectos encontrados fueron corregidos. Estos valores serán registrados por el RDM asignado al proyecto quien deberá revisar los Reportes de Actividades para obtener el valor exacto.
<b>Número de defectos</b>	Total de defectos encontrados por cada producto verificado.
<b>Registro de tiempos (esfuerzo en horas) – En validación</b>	Tiempo total (en horas) que el responsable de la validación tardó en revisar el producto en cuestión. Estos valores serán registrados por el RDM asignado al proyecto quien deberá revisar los Reportes de Actividades para obtener el valor exacto.
<b>Registro de tiempos (esfuerzo en horas) – En corrección</b>	Tiempo total (en horas) en que los defectos encontrados fueron corregidos. Estos valores serán registrados por el RDM asignado al proyecto quien deberá revisar los Reportes de Actividades para obtener el valor exacto.
<b>Número de defectos</b>	Total de defectos encontrados por cada producto validado.
<b>Productos probados</b>	Se deberán listar todos los prototipos, módulos, componentes, etc., que los PR someten a pruebas unitarias o pruebas de integración.
<b>Registro de tiempos (esfuerzo en horas) – En prueba</b>	Tiempo total (en horas) que el PR tardó en probar el producto en cuestión. Estos valores serán registrados por el RDM asignado al proyecto quien deberá revisar los Reportes de Actividades para obtener el valor exacto.
<b>Registro de tiempos (esfuerzo en horas) – En corrección</b>	Tiempo total (en horas) en que los defectos encontrados fueron corregidos. Estos valores serán registrados por el RDM asignado al proyecto quien deberá revisar los Reportes de Actividades para obtener el valor exacto.

<b>Número de defectos</b>	Total de defectos encontrados por cada producto probado.
<b>Tamaño de los productos</b>	
<b>Producto</b>	Se deberán listar todos los productos generados en el proceso de DMS. Por ejemplo: Casos de Uso, Especificación de Requerimientos, Análisis y Diseño, Componente de Software, Manual de Usuario, etc.
<b>Tamaño real</b>	Se deberá registrar el tamaño real (es decir, cuando el producto ya esté terminado) para cada producto listado anteriormente. Por ejemplo, para los Casos de Uso se podría registrar: # de casos de uso y # de páginas; para la Especificación de Requerimientos: # de requerimientos y # de páginas de la Especificación de Requerimientos; para el Análisis y Diseño: # de diagramas de clases, # de tablas de la BD, # de páginas del Análisis y Diseño; para el Componente de Software: LOC (Líneas de Código); y para el Manual de Usuario: # de páginas del Manual de Usuario.

## Plantilla para Necesidades del Sistema: DMS Actividad A.2.3

### 1. Alcance

[Inserte aquí el texto]

*El contenido de este apartado debe proporcionar una breve introducción del documento, su relación con otros posibles documentos, su finalidad y los destinatarios en relación a la información que contiene.*

#### 1.1. Identificación

[Inserte aquí el texto]

*Identificación del sistema, proporcionando su nombre y abreviatura (si procede).*

#### 1.2. Visión general del documento

[Inserte aquí el texto]

*Explicación del propósito, audiencia, y consideraciones de seguridad o privacidad del documento.*

- *Generalmente, el propósito del documento suele ser uno de los siguientes:*
- *Comunicar las necesidades y expectativas del cliente.*
- *Comunicar el entendimiento del proyecto.*
- *Obtener acuerdos entre las partes implicadas (personal representante de la parte cliente, personal del equipo de desarrollo, etc.).*

#### 1.3. Visión general del sistema

[Inserte aquí el texto]

*Resumen del propósito del sistema o subsistema propuesto (al cual se aplica la descripción del sistema).*

#### 1.4. Personal involucrado

<b>Nombre</b>	[Inserte aquí el texto]
<b>Rol</b>	[Inserte aquí el texto]
<b>Categoría profesional</b>	[Inserte aquí el texto]
<b>Responsabilidades</b>	[Inserte aquí el texto]
<b>Información de contacto</b>	[Inserte aquí el texto]
<b>Aprobación</b>	[Inserte aquí el texto]

*Relación de personas involucradas en el desarrollo del sistema, con información de contacto. Esta información es útil para que el RDM pueda localizar a todos los participantes y recabar la información necesaria para la obtención de requerimientos, validaciones de seguimiento, etc.*

### 2. Documentos referenciados

Nº	Título	Ruta	Versión	Fecha	Autor
	[Título]	[Ruta]	[Rev]	[Fecha]	[Autor]

*Relación de los documentos a los que se hace referencia en las necesidades del sistema, indicando, según proceda: el nombre del manual, documento, código del documento, título, ruta, revisión, fecha y origen.*

### 3. Situación actual

[Inserte aquí el texto]

*Cuando el sistema propuesto tenga como finalidad la sustitución o ampliación de un sistema o situación existente, este apartado contendrá una descripción suficiente del actual (aunque éste último sea un sistema manual).*

#### 3.1. Antecedentes

[Inserte aquí el texto]

*Visión general del sistema o situación actual, incluyendo: misión, objetivos, alcance, modo de uso o aplicación, etc.*

#### 3.2. Políticas y restricciones operacionales

[Inserte aquí el texto]

*Relación de políticas o restricciones de cualquier índole impuestas sobre el sistema o situación actual. Algunos ejemplos pueden ser:*

- *Restricciones sobre el número de usuarios capaces de utilizar el sistema.*
- *Restricciones de hardware (uso obligado de determinadas plataformas, redes telemáticas, etc.)*
- *Restricciones de seguridad o relativas a protección de datos.*
- *Restricciones de software (uso obligado de una determinada BD, Sistema Operativo, etc.)*
- *Restricciones de recursos operacionales como espacio físico.*

#### 3.3. Descripción del sistema o situación actual

[Inserte aquí el texto]

*Descripción detallada del sistema o situación actual y de su funcionamiento, incluyendo los siguientes subapartados:*

- *Enumeración y descripción de funciones, características y capacidades del sistema o situación actual.*
- *En el caso de considerarse apropiado puede aportarse información adicional como:*
  - *Diagramas, flujos y procesos con un nivel de detalle suficiente para comprender la función o un conjunto de funciones del sistema o situación actual.*
  - *Interacción entre componentes del sistema.*
- *Descripción del entorno de operación y sus características.*
- *Interacción del sistema con otros sistemas externos.*
- *Características de rendimiento como velocidad, rendimiento de trabajo, volumen, frecuencia, y demás.*
- *Atributos de calidad como disponibilidad, eficiencia, flexibilidad, portabilidad, reusabilidad, usabilidad, y demás.*
- *Provisiones de seguridad, emergencia, privacidad y continuidad de las operaciones en circunstancias de emergencia.*
- *Otro tipo de información relevante en la descripción del sistema como: factores de riesgo, coste de las operaciones, y demás.*

*Es importante que la descripción del sistema o situación actual sea lo más simple y clara posible, para que todos los lectores del documento puedan entenderla completamente. Es importante que ésta se realice usando la terminología del usuario.*

### 3.4. Tipos de usuarios

<b>Tipo de usuario</b>	[Inserte aquí el texto]
<b>Responsabilidad</b>	[Inserte aquí el texto]
<b>Formación</b>	[Inserte aquí el texto]
<b>Habilidades</b>	[Inserte aquí el texto]
<b>Actividades</b>	[Inserte aquí el texto]
<b>Interacción con el sistema</b>	[Inserte aquí el texto]

*Relación de los tipos de usuario. Un tipo de usuario se distingue por el modo en el cual interactúa con el sistema. Factores como la responsabilidad, habilidades, competencias, etc., distinguen a los distintos tipos de usuarios.*

### 3.5 Mantenimiento / soporte

[Inserte aquí el texto]

*Descripción de las necesidades de mantenimiento, reparación, almacenamiento, distribución, sistemas de copias de seguridad, sistemas de emergencia, etc.*

### 3.6. Necesidad y naturaleza de los cambios

[Inserte aquí el texto]

*Descripción de las carencias, defectos o debilidades del sistema o situación actual y que motivan al desarrollo de un nuevo sistema o, a una modificación del existente.*

*De no existir un sistema anterior (ni tan siquiera manual), en este apartado y subapartados deben quedar reflejadas las justificaciones que llevan al cambio.*

### 3.7. Descripción de los cambios deseados

[Inserte aquí el texto]

*Enumeración de las capacidades, funciones, procesos, etc. que deben generarse o modificarse para satisfacer las nuevas necesidades.*

*Los cambios deben basarse en el sistema descrito en el punto 3.3. Si no existe un sistema anterior, en este apartado se enumeran las capacidades requeridas del nuevo sistema.*

*De forma apropiada podrá hacerse referencia a:*

- *Cambios en la capacidad. Descripción de las funciones y características que deben añadirse, eliminarse o modificarse para conseguir los objetivos y requisitos del nuevo sistema.*
- *Cambios en el proceso del sistema. Descripción de los cambios en el proceso o procesos de transformación de datos que darán lugar a nuevos resultados con los mismos datos, mismos resultados con nuevos datos o ambos.*
- *Cambios de interfaces. Descripción de los cambios en el sistema que provocan cambios en los interfaces y cambios en los interfaces que causan cambios en el sistema.*
- *Cambios de personal. Descripción de cambios en personal causados por nuevos requisitos, cambios en tipos de usuarios o ambos.*
- *Cambios de entorno. Descripción de cambios en el entorno operacional que causan cambios en las funciones, procesos, interfaces, o personal del sistema y/o cambios que deben realizarse en el entorno por causa de cambios de las funciones, procesos, interfaces, o personal del sistema.*
- *Cambios operacionales. Descripción de cambios a procedimientos, métodos, rutinas de trabajo del usuario causados por los cambios mencionados.*

- *Cambios de soporte. Cambios en los requisitos de soporte causados por cambios en las funciones, procesos, interfaces o personal del sistema y/o cambios en las funciones, procesos, interfaces o personal del sistema causados por cambios de soporte.*
- *Otros cambios. Descripción de otros cambios que afectarán a los usuarios.*

*Es necesario identificar las prioridades entre los cambios deseados y las nuevas características. Esta identificación puede realizarse clasificando cada uno de los cambios como esencial, deseado u opcional. Esta información será útil en la toma de decisiones durante el desarrollo y en el caso de interrupciones o invasiones de agendas o presupuestos.*

#### **4. Sistema propuesto**

[Inserte aquí el texto]

*La descripción del sistema propuesto debe ceñirse a los conceptos básicos que indican las nuevas capacidades operacionales sin entrar en especificaciones de diseño, a no ser que sean restricciones impuestas.*

##### **4.1. Antecedentes**

[Inserte aquí el texto]

*Visión general del sistema propuesto, incluyendo: misión, objetivos, alcance, modo de uso o aplicación, etc.*

##### **4.2. Políticas y restricciones operacionales**

[Inserte aquí el texto]

*Relación de políticas o restricciones de cualquier índole aplicables al sistema propuesto. Algunos ejemplos pueden ser:*

- *Restricciones sobre el número de usuarios capaces de utilizar el sistema.*
- *Restricciones de hardware (uso obligado de determinadas plataformas, redes telemáticas, etc.)*
- *Restricciones de seguridad o relativas a protección de datos.*
- *Restricciones de software (uso obligado de una determinada BD, Sistema Operativo, etc.)*
- *Restricciones de recursos operacionales como espacio físico.*

##### **4.3. Descripción del sistema propuesto**

[Inserte aquí el texto]

*Descripción detallada del sistema propuesto y de su funcionamiento, incluyendo los siguientes subapartados:*

- *Enumeración y descripción de funciones, características y capacidades del sistema propuesto.*
- *Deben identificarse las prioridades entre funcionalidades y características. Esta identificación puede realizarse clasificando cada una las funciones, características y capacidades como esencial, deseado u opcional.*
- *Deben incluirse en la medida de lo posible:*
  - *Diagramas, flujos y procesos con un nivel de detalle suficiente para comprender la función o un conjunto de funciones del sistema propuesto desde el punto de vista del usuario.*
  - *Identificación de interacciones entre componentes del sistema.*

- *Descripción del entorno de operación y sus características.*
- *Interacción del sistema con otros sistemas.*
- *Características de rendimiento como velocidad, rendimiento de trabajo, volumen, frecuencia, etc.*
- *Atributos de calidad como disponibilidad, eficiencia, flexibilidad, portabilidad, reusabilidad, usabilidad, etc.*
- *Provisiones de seguridad, emergencia, privacidad y continuidad de las operaciones en circunstancias de emergencia.*
- *Otro tipo de información relevante en la descripción del sistema como: factores de riesgo, costo de las operaciones, etc.*

*Es importante que las funcionalidades del sistema propuesto sean lo más simples y claras posibles, para que todos los lectores del documento puedan entenderlos completamente. Debe realizarse usando la terminología del usuario.*

*Nota: El RA debe organizar la información en este apartado como considere apropiado para el sistema o situación. Aquellas partes de la descripción que sean extensas podrán incluirse como apéndices del documento.*

#### **4.4. Tipos de usuarios**

<b>Tipo de usuario</b>	[Inserte aquí el texto]
<b>Responsabilidad</b>	[Inserte aquí el texto]
<b>Formación</b>	[Inserte aquí el texto]
<b>Habilidades</b>	[Inserte aquí el texto]
<b>Actividades</b>	[Inserte aquí el texto]
<b>Interacción con el sistema</b>	[Inserte aquí el texto]

*Relación de los tipos de usuario. Un tipo de usuario se distingue por el modo en el cual un usuario interactúa con el sistema. Factores como la responsabilidad, habilidades, competencias, etc. distinguen a los distintos tipos de usuarios.*

#### **4.5. Mantenimiento / soporte**

[Inserte aquí el texto]

*Descripción de las necesidades de mantenimiento, reparación, almacenamiento, distribución, sistemas de copias de seguridad, sistemas de emergencia, etc.*

#### **4.6. Escenarios operacionales**

[Inserte aquí el texto]

*Relación de escenarios operacionales. Un escenario es una descripción paso a paso de cómo el sistema propuesto debe operar e interactuar con sus usuarios y sus interfaces externas, bajo unas determinadas circunstancias.*

*Los escenarios operacionales proporcionan información sobre cómo interactúan todas las partes del sistema propuesto, los usuarios y otras entidades externas. También pueden utilizarse para describir lo que no debe hacer el sistema.*

#### **4.7. Futuras evoluciones**

[Inserte aquí el texto]

*Descripción de las previsiones de evolución que se tienen previstas para el sistema (si se tiene).*

*Esta información resulta útil para el personal involucrado en el desarrollo del nuevo sistema, y podrá preparar la arquitectura del sistema para asimilar cambios futuros con el menor impacto.*

#### **4.8. Cambios considerados pero no incluidos**

[Inserte aquí el texto]

*Identificación de cambios considerados pero no incluidos en el sistema propuesto y el motivo por el que no han sido incluidos.*

#### **5. Resumen de mejoras**

[Inserte aquí el texto]

*Resumen de los beneficios proporcionados por el sistema propuesto. Este resumen puede incluir las siguientes secciones:*

- *Nuevas características o funcionalidades.*
- *Actualización de características existentes.*
- *Eliminación de características obsoletas, no útiles o confusas.*
- *Mejoras de rendimiento como tiempo de respuesta, mejoras de calidad, etc.*

#### **6. Información adicional**

[Inserte aquí el texto]

*Cualquier información adicional que facilite la comprensión del documento en sí.*



## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha del reporte</b>	Fecha en la que tanto el RE y el RAN empiezan a realizar la verificación del documento. Es importante considerar que cada verificación realizada genera un Reporte de Verificación si es necesario.
<b>Lugar</b>	Lugar en donde el RE y el RAN realizan la verificación del documento.
<b>Hora de inicio</b>	Hora exacta en la que comienza la verificación del documento.
<b>Hora de fin</b>	Hora exacta en la que termina la verificación del documento.
<b>Duración total</b>	Cálculo numérico (en horas) para expresar el tiempo total que RAN y RE tardaron en verificar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RAN como RE hayan tenido interrupciones durante la verificación y esta relación no sea del todo cierta.
<b>Nombre completo</b>	Nombres del RAN y RE que participan en la verificación.
<b>Puesto (rol)</b>	Puestos o roles de los participantes en la verificación.
<b>Elemento a verificar</b>	Se deberán listar todos los elementos que se deban verificar dado lo definido en los Criterios de Verificación de la documentación de procesos. Por ejemplo: Claridad de redacción, Consistencia o adherencia con la Descripción del producto, Conformidad con el estándar de documentación. Se recomienda incluir todos los elementos que forman el documento de las Necesidades del Cliente y aplicar los criterios definidos.
<b>Descripción del defecto encontrado</b>	El RAN y RE deben describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento está redactado claramente”. Un ejemplo de cómo describir un defecto puede ser: “Las necesidades no son consistentes con la descripción del producto dado que solo considera uno de los objetivos del producto y no los dos.”
<b>Fecha de aprobación</b>	Fecha en la que RAN y RE se aseguran que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.

## Plantilla para Especificación de Requerimientos: DMS Actividad A.2.6

### 1. Introducción

[Inserte aquí el texto]

*La introducción de la Especificación de Requerimientos (SRS) debe proporcionar una visión general de la misma. Debe incluir el propósito, el alcance, las definiciones y acrónimos, y las referencias, de acuerdo al proyecto.*

#### 1.1. Propósito

[Inserte aquí el texto]

- *Propósito del documento.*
- *Audiencia a la que va dirigido.*

#### 1.2. Alcance

[Inserte aquí el texto]

- *Identificación del producto(s) a desarrollar mediante el nombre asignado en las Necesidades del Sistema o bien la propuesta de uno diferente.*
- *Consistencia con definiciones similares de documentos de mayor nivel (ej. Descripción del sistema) que puedan existir.*

#### 1.3. Personal involucrado

<b>Nombre</b>	[Inserte aquí el texto]
<b>Rol</b>	[Inserte aquí el texto]
<b>Categoría profesional</b>	[Inserte aquí el texto]
<b>Responsabilidades</b>	[Inserte aquí el texto]
<b>Información de contacto</b>	[Inserte aquí el texto]
<b>Aprobación</b>	[Inserte aquí el texto]

*Relación de personas involucradas en el desarrollo del sistema, con información de contacto. Esta información es útil para que el RDM pueda localizar a todos los participantes y recabar la información necesaria para la obtención de requerimientos, validaciones de seguimiento, etc.*

#### 1.4. Definiciones, acrónimos y abreviaturas

[Inserte aquí el texto]

*Definición de todos los términos, abreviaturas y acrónimos necesarios para interpretar apropiadamente este documento. En ella se pueden indicar referencias a uno o más apéndices, o a otros documentos.*

#### 1.5. Referencias

Referencia	Título	Ruta	Fecha	Autor
[Ref.]	[Título]	[Ruta]	[Fecha]	[Autor]

*Relación completa de todos los documentos relacionados en la especificación de requerimientos, identificando de cada documento el título, referencia (si procede), fecha y organización que lo proporciona.*

## 1.6. Resumen del documento

[Inserte aquí el texto]

- *Descripción del contenido del resto del documento.*
- *Explicación de la organización del documento.*

## 2. Descripción general

### 2.1. Perspectiva del producto

[Inserte aquí el texto]

*Indicar si es un producto independiente o parte de un sistema mayor. En el caso de tratarse de un producto que forma parte de un sistema mayor, el uso de un diagrama que sitúe al producto dentro del sistema e identifique sus conexiones facilita la comprensión.*

### 2.2. Funcionalidad del producto

[Inserte aquí el texto]

*Resumen de las funcionalidades principales que el producto debe realizar, sin entrar en detalle. En ocasiones la información de esta sección puede tomarse de un documento de mayor nivel (ej. Necesidades del Sistema).*

*Las funcionalidades deben estar organizadas de manera que el cliente o cualquier interlocutor puedan entenderlo perfectamente. Para ello se pueden utilizar métodos textuales o gráficos.*

### 2.3. Características de los usuarios

<b>Tipo de usuario</b>	[Inserte aquí el texto]
<b>Formación</b>	[Inserte aquí el texto]
<b>Habilidades</b>	[Inserte aquí el texto]
<b>Actividades</b>	[Inserte aquí el texto]

*Descripción de los usuarios del producto, incluyendo nivel educacional, experiencia y experiencia técnica.*

### 2.4. Restricciones

[Inserte aquí el texto]

*Descripción de aquellas limitaciones a tener en cuenta a la hora de diseñar y desarrollar el sistema, tales como el empleo de determinadas metodologías de desarrollo, lenguajes de programación, normas particulares, restricciones de hardware, de sistema operativo etc.*

### 2.5. Suposiciones y dependencias

[Inserte aquí el texto]

*Descripción de aquellos factores que, si cambian, pueden afectar a los requerimientos. Por ejemplo, una suposición puede ser que determinado sistema operativo está disponible para el hardware requerido. De hecho, si el sistema operativo no estuviera disponible, la Especificación de Requerimientos debería modificarse.*

### 2.6. Evolución previsible del sistema

[Inserte aquí el texto]

*Identificación de futuras mejoras al sistema, que podrán analizarse e implementarse en un futuro.*

## 2.7. Diagrama de contexto de casos de uso para el sistema

[Inserte aquí el texto]

Este diagrama de casos de uso debe ser general y debe resumir las funcionalidades identificadas en las Necesidades del Sistema. Los Analistas deben partir del documento de Necesidades del Sistema para resumir con un diagrama de casos de uso las funcionalidades requeridas (y confirmadas) por el Cliente.

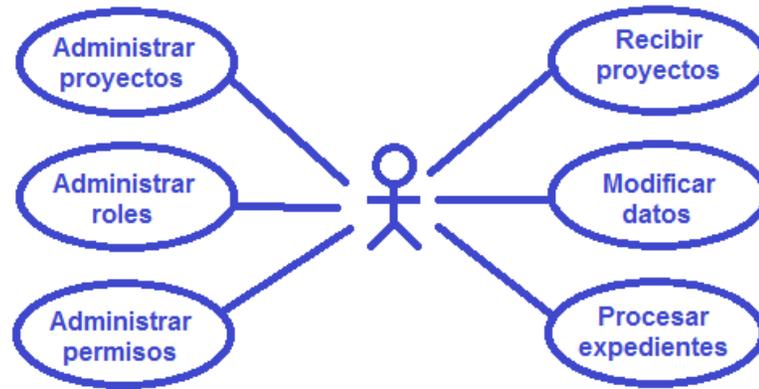


Figura 1. Ejemplo de diagrama de contexto de casos de uso

## 3. Requerimientos funcionales

Esta es la sección más extensa y más importante del documento. Debe contener una lista detallada y completa de los requerimientos que debe cumplir el sistema a desarrollar. El nivel de detalle de los requerimientos debe ser el suficiente para que los Diseñadores puedan diseñar un sistema que los satisfaga y los encargados de las pruebas (Programadores) puedan determinar si éstos se satisfacen.

Los requerimientos se dispondrán en forma de listas numeradas para su identificación, seguimiento, trazabilidad y validación (RF 10.1, RF 10.2, etc.). Es muy importante tener presente que el primer número después de RF corresponderá a cada funcionalidad identificada en el diagrama de contexto de casos de uso para el sistema. El segundo número corresponderá al número de requerimiento en sí.

Para cada requerimiento debe rellenarse la siguiente tabla, obviamente deberán ser agregadas tantas tablas como número de requerimientos que se desee especificar, por ejemplo:

<b>Número y nombre del requerimiento</b>	RF 1.12 Modificación de contraseña de usuario
<b>Origen del requerimiento</b>	Usuario final
<b>Prioridad del requerimiento</b>	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
<b>Descripción</b>	Este requerimiento indica que el sistema podrá permitir la modificación de la contraseña del usuario.
<b>Entrada</b>	Se mostrará el nombre del usuario y se requerirán los siguientes datos: Contraseña (ob), Nueva Contraseña (ob), Repetir nueva contraseña (ob).
<b>Salida</b>	Se modifica y registra la nueva contraseña con la que el usuario accede al sistema. Se mostrarán mensajes de error en caso de que la modificación haya sido fallida, bien por problemas sintácticos, por falta de datos, o por problemas de autenticación con la antigua contraseña.
<b>Proceso</b>	El sistema muestra la pantalla para introducir los datos requeridos. Una vez rellenos los campos, y mediante una opción de modificación se

	procederá a la introducción de estos datos en la BD del sistema. Se dispondrá de una opción de borrado que permita hacer correcciones antes del envío de datos. <i>Se mostrará al usuario una pantalla con los nuevos datos con los que ha sido registrado.</i>		
Referencias	Con otros requerimientos	RF 1.10, RF 1.11	
	Con casos de uso	Gestión de Usuarios	
Fecha de creación	12/03/2013	Último cambio	06/05/2013

Número y nombre de requerimiento	[Inserte aquí el texto]		
Origen del requerimiento	[Inserte aquí el texto]		
Prioridad del requerimiento	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional		
Descripción	[Inserte aquí el texto]		
Entrada	[Inserte aquí el texto]		
Salida	[Inserte aquí el texto]		
Proceso	[Inserte aquí el texto]		
Referencias	Con otros requerimientos	[Inserte aquí el texto]	
	Con casos de uso	[Inserte aquí el texto]	
Fecha de creación	[Fecha]	Último cambio	[Fecha]

Número y nombre de requerimiento	[Inserte aquí el texto]		
Origen del requerimiento	[Inserte aquí el texto]		
Prioridad del requerimiento	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional		
Descripción	[Inserte aquí el texto]		
Entrada	[Inserte aquí el texto]		
Salida	[Inserte aquí el texto]		
Proceso	[Inserte aquí el texto]		
Referencias	Con otros requerimientos	[Inserte aquí el texto]	
	Con casos de uso	[Inserte aquí el texto]	
Fecha de creación	[Fecha]	Último cambio	[Fecha]

Número y nombre de requerimiento	[Inserte aquí el texto]		
Origen del requerimiento	[Inserte aquí el texto]		
Prioridad del requerimiento	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional		
Descripción	[Inserte aquí el texto]		
Entrada	[Inserte aquí el texto]		
Salida	[Inserte aquí el texto]		
Proceso	[Inserte aquí el texto]		
Referencias	Con otros requerimientos	[Inserte aquí el texto]	
	Con casos de uso	[Inserte aquí el texto]	
Fecha de creación	[Fecha]	Último cambio	[Fecha]

Número y nombre de requerimiento	[Inserte aquí el texto]		
Origen del requerimiento	[Inserte aquí el texto]		
Prioridad del requerimiento	<input type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional		

<b>Descripción</b>	[Inserte aquí el texto]		
<b>Entrada</b>	[Inserte aquí el texto]		
<b>Salida</b>	[Inserte aquí el texto]		
<b>Proceso</b>	[Inserte aquí el texto]		
<b>Referencias</b>	<b>Con otros requerimientos</b>	[Inserte aquí el texto]	
	<b>Con casos de uso</b>	[Inserte aquí el texto]	
<b>Fecha de creación</b>	[Fecha]	<b>Último cambio</b>	[Fecha]

<b>Número y nombre de requerimiento</b>	[Inserte aquí el texto]		
<b>Origen del requerimiento</b>	[Inserte aquí el texto]		
<b>Prioridad del requerimiento</b>	<input type="checkbox"/> Alta/Eencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/ Opcional
<b>Descripción</b>	[Inserte aquí el texto]		
<b>Entrada</b>	[Inserte aquí el texto]		
<b>Salida</b>	[Inserte aquí el texto]		
<b>Proceso</b>	[Inserte aquí el texto]		
<b>Referencias</b>	<b>Con otros requerimientos</b>	[Inserte aquí el texto]	
	<b>Con casos de uso</b>	[Inserte aquí el texto]	
<b>Fecha de creación</b>	[Fecha]	<b>Último cambio</b>	[Fecha]

### 3.1. Requerimientos de las interfaces

[Inserte aquí el texto]

*Descripción detallada de todas las entradas y salidas del sistema de software.*

#### 3.1.1. Interfaces de usuario

[Inserte aquí el texto]

*Describir los requerimientos de la interfaz de usuario para el producto. Esto puede estar en la forma de descripciones del texto o pantallas de interfaz. Por ejemplo, posiblemente el cliente ha especificado el estilo y los colores de su producto. En este sentido, el AN debe describir exactamente cómo el producto aparecerá a su usuario previsto.*

<b>Descripción</b>	El producto debe ser fácil de usar cuando se trate de un usuario sin entrenamiento previo.
<b>Justificación</b>	El cliente indica que los usuarios potenciales pueden nunca antes haber usado este tipo de productos.

<b>Descripción</b>	[Inserte aquí el texto]
<b>Justificación</b>	[Inserte aquí el texto]

<b>Descripción</b>	[Inserte aquí el texto]
<b>Justificación</b>	[Inserte aquí el texto]

#### 3.1.2. Interfaces de hardware

[Inserte aquí el texto]

*El AN debe especificar las características lógicas para cada interfaz entre el producto y los componentes de hardware del sistema si éste fuera el caso. Se incluirán características de configuración. Por ejemplo:*

<b>Descripción</b>	El producto debe obtener la información de llegada a la oficina de los trabajadores de un dispositivo lector de huellas digitales.
--------------------	--

<b>Justificación</b>	Es importante que para registrar la entrada al trabajo se tenga un control automático y se reduzca el tiempo de retardos. El sistema lector debe ser configurado de acuerdo a las especificaciones técnicas del fabricante.
----------------------	---

<b>Descripción</b>	[Inserte aquí el texto]
<b>Justificación</b>	[Inserte aquí el texto]

<b>Descripción</b>	[Inserte aquí el texto]
<b>Justificación</b>	[Inserte aquí el texto]

### 3.1.3. Interfaces de software

[Inserte aquí el texto]

*El AN indicará si es necesario integrar el producto requerido por el cliente con otros productos de software. Para cada producto de software debe especificarse lo siguiente:*

- *Descripción del producto software utilizado.*
- *Propósito de la interfaz.*
- *Definición de la interfaz: contiendo y formato.*

### 3.1.4. Interfaces de comunicación

[Inserte aquí el texto]

*Describir los requerimientos de interfaces de comunicación si existen comunicaciones con otros sistemas y cuáles son los protocolos de comunicación. Estos requerimientos son registrados con descripciones textuales y deberá usarse una viñeta por cada uno. Por ejemplo:*

- *El sistema debe obtener la lectura de las ventas del dispositivo móvil que se entrega a cada empleado a través de una comunicación Bluetooth.*
- *Las lecturas de pedidos serán descargadas a través del puerto RS232 entre dispositivos.*

## 3.2. Requerimientos no funcionales

### 3.2.1. Requerimientos de rendimiento

[Inserte aquí el texto]

*Especificación de los requerimientos relacionados con la carga que se espera tenga que soportar el sistema. Por ejemplo, el número de terminales, el número esperado de usuarios simultáneamente conectados, número de transacciones por segundo que deberá soportar el sistema, etc. Todos estos requerimientos deben ser medibles.*

*Estos requerimientos son registrados con descripciones textuales y deberá usarse una viñeta por cada uno. Por ejemplo:*

- *El 95% de las transacciones deben realizarse en menos de 1 segundo.*
- *Los registros de todas las ventas realizadas deben ser actualizados cada 3 horas.*

### 3.2.2. Seguridad

[Inserte aquí el texto]

*Especificación de elementos que protegerán al software de accesos, usos y sabotajes maliciosos, así como de modificaciones o destrucciones maliciosas o accidentales. Los requerimientos pueden especificar:*

- *Empleo de técnicas criptográficas.*

- Registro de ficheros con “logs” de actividad.
- Asignación de determinadas funcionalidades a determinados módulos.
- Forma de acceso al sistema.
- Restricciones de comunicación entre determinados módulos.
- Comprobaciones de integridad de información crítica.

### 3.2.3. Fiabilidad

[Inserte aquí el texto]

*Especificación de los factores de fiabilidad necesaria del sistema que son subjetivos y no medibles. Estos se expresan generalmente como el tiempo entre los incidentes permisibles, o el total permisible de incidentes. Se recomienda utilizar la siguiente clasificación de incidentes en el funcionamiento de un sistema:*

- *Transitorio: Ocurre solamente con ciertas entradas.*
- *Permanente: Ocurre con todas las entradas.*
- *Recuperable: El sistema puede recuperarse sin la intervención del operador.*
- *Irrecuperable: Es necesaria la intervención del operador para recuperarse del fallo.*
- *No corruptivo: El fallo no corrompe el estado del sistema o los datos.*
- *Corruptivo: El fallo corrompe el estado del sistema o los datos.*

<b>Clase de incidente</b>	Permanente, no corruptivo.
<b>Descripción</b>	Incidente al leer tarjeta.
<b>Ejemplo</b>	El sistema deja de funcionar con cualquier tarjeta que se le introduzca. El software debe reinicializarse para corregir el fallo.
<b>Métrica de fiabilidad</b>	ROCOF (Rate Of Change Of Frequency) 1 ocurrencia/1,000 días

<b>Clase de incidente</b>	[Inserte aquí el texto]
<b>Descripción</b>	[Inserte aquí el texto]
<b>Ejemplo</b>	[Inserte aquí el texto]
<b>Métrica de fiabilidad</b>	[Inserte aquí el texto]

<b>Clase de incidente</b>	[Inserte aquí el texto]
<b>Descripción</b>	[Inserte aquí el texto]
<b>Ejemplo</b>	[Inserte aquí el texto]
<b>Métrica de fiabilidad</b>	[Inserte aquí el texto]

### 3.2.4. Disponibilidad

[Inserte aquí el texto]

*Especificación de los factores de disponibilidad final exigidos al sistema. Normalmente expresados en % de tiempo en los que el software tiene que mostrar disponibilidad. Por ejemplo:*

<b>Descripción</b>	Disponibilidad total
<b>Justificación</b>	La disponibilidad del sistema debe ser del 100% con un nivel de servicio para los usuarios de 7 días X 24 horas, garantizando un esquema adecuado que permita ante una posible falla en cualquiera de sus componentes, contar con una contingencia y generación de alarmas.

### 3.2.5. Mantenibilidad

[Inserte aquí el texto]

*Identificación del tipo de mantenimiento necesario del sistema. Especificación de quién debe realizar las tareas de mantenimiento, por ejemplo usuarios, o un desarrollador. Especificación de cuándo deben realizarse las tareas de mantenimiento. Por ejemplo, generación de estadísticas de acceso semanal y mensual.*

<b>Descripción</b>	Documentación de software.
<b>Justificación</b>	Todo el sistema deberá estar complemente documentado, cada uno de los componentes de software que forman parte del producto final deberán estar debidamente documentados tanto en el código fuente como en los manuales de uso y de mantenimiento.

<b>Descripción</b>	Errores futuros.
<b>Justificación</b>	El sistema debe estar en capacidad de permitir en el futuro su fácil mantenimiento con respecto a los posibles errores que se puedan presentar durante la operación del sistema

### 3.2.6. Portabilidad

[Inserte aquí el texto]

*Especificación de atributos que debe presentar el software para facilitar su traslado a otras plataformas u entornos. Pueden incluirse:*

- *Porcentaje de componentes dependientes del servidor.*
- *Porcentaje de código dependiente del servidor.*
- *Uso de un determinado lenguaje por su portabilidad.*
- *Uso de un determinado compilador o plataforma de desarrollo.*
- *Uso de un determinado sistema operativo.*

### 3.3. Otros requerimientos

[Inserte aquí el texto]

*Cualquier otro requerimiento que no encaje en ninguna de las secciones anteriores, por ejemplo:*

- *Requerimientos culturales y políticos*
- *Requerimientos legales*
- *Requerimientos ambientales.*

<b>Descripción</b>	El producto deberá respetar la norma vigente para el cálculo de impuesto y exenciones.
<b>Justificación</b>	El cliente establece que los cálculos de impuestos deber respetar la ley fiscal actual del país.

<b>Descripción</b>	El producto deberá mostrar los textos en Zapoteco y Quechua, además del Español.
<b>Justificación</b>	Es importante mantener el equilibrio cultural entre los tipos de usuario planeados para el producto.

## 4. Apéndices

[Inserte aquí el texto]

*Este apartado es opcional y puede contener todo tipo de información relevante para la Especificación de Requerimientos pero que, propiamente, no forme parte de la misma. (ej. Documentos entregados por el cliente, Manuales utilizados, Políticas, etc.).*



## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha del reporte</b>	Fecha en la que tanto el RE y el RAN empiezan a realizar la verificación del documento. Es importante considerar que cada verificación realizada genera un Reporte de Verificación si es necesario.
<b>Lugar</b>	Lugar en donde el RE y el RAN realizan la verificación del documento.
<b>Hora de inicio</b>	Hora exacta en la que comienza la verificación del documento.
<b>Hora de fin</b>	Hora exacta en la que termina la verificación del documento.
<b>Duración total</b>	Cálculo numérico (en horas) para expresar el tiempo total que RE y RAN tardaron en verificar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RAN como RDM hayan tenido interrupciones durante la verificación y esta relación no sea del todo cierta.
<b>Nombre completo</b>	Nombres del RE y RAN que participan en la verificación.
<b>Puesto (rol)</b>	Puestos o roles de los participantes en la verificación.
<b>Elemento a verificar</b>	Se deberán listar todos los elementos que se deban verificar usando los Criterios de Verificación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Descripción del producto, Descripción de requerimientos, etc. Se recomienda incluir todos los elementos que forman el documento de la Especificación de Requerimientos y aplicar los criterios definidos, por ejemplo: Redacción, Formato de plantilla, Requerimientos Funcionales, Requerimientos de Interfaz, etc.
<b>Descripción del defecto encontrado</b>	El RE y RAN deben describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento describe claramente todos los requerimientos del sistema”. Un ejemplo de cómo describir un defecto puede ser: “Los requerimientos de mantenibilidad no consideran quién es el responsable del mantenimiento del sistema.”
<b>Fecha de aprobación</b>	Fecha en la que RE y RAN se aseguran que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.



## Instrucciones de llenado de la plantilla

Campo	Descripción
<b>Fecha del reporte</b>	Fecha en la que tanto el RDM y el CL y US empiezan a realizar la validación del documento. Es importante considerar que cada validación realizada puede generar un Reporte de Validación si es necesario.
<b>Lugar</b>	Lugar en donde el RDM y el CL y US realizan la validación del documento.
<b>Hora de inicio</b>	Hora exacta en la que comienza la validación del documento.
<b>Hora de fin</b>	Hora exacta en la que termina la validación del documento.
<b>Duración total</b>	Cálculo numérico (en horas) para expresar el tiempo total que RDM y CL y US tardaron en validar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RDM como CL y US hayan tenido interrupciones durante la validación y esta relación no sea del todo cierta.
<b>Nombre completo</b>	Nombres del RDM, CL y US que participan en la validación.
<b>Puesto (rol)</b>	Puestos o roles de los participantes en la validación.
<b>Elemento a validar</b>	Se deberán listar todos los elementos que se deban validar usando los Criterios de Validación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Descripción del producto, Requerimientos entendibles, etc. Se recomienda incluir todos los elementos que forman el documento de la Especificación de Requerimientos y aplicar los criterios definidos para su validación.
<b>Descripción del defecto encontrado</b>	El RDM debe describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento describe claramente todos los requerimientos del sistema”. Un ejemplo de cómo describir un defecto puede ser: “El requerimiento que refleja la necesidad de imprimir un comprobante de ventas en una impresora térmica no ha sido agregado.”
<b>Fecha de aprobación</b>	Fecha en la que el RDM se asegura que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.





## Definición de los Casos de Prueba

<b>Nombre del caso de prueba</b>	[Inserte aquí el texto]
<b>Propósito</b>	[Inserte aquí el texto]
<b>Pre-requerimientos</b>	[Inserte aquí el texto]
<b>Datos de Prueba</b>	[Inserte aquí el texto]
<b>Pasos</b>	[Inserte aquí el texto]
<b>Notas y Preguntas</b>	[Inserte aquí el texto]

<b>Nombre del caso de prueba</b>	[Inserte aquí el texto]
<b>Propósito</b>	[Inserte aquí el texto]
<b>Pre-requerimientos</b>	[Inserte aquí el texto]
<b>Datos de Prueba</b>	[Inserte aquí el texto]
<b>Pasos</b>	[Inserte aquí el texto]
<b>Notas y Preguntas</b>	[Inserte aquí el texto]

<b>Nombre del caso de prueba</b>	[Inserte aquí el texto]
<b>Propósito</b>	[Inserte aquí el texto]
<b>Pre-requerimientos</b>	[Inserte aquí el texto]
<b>Datos de Prueba</b>	[Inserte aquí el texto]
<b>Pasos</b>	[Inserte aquí el texto]
<b>Notas y Preguntas</b>	[Inserte aquí el texto]

<b>Nombre del caso de prueba</b>	[Inserte aquí el texto]
<b>Propósito</b>	[Inserte aquí el texto]
<b>Pre-requerimientos</b>	[Inserte aquí el texto]
<b>Datos de Prueba</b>	[Inserte aquí el texto]
<b>Pasos</b>	[Inserte aquí el texto]
<b>Notas y Preguntas</b>	[Inserte aquí el texto]

<b>Nombre del caso de prueba</b>	[Inserte aquí el texto]
<b>Propósito</b>	[Inserte aquí el texto]
<b>Pre-requerimientos</b>	[Inserte aquí el texto]
<b>Datos de Prueba</b>	[Inserte aquí el texto]
<b>Pasos</b>	[Inserte aquí el texto]
<b>Notas y Preguntas</b>	[Inserte aquí el texto]

<b>Nombre del caso de prueba</b>	[Inserte aquí el texto]
<b>Propósito</b>	[Inserte aquí el texto]
<b>Pre-requerimientos</b>	[Inserte aquí el texto]
<b>Datos de Prueba</b>	[Inserte aquí el texto]
<b>Pasos</b>	[Inserte aquí el texto]
<b>Notas y Preguntas</b>	[Inserte aquí el texto]

<b>Nombre del caso de prueba</b>	[Inserte aquí el texto]
<b>Propósito</b>	[Inserte aquí el texto]
<b>Pre-requerimientos</b>	[Inserte aquí el texto]
<b>Datos de Prueba</b>	[Inserte aquí el texto]
<b>Pasos</b>	[Inserte aquí el texto]
<b>Notas y Preguntas</b>	[Inserte aquí el texto]

## Instrucciones de llenado de la plantilla

Campo	Descripción
<b>Definición de pruebas</b>	
<b>Fecha</b>	Fecha en la que se realiza el plan de pruebas.
<b>Hora de inicio</b>	Hora exacta en la que comienza la redacción del plan de pruebas.
<b>Hora de fin</b>	Hora exacta en la que termina la redacción del plan de pruebas.
<b>Identificación del requerimiento</b>	Mediante su referencia deberán listarse todos los requerimientos que fueron aceptados en la Especificación de Requerimientos.
<b>Tipo de requerimiento</b>	De acuerdo a su especificación, se deberá indicar si es un requerimiento funcional, de interfaz, de rendimiento, de mantenimiento, etc.
<b>Tipo de prueba</b>	<p>Los tipos de pruebas que se podrán manejar son las siguientes:</p> <ul style="list-style-type: none"> <li>• Pruebas de integridad sobre la base de datos e información. Las bases de datos y los procesos de acceso a la misma deben ser evaluados como un subsistema. Esta prueba debe probar los subsistemas independientemente de la interfaz de usuario.</li> <li>• Prueba funcional. Esta prueba se enfoca en verificar que el sistema cumpla con los requerimientos definidos, esta es una prueba de caja negra que se basa en los resultados obtenidos a través de la interfaz gráfica.</li> <li>• Prueba de interfaz de usuario. Verifica la interacción con el software, asegurando que el usuario cuenta con el acceso y navegación adecuada para desarrollar las funcionalidades del sistema. Además revisa que los objetos de la interfaz gráfica se comporten de manera adecuada y cumplan con los estándares establecidos por la empresa o la industria.</li> <li>• Pruebas de rendimiento. Se verifica el desempeño del sistema para cumplir con los requerimientos establecidos. Por ejemplo: Tiempo de Respuesta, Número de Transacciones procesadas por unidad de tiempo, etc.</li> <li>• Pruebas de carga. Se verifica la funcionalidad del sistema en diferentes situaciones de carga de trabajo, ya sea esperada o más allá del límite. Se verifica tiempo de respuesta, número de transacciones procesadas, etc., para picos de carga, carga alta sostenida, simulación de carga en un periodo de tiempo (ej. Cierre mensual, Comportamiento de las cargas en diferentes configuraciones de equipo, etc.).</li> <li>• Pruebas de estrés. Se verifica la funcionalidad del sistema bajo condiciones de recursos que no se presentan de manera normal, por ejemplo, poca memoria disponible, usuarios realizando la misma transacción a la vez, ancho de banda disponible. Algunas de estas pruebas pueden haberse considerado en pruebas de funcionalidad (si es un requerimiento de la aplicación) o pruebas de carga (al haber probado situaciones de carga fuera de los límites esperados, ej. Número de usuarios conectados más allá del límite esperado).</li> <li>• Pruebas de control de acceso y seguridad. Pruebas a nivel de acceso al sistema, a fin de verificar que éste es adecuado para los datos o funciones del negocio, según se requiera. Se requiere de pruebas de seguridad a nivel sistema que aseguren que solamente los usuarios autorizados tienen acceso al sistema y son capaces de acceder al sistema a través de los canales apropiados.</li> <li>• Pruebas de falla y recuperación. Pruebas que permiten asegurar que el sistema es capaz de recuperarse ante una falla de software o hardware con la consecuente pérdida de integridad en los datos. Es decir, se prueba que existan los mecanismos adecuados de respaldo y recuperación de información, o de autocorrección de transacciones que no fueron completadas, identificación de datos inconsistentes en la base de datos y su respectiva corrección.</li> <li>• Pruebas de configuración. Verifican que el sistema se comporta adecuadamente en diferentes plataformas de hardware o software para las cuales fue realizada. Incluyendo uso de diferentes versiones del sistema operativo, marcas de browsers, etc.</li> <li>• Pruebas de instalación. Verifican que la aplicación y sus actualizaciones sean instaladas adecuadamente.</li> </ul>
<b>Descripción de la prueba</b>	Representación y vinculación con el caso de prueba que corresponda en la Definición de los

	Casos de Prueba.
<b>Resultado esperado</b>	Salida esperada del sistema y que corresponde con la funcionalidad reflejada en el requerimiento relacionado. Es necesario indicar la tolerancia a fallas o criterios para decidir si el sistema es satisfactorio en este tipo de pruebas.
<b>Calendarización</b>	Fecha propuesta para realizar la prueba, ésta debe ser consistente con las fechas establecidas en el Calendario del Plan de Desarrollo.
<b>Recursos para establecer el entorno de pruebas</b>	
<b>Recurso</b>	Cualquier tipo de recurso (infraestructura o equipo electrónico, de cómputo o tecnológico, documentos, manuales, etc.) que debe estar disponible para realizar las pruebas sin contratiempo. Por ejemplo: Servidor de Base de Datos, Nombre del Servidor, Nombre de la Base de Datos, Computadoras Personales, Impresoras, etc.
<b>Nombre/Tipo/Descripción</b>	Descripción textual del recurso y su justificación en relación con las pruebas.
<b>Herramientas necesarias para ejecutar el plan de pruebas</b>	
<b>Herramienta</b>	Definición de las herramientas que serán utilizadas para ejecutar las pruebas descritas en la Definición de Pruebas (ej. Diseño de Casos de Prueba, Registros de Defectos, Reporte de Defectos, Gráficas/Métricas de Defectos, Automatización de Pruebas, etc.). Para este caso se ha definido por default el uso de Casos de Pruebas, pero queda abierto a las herramientas usadas actualmente en la empresa.
<b>Descripción</b>	Descripción textual de la herramienta y su justificación en relación con las pruebas.
<b>Recursos que el cliente debe proporcionar para realizar las pruebas</b>	
<b>Recurso</b>	Lista de recursos necesarios para realizar las pruebas y que deben ser provistos por el cliente. Por ejemplo: el sistema asume que el cliente tiene un servidor propio para instalar y probar el sistema.
<b>Información relevante</b>	Justificación de la importancia del recurso en relación con la realización de las pruebas.
<b>Fecha compromiso</b>	Fecha en la que deben estar disponibles los recursos y cualquier información relevante con respecto a los mismos.

<b>Definición de los casos de prueba</b>	
<b>Nombre del caso de prueba</b>	Nombre único que le es asignado a cada caso de prueba. Es común que el nombre tenga relación con la funcionalidad que se desea probar.
<b>Propósito</b>	Una o dos oraciones cortas sobre el aspecto del sistema que está siendo probado. Si esto toma mucho tiempo, se recomienda interrumpir el caso de prueba o proporcionar más información en la descripción del caso de prueba.
<b>Pre-requerimientos</b>	Suposiciones que deben cumplirse antes de que correr el caso de prueba. Por ejemplo: "el usuario <i>admin</i> ya está registrado", "inicio de sesión como invitado permitido", etc.
<b>Datos de Prueba</b>	Lista de variables y sus posibles valores usados en el caso de prueba. Es posible enlistar valores específicos o describir rangos de valores. El caso de prueba deberá ser ejecutado una vez por cada combinación de valores. Estos valores se escriben con notación de asignación, uno por línea. Por ejemplo: <ul style="list-style-type: none"> <li>• loginID = {loginID válido, loginID inválido, email válido, email inválido, vacío}</li> <li>• password = {válido, inválido, vacío}</li> </ul>
<b>Pasos</b>	Pasos ordenados para ejecutar la prueba. Estos pasos deben respetar el formato establecido por ejemplo en: <ol style="list-style-type: none"> <li>1. visitar LoginPage</li> <li>2. teclear usernameOrEmail</li> <li>3. teclear password</li> <li>4. hacer click en Entrar</li> <li>5. ver: la página de los términos de uso</li> <li>6. hacer click hasta el fondo de la página</li> <li>7. hacer en click Aceptar</li> <li>8. ver: PersonalPage</li> <li>9. verificar el mensaje de bienvenida si el inicio de sesión es correcto</li> </ol>
<b>Notas y Preguntas</b>	Cualquier información adicional que extienda la descripción del caso de prueba o bien alguna pregunta sobre la funcionalidad y que servirá para determinar algún error en el sistema.





## Instrucciones de llenado de la plantilla

Campo	Descripción
<b>Fecha del reporte</b>	Fecha en la que tanto el RE y el RAN empiezan a realizar la verificación del documento. Es importante considerar que cada verificación realizada genera un Reporte de Verificación si es necesario.
<b>Lugar</b>	Lugar en donde el RE y el RAN realizan la verificación del documento.
<b>Hora de inicio</b>	Hora exacta en la que comienza la verificación del documento.
<b>Hora de fin</b>	Hora exacta en la que termina la verificación del documento.
<b>Duración total</b>	Cálculo numérico (en horas) para expresar el tiempo total que RE y RAN tardaron en verificar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RAN como RDM hayan tenido interrupciones durante la verificación y esta relación no sea del todo cierta.
<b>Nombre completo</b>	Nombres del RE y RAN que participan en la verificación.
<b>Puesto (rol)</b>	Puestos o roles de los participantes en la verificación.
<b>Elemento a verificar</b>	Se deberán listar todos los elementos que se deban verificar usando los Criterios de Verificación de la documentación de procesos. . Por ejemplo: Consistencia o adherencia con la Especificación de Requerimientos, Descripción de los Casos de Prueba, etc. Se recomienda incluir todos los elementos que forman el documento del Plan de Pruebas del Sistema y aplicar los criterios definidos, por ejemplo: Redacción, Formato de plantilla, Descripciones de recursos, Identificación de herramientas requeridas para la ejecución de las pruebas, etc.
<b>Descripción del defecto encontrado</b>	El RE y RAN deben describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento describe claramente todas las pruebas que serán realizadas”. Un ejemplo de cómo describir un defecto puede ser: “Los recursos no contemplan quién tendrá la responsabilidad de proporcionar el servidor para instalar el sistema y proceder con las pruebas.”
<b>Fecha de aprobación</b>	Fecha en la que RE y RAN se aseguran que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.



## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha del reporte</b>	Fecha en la que tanto el RDM y el CL y US empiezan a realizar la validación del documento. Es importante considerar que cada validación realizada puede generar un Reporte de Validación si es necesario.
<b>Lugar</b>	Lugar en donde el RDM y el CL y US realizan la validación del documento.
<b>Hora de inicio</b>	Hora exacta en la que comienza la validación del documento.
<b>Hora de fin</b>	Hora exacta en la que termina la validación del documento.
<b>Duración total</b>	Cálculo numérico (en horas) para expresar el tiempo total que RDM y CL y US tardaron en validar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RDM como CL y US hayan tenido interrupciones durante la validación y esta relación no sea del todo cierta.
<b>Nombre completo</b>	Nombres del RDM, CL y US que participan en la validación.
<b>Puesto (rol)</b>	Puestos o roles de los participantes en la validación.
<b>Elemento a validar</b>	Se deberán listar todos los elementos que se deban validar usando los Criterios de Validación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Especificación de Requerimientos, Recursos requeridos, etc. Se recomienda incluir todos los elementos que forman el documento del Plan de Pruebas de Sistema y aplicar los criterios definidos para su validación.
<b>Descripción del defecto encontrado</b>	El RDM debe describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento cubre claramente los requerimientos establecidos por el cliente en relación al sistema”. Un ejemplo de cómo describir un defecto puede ser: “No se ha incluido un caso de prueba que incluya la generación de cortes de caja.”
<b>Fecha de aprobación</b>	Fecha en la que el RDM se asegura que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.

## Plantilla para Manual de Usuario: DMS Actividad A.2.20

### 1. Datos de control

Estos datos deben ser llenados por el RM al momento de empezar y terminar con la generación del Manual de Usuario. Así, esta información sirve únicamente para que el RDM controle el proyecto, una vez que este manual esté listo para entregarse al CL deberán ser borrados del mismo.

Preparado por	[Inserte aquí el texto]	Fecha	[Fecha]
Hora de inicio	[Rev]	Hora de fin	[Rev]

### 2. Derechos de autor de OBX

El CL es libre de:

Copiar, distribuir y comunicar públicamente este Manual de Usuario.

Realizar obras derivadas del mismo bajo las condiciones siguientes:

- Reconocimiento: Debe reconocer y citar a la Empresa [Inserte aquí el texto] como autor original.
- Fines no comerciales: No puede utilizar este Manual de Usuario para fines comerciales.
- Compartir bajo la misma licencia: Si altera o transforma este Manual de Usuario, o genera una obra derivada de éste, solamente puede distribuir la obra generada bajo una licencia de la Empresa [Inserte aquí el texto].

### 3. Objetivo del manual

El objetivo primordial de este Manual de Usuario es ayudar y guiar al usuario en la instalación y uso del Sistema [Inserte aquí el texto] a través de información resumida que le permitirá despejar todas las dudas existentes y facilitar su entendimiento sobre el mismo.

### 4. Descripción general del sistema

[Inserte aquí el texto]

*Este apartado debe proporcionar a la audiencia una introducción y visión general del sistema para el cual se está generando el Manual de Usuario.*

### 5. Audiencia

*Es necesario definir al personal específico que se dirige este Manual de Usuario. Sería recomendable que esta misma audiencia tenga los roles del personal involucrado en la obtención de los requerimientos.*

<b>Descripción de rol</b>	[Inserte aquí el texto]
<b>Categoría profesional</b>	[Inserte aquí el texto]
<b>Responsabilidades</b>	[Inserte aquí el texto]

<b>Descripción de rol</b>	[Inserte aquí el texto]
<b>Categoría profesional</b>	[Inserte aquí el texto]
<b>Responsabilidades</b>	[Inserte aquí el texto]

<b>Descripción de rol</b>	[Inserte aquí el texto]
<b>Categoría profesional</b>	[Inserte aquí el texto]
<b>Responsabilidades</b>	[Inserte aquí el texto]

## 6. Requerimientos mínimos para usar el manual

[Inserte aquí el texto]

*Se deberán listar los requerimientos relacionados con el conocimiento previo que debe tener la audiencia de este Manual de Usuario. Por ejemplo:*

- *Conocimientos básicos sobre programas utilitarios.*
- *Conocimientos básicos de Navegación Web.*
- *Conocimientos básicos de Internet.*
- *Conocimientos básicos de Windows.*

## 7. Convenciones y estándares

*Es necesario especificarle a la audiencia las convenciones de formato que encontrará en el texto del Manual de Usuario, por ejemplo:*

Formato	Tipo de información
Viñeta numérica	Numeración de procedimientos paso a paso
Negrita	Términos de importancia que requieren especial atención.
Menor que (<) Mayor que (>)	Nombre de opciones que el usuario puede elegir ya sea haciendo uso del teclado o el mouse.
! (signo de admiración)	Consejo en caso de error.

*Se deberá especificar también la convención para el uso del mouse y del teclado, por ejemplo:*

Término	Significado
Señalar	Colocar el extremo superior del mouse sobre el elemento que se desea señalar.
Hacer clic.	Presionar el botón principal del mouse (generalmente el botón izquierdo) y soltarlo lentamente.
Desplegar el menú contextual	Presionar el botón secundario del mouse (generalmente el botón derecho) y soltarlo lentamente.

Término	Significado
Teclas de método abreviado 	Teclas que se utilizan como método abreviado para ejecutar un proceso. Por ejemplo: F1 para mostrar la ayuda.
Tecla Enter 	Tecla utilizada para ejecutar un proceso cuando el manual indique "Presionar Enter".

## 8. Actividades de instalación

[Inserte aquí el texto]

*Utilizando las convenciones y estándares definidos en el apartado anterior, se enlistan los pasos para realizar la correcta instalación del sistema. El RM deberá agregar imágenes que le sirvan de soporte para describir los pasos y por consecuencia mejorar el entendimiento del CL y US.*

## 9. Descripción de usos y funcionalidades

[Inserte aquí el texto]

*Utilizando las convenciones y estándares definidos en el apartado anterior, se enlistan los pasos para realizar las funcionalidades generales del sistema. El RM deberá agregar imágenes que le*

*sirvan de soporte para describir los pasos y por consecuencia mejorar el entendimiento del CL y US. En este caso, el RM deberá apoyarse en el RAN para establecer una secuencia de presentación que resuma las funcionalidades del sistema.*

## **10. Preguntas frecuentes**

[Inserte aquí el texto]

*Es recomendable agregar una sección de preguntas y respuestas que ayuden al CL o US a encontrar soluciones rápidas a problemas sencillos.*

## **11. Glosario de términos y acrónimos**

[Inserte aquí el texto]

*Para un mejor entendimiento se recogen todos aquellos términos del Manual de Usuario que pueden confundir a la audiencia y afectar el entendimiento del sistema.*

<b>Término</b>	<b>Acrónimo</b>	<b>Descripción</b>
[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]





### Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha del reporte</b>	Fecha en la que tanto el RM y el RAN empiezan a realizar la verificación del documento. Es importante considerar que cada verificación realizada genera un Reporte de Verificación si es necesario.
<b>Lugar</b>	Lugar en donde el RM y el RAN realizan la verificación del documento.
<b>Hora de inicio</b>	Hora exacta en la que comienza la verificación del documento.
<b>Hora de fin</b>	Hora exacta en la que termina la verificación del documento.
<b>Duración total</b>	Cálculo numérico (en horas) para expresar el tiempo total que RM y RAN tardaron en verificar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RAN como RM hayan tenido interrupciones durante la verificación y esta relación no sea del todo cierta.
<b>Nombre completo</b>	Nombres del RM y RAN que participan en la verificación.
<b>Puesto (rol)</b>	Puestos o roles de los participantes en la verificación.
<b>Elemento a verificar</b>	Se deberán listar todos los elementos que se deban verificar usando los Criterios de Verificación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Especificación de Requerimientos. Se recomienda incluir todos los elementos que forman el documento de Manual de Usuario y aplicar los criterios definidos, por ejemplo: Redacción, Formato de plantilla, Descripciones de pasos de instalación del sistema, Uso de convenciones y estándares definidos para describir los usos o funcionalidades, etc.
<b>Descripción del defecto encontrado</b>	El RM y RAN deben describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento describe claramente todos los pasos que el US debe realizar”. Un ejemplo de cómo describir un defecto puede ser: “El manual no describe correctamente cómo realizar el alta de un empleado.”
<b>Fecha de aprobación</b>	Fecha en la que RM y RAN se aseguran que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.



--	--	--

### Instrucciones de llenado de la plantilla

Campo	Descripción
<b>Fecha del reporte</b>	Fecha en la que tanto el RDM y el RAPE actualizan la Configuración de Software. Es importante considerar que cada actualización realizada genera un cambio de fecha si es necesario.
<b>Proyecto</b>	Referencia al proyecto para el cual se han generado todos los productos específicos, ya sea por ciclo o en su totalidad.
<b>Hora de inicio</b>	Hora exacta en la que comienza la realización del documento.
<b>Hora de fin</b>	Hora exacta en la que termina la realización del documento.
<b>Duración total</b>	Cálculo numérico (en horas) para expresar el tiempo total que RDM y RAPE tardaron en realizar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RDM como RAPE hayan tenido interrupciones realizando la actividad y esta relación no sea del todo cierta.
<b>Nombre completo</b>	Nombres del RDM y RAPE que participan en la documentación de los productos.
<b>Producto</b>	Todo aquel producto generado durante el ciclo/proyecto y que debe ser identificado y consultado fácilmente. La lista de productos ya está definida por default.
<b>Ubicación del producto</b>	Ubicación de los productos dentro de la estructura de carpetas de la Base de Conocimiento (servidor) que la empresa haya generado para actualizar toda la información relacionada con el proyecto.
<b>Última versión activa</b>	Última versión generada mediante el control de versiones de esta plantilla y que evitará utilizar documentos duplicados o con información incorrecta.



## Instrucciones de llenado de la plantilla

Campo	Descripción
<b>Datos generales</b>	
<b>Nombre del proyecto</b>	Nombre que le fue asignado al proyecto en el Plan de Desarrollo.
<b>Referencia del proyecto</b>	Referencia única que se le da a cada proyecto para referirse a él de forma abreviada. Estas referencias suelen tener identificadores que ayudan al personal a determinar la naturaleza del proyecto sin consultar el nombre o descripción completa del mismo. El estilo de las referencias debe ser definido por la empresa y debe ser respetado por todo el Equipo de Trabajo.
<b>Nombre del RAN</b>	Nombre del Responsable de Analistas que fue asignado al proyecto en cuestión.
<b>Nombre del RPR</b>	Nombre del Responsable de Programadores que fue asignado al proyecto en cuestión.
<b>Registro de rastreo</b>	
<b>#</b>	Identificador numérico único para cada requerimiento a rastrear.
<b>Requerimiento</b>	Nombre dado al requerimiento en la Especificación de Requerimientos (ej. RF 1.13).
<b>Origen del requerimiento</b>	Proveedor del requerimiento que está siendo rastreado (ej. Gerente de Ventas, Dueño, Secretaría).
<b>Referencias</b>	Relaciones que tiene el requerimiento con otros requerimientos (ej. RF 1.1, RF 1.2).
<b>Caso de uso</b>	Nombre del caso de uso que incluye al requerimiento y que ha sido obtenido en el Análisis y Diseño.
<b>Especificación de diseño</b>	Diagrama de caso de uso que incluye al requerimiento en cuestión.
<b>Programado</b>	Funcionalidad que es descrita por el requerimiento y pantalla en la que cual se implementa.
<b>Tabla de la Base de Datos</b>	Nombre de la tabla de la BD donde se ve reflejada la implementación del requerimiento.
<b>Caso de prueba</b>	Nombre del caso de prueba con el que se relaciona el requerimiento. Este nombre se obtiene del Plan de Pruebas de Sistema.
<b>Estado</b>	Situación actual del requerimiento ( <i>Activo</i> para decir que está en implementación, <i>Terminado</i> para indicar que se ha terminado su implementación, <i>Pendiente</i> para decir que se está realizando un análisis más exhaustivo sobre él, y <i>Eliminado</i> para indicar que el requerimiento ha sido borrado de la especificación).
<b>Observaciones</b>	Cualquier información adicional que se desee incorporar al registro de rastreo.

## Plantilla para Análisis y Diseño: DMS Actividad A.3.3

### 1. Introducción

Los requerimientos documentados en la Especificación de Requerimientos son analizados meticulosamente para producir una descripción detallada de la estructura de los componentes de software, la cual servirá de base para la construcción del sistema.

### 2. Análisis y Diseño

Este documento contiene la descripción textual y gráfica de la estructura de los componentes de software divididos en dos partes: Arquitectónica (Diseño de alto nivel) y Detallada (Diseño de bajo nivel).

#### 2.1. Estructura arquitectónica

[Inserte aquí el texto]

*La estructura arquitectónica comienza por describir con mayor detalle los casos de uso que representen las funcionalidades descritas en la Especificación de Requerimientos a través del Diagrama de Contexto de Casos de Uso. Por ejemplo, el diagrama ejemplo de la Especificación de Requerimientos debería descomponerse en funcionalidades más específicas, una de ellas para el caso de uso Administrar proyectos quedaría como:*

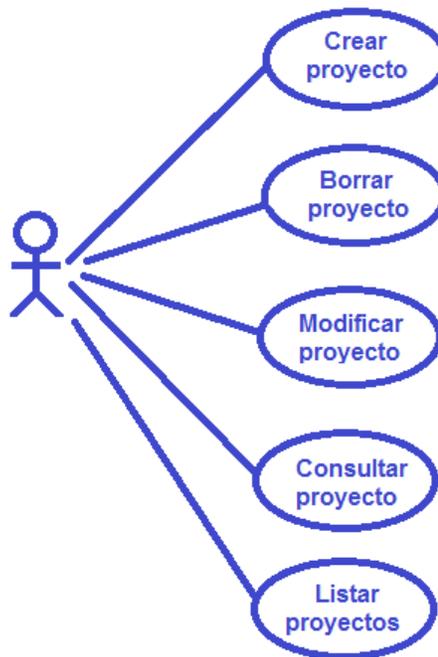


Fig 1. Ejemplo de casos de uso para Administrar proyectos

*Utilizando estos diagramas como punto de partida, los AN deberán detallar cada caso de uso de la siguiente forma.*

<b>Caso de uso</b>	<i>Crear proyecto</i>
<b>Actores</b>	<i>Administrador</i>
<b>Propósito</b>	<i>Dar de alta o crear un nuevo proyecto o grupo de proyectos</i>
<b>Visión general</b>	<i>El Administrador crea un proyecto o grupo de proyectos de</i>

	<i>acuerdo con un formato establecido</i>	
<b>Referencias</b>	RF 1.1, RF 1.2	
<b>Curso típico de eventos</b>		
	<i>Administrador</i>	<i>Sistema</i>
	1. El Usuario accede a la pantalla de alta de proyecto desde el menú principal del sistema.	2. El sistema muestra una lista de proyectos existentes (si los hubiera).
	3. El Administrador rellena los datos de identificación del proyecto seleccionando el número de proyectos a insertar, los ciclos y el patrón que usará.	4. El sistema comprueba que los datos son correctos, los almacena en la BD y vuelve a mostrar la lista con todos los proyectos creados.
<b>Cursos alternativos</b>		

<b>Caso de uso</b>	[Inserte aquí el texto]	
<b>Actores</b>	[Inserte aquí el texto]	
<b>Propósito</b>	[Inserte aquí el texto]	
<b>Visión general</b>	[Inserte aquí el texto]	
<b>Referencias</b>	[Inserte aquí el texto]	
<b>Curso típico de eventos</b>		
	[Inserte aquí el texto]	[Inserte aquí el texto]
	[Inserte aquí el texto]	[Inserte aquí el texto]
<b>Cursos alternativos</b>		
[Inserte aquí el texto]		

<b>Caso de uso</b>	[Inserte aquí el texto]	
<b>Actores</b>	[Inserte aquí el texto]	
<b>Propósito</b>	[Inserte aquí el texto]	
<b>Visión general</b>	[Inserte aquí el texto]	
<b>Referencias</b>	[Inserte aquí el texto]	
<b>Curso típico de eventos</b>		
	[Inserte aquí el texto]	[Inserte aquí el texto]
	[Inserte aquí el texto]	[Inserte aquí el texto]
<b>Cursos alternativos</b>		
[Inserte aquí el texto]		

## 2.2. Arquitectura del software

[Inserte aquí el texto]

*La arquitectura de software representa el enlace entre la Especificación de Requerimientos y el diseño en sí. A efectos de agilizar el ciclo de desarrollo, esta tarea se puede realizar en paralelo con las actividades para la Especificación de Requerimientos; sin embargo cuando la empresa carece de madurez en el uso de estas prácticas lo mejor es ir paso a paso. El diseño de la arquitectura del software implica un esfuerzo creativo por parte de los AN, de forma que las actividades a realizar pueden cambiar de acuerdo a la naturaleza del sistema a desarrollar.*

## 2.3. Descomposición en subsistemas

*La descomposición modular o con subsistemas proporciona pequeños componentes que proveen servicios a otros componentes y que no se consideran un sistema separado. Es decir, en base al entendimiento de los requerimientos funcionales el sistema se descompone en objetos que*

interactúan para realizar una tarea específica. El paso inicial consiste en obtener un diagrama de bloques que resuma la estructura del sistema en términos de los componentes que lo forman. Por ejemplo:

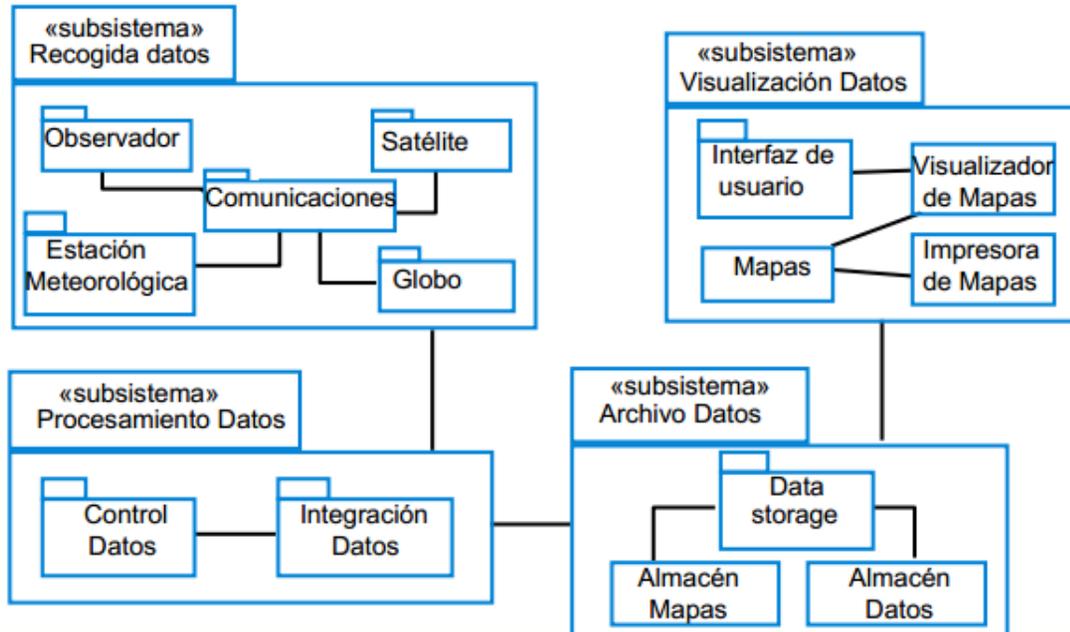


Fig. 1. Ejemplo de diagrama de bloques

<b>Nombre del subsistema</b>	Recogida de datos
<b>Descripción</b>	Este subsistema se encargará de recibir la información a través de una estación meteorológica, un observador, el satélite, el globo lanzado a la atmosfera o bien a través de un protocolo de comunicaciones.

<b>Nombre del subsistema</b>	[Inserte aquí el texto]
<b>Descripción</b>	[Inserte aquí el texto]

<b>Nombre del subsistema</b>	[Inserte aquí el texto]
<b>Descripción</b>	[Inserte aquí el texto]

#### 2.4. Arquitectura por capas

Descomposición del diagrama de bloques en entidades que proporcionan información concisa sobre cada subsistema. Se recomienda no tener más de siete entidades en un modelo arquitectónico. Por ejemplo:

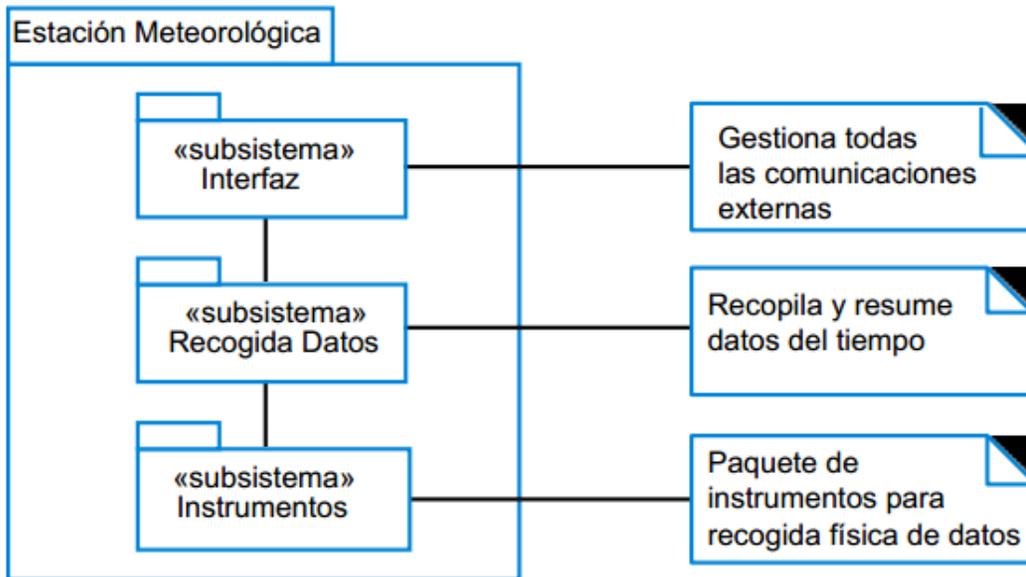


Fig. 2. Ejemplo de descomposición del diagrama de bloques

## 2.5. Representación estructural o estática del software

[Inserte aquí el texto]

Agregar el diagrama de clases y objetos para representar el conjunto de componentes (objetos) y sus interrelaciones. Este modelo de objetos mostrará los componentes o subsistemas principales del software.

### 2.5.1. Identificación de objetos

[Inserte aquí el texto]

La representación estructural inicia con la identificación de objetos para posteriormente relacionarlos en un modelo estático. Por ejemplo:

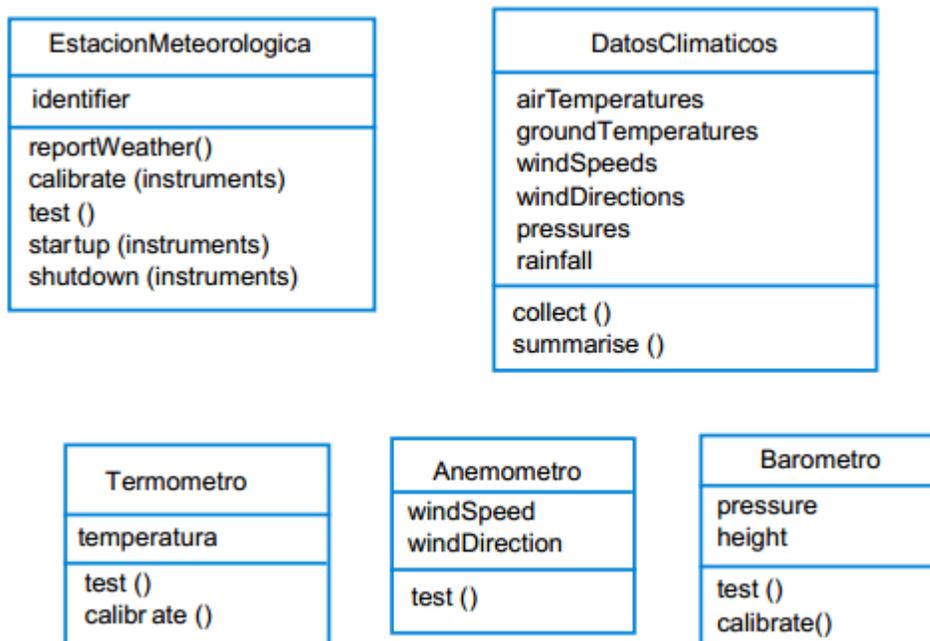


Fig. 3. Ejemplo de objetos identificados

**2.5.2. Modelo estático**

*El modelo estático empezará a sustituir los subsistemas por objetos que mejoren la comprensión para su implementación.*

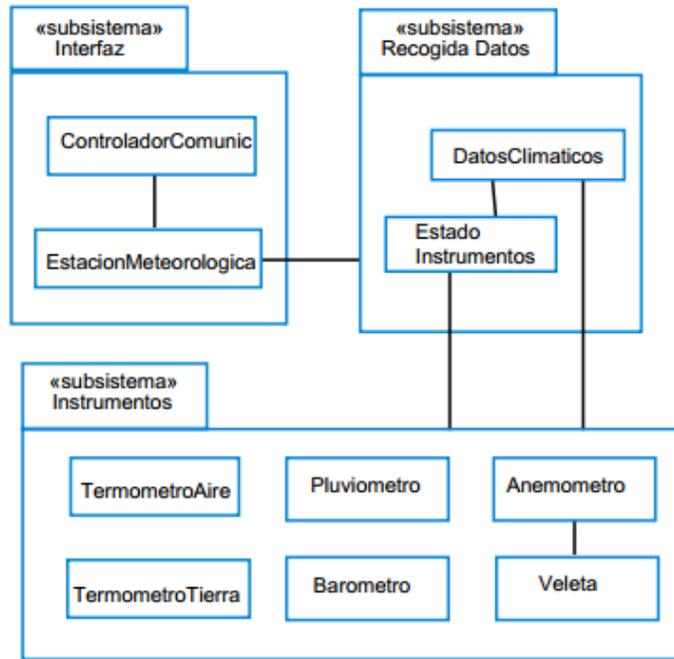


Fig. 4. Ejemplo de modelo estático

**2.6. Representación dinámica del software**

[Inserte aquí el texto]

*En base a la arquitectura que se está estableciendo, los AN comenzarán a diseñar modelos dinámicos de diseño en base a los casos de uso descritos en 2.1. Se recomienda obtener de entre 3 a no más de 36 diagramas de secuencia por caso de uso. Por ejemplo:*

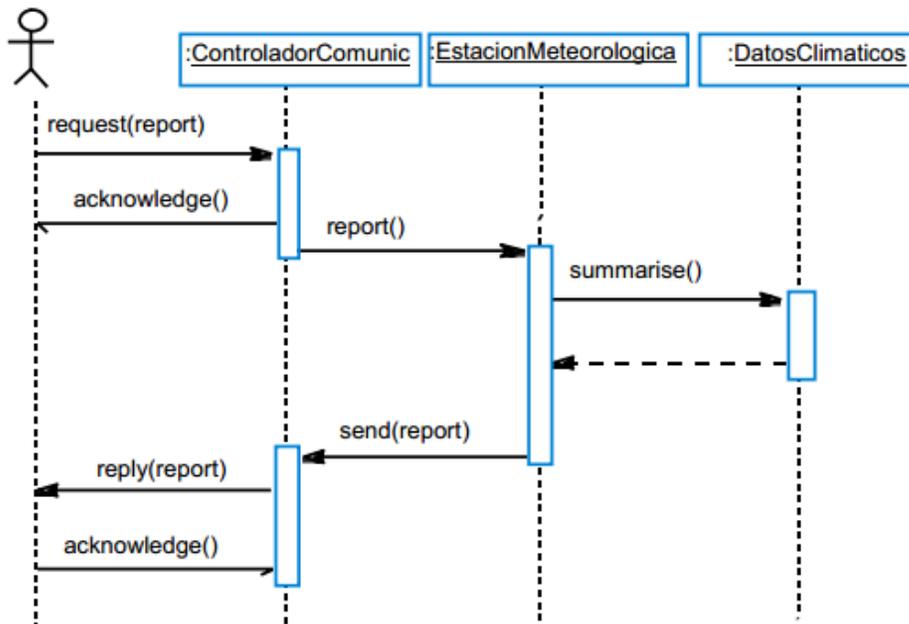


Fig. 5. Ejemplo de modelo dinámico con diagrama de secuencias

## 2.7. Representación por interfaces

La representación por interfaces define con claridad las decisiones de los AN para facilitar la construcción del sistema. Por ejemplo, una interfaz para la EstacionMeteorologica podría ser:

```
interface WeatherStation {
public void WeatherStation ();
public void startup ();
public void startup (Instrument i);
public void shutdown ();
public void shutdown (Instrument i);
public void reportWeather ( );
public void test ();
public void test ( Instrument i );
public void calibrate ( Instrument i);
public int getID ();
} //WeatherStation.
```

## 2.8. Modelo de datos (E-R)

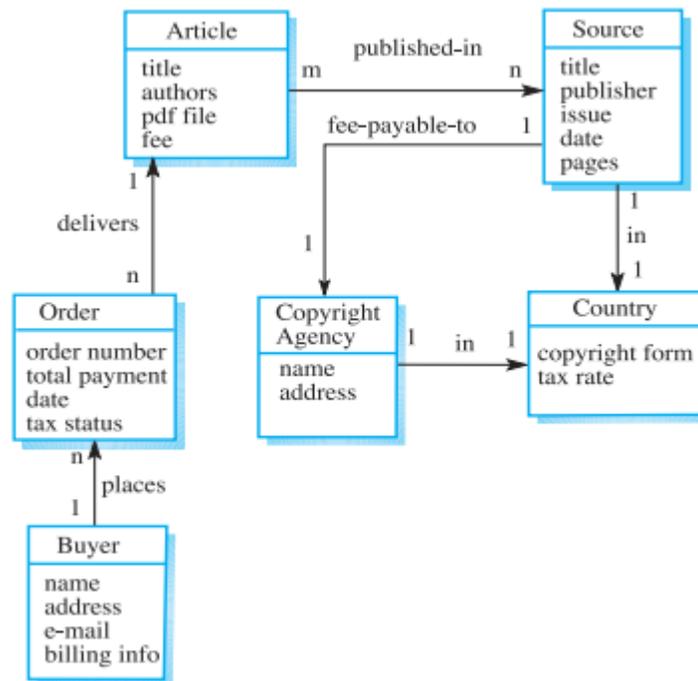


Fig. 6. Ejemplo de modelo entidad-relación

### 2.8.1. Diseño de tablas y atributos

La siguiente descripción presenta en orden alfabético las diversas tablas que componen al modelo completo de datos (E-R). Para cada tabla se debe hacer la descripción de los campos, así como las claves propias y ajenas que la componen. En algunos casos, se muestra una representación de la tabla y se incluye el código SQL que la crea en la BD. Cada tabla se describe de la siguiente manera.

Nombre de la tabla		<i>calendario</i>				
Campo	Tipo	Nulo	PK	FK	Descripción	
<i>IdCiclo</i>	<i>Integer</i>	<i>No</i>	<i>Si</i>	<i>Si</i>	<i>Número de ciclo dentro del proyecto</i>	
<i>IdProyecto</i>	<i>Integer</i>	<i>No</i>	<i>Si</i>	<i>Si</i>	<i>Proyecto al que se asocia el calendario</i>	
<i>IdSemana</i>	<i>Integer</i>	<i>No</i>	<i>Si</i>	<i>No</i>	<i>Identificativo de la sema del proyecto</i>	
<i>Fecha</i>	<i>Date</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>Fecha que representa el lunes de cada semana</i>	
<i>Horas</i>	<i>Integer</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>Horas reales en la semana</i>	

Nombre de la tabla		[Inserte aquí el texto]				
Campo	Tipo	Nulo	PK	FK	Descripción	
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]	
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]	
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]	
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]	
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]	

Nombre de la tabla		[Inserte aquí el texto]				
Campo	Tipo	Nulo	PK	FK	Descripción	
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]	
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]	
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]	
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]	
[Rev]	[Rev]	[Rev]	[Rev]	[Rev]	[Inserte aquí el texto]	

### 3. Apéndices

[Inserte aquí el texto]

*Este apartado es opcional y puede contener todo tipo de información relevante para el Análisis y Diseño pero que, propiamente, no forme parte de la información requerida.*





## Instrucciones de llenado de la plantilla

Campo	Descripción
<b>Fecha del reporte</b>	Fecha en la que tanto el RE y el RAN empiezan a realizar la verificación del documento. Es importante considerar que cada verificación realizada genera un Reporte de Verificación si es necesario.
<b>Lugar</b>	Lugar en donde el RE y el RAN realizan la verificación del documento.
<b>Hora de inicio</b>	Hora exacta en la que comienza la verificación del documento.
<b>Hora de fin</b>	Hora exacta en la que termina la verificación del documento.
<b>Duración total</b>	Cálculo numérico (en horas) para expresar el tiempo total que RE y RAN tardaron en verificar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RAN como RDM hayan tenido interrupciones durante la verificación y esta relación no sea del todo cierta.
<b>Nombre completo</b>	Nombres del RE y RAN que participan en la verificación.
<b>Puesto (rol)</b>	Puestos o roles de los participantes en la verificación.
<b>Elemento a verificar</b>	Se deberán listar todos los elementos que se deban verificar usando los Criterios de Verificación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Especificación de Requerimientos, Descripción detallada de los Casos de Uso, etc. Se recomienda incluir todos los elementos que forman el documento del Plan de Pruebas del Sistema y aplicar los criterios definidos, por ejemplo: Redacción, Formato de plantilla, Modelo estructural, Modelo dinámico, etc.
<b>Descripción del defecto encontrado</b>	El RE y RAN deben describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento describe claramente todas las decisiones de diseño”. Un ejemplo de cómo describir un defecto puede ser: “El diagrama de secuencia no contempla todos los casos de uso descritos.”
<b>Fecha de aprobación</b>	Fecha en la que RE y RAN se aseguran que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.



## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha del reporte</b>	Fecha en la que tanto el RE y el RAN empiezan a realizar la verificación del documento. Es importante considerar que cada verificación realizada genera un Reporte de Verificación si es necesario.
<b>Lugar</b>	Lugar en donde el RE y el RAN realizan la verificación del documento.
<b>Hora de inicio</b>	Hora exacta en la que comienza la verificación del documento.
<b>Hora de fin</b>	Hora exacta en la que termina la verificación del documento.
<b>Duración total</b>	Cálculo numérico (en horas) para expresar el tiempo total que RE y RAN tardaron en verificar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RAN como RDM hayan tenido interrupciones durante la verificación y esta relación no sea del todo cierta.
<b>Nombre completo</b>	Nombres del RE y RAN que participan en la verificación.
<b>Puesto (rol)</b>	Puestos o roles de los participantes en la verificación.
<b>Elemento a verificar</b>	Se deberán listar todos los elementos que se deban verificar usando los Criterios de Verificación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Especificación de Requerimientos, Incorporación de todos los requerimientos funcionales, etc. Se recomienda incluir todos los elementos que forman el documento del Registro de Rastreo y aplicar los criterios definidos, por ejemplo: Redacción, Formato de plantilla, Incorporación de los elementos de diseño, etc.
<b>Descripción del defecto encontrado</b>	El RE y RAN deben describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento describe claramente la trazabilidad de cada requerimiento funcional”. Un ejemplo de cómo describir un defecto puede ser: “El RF 19.5 no está siendo rastreado.”
<b>Fecha de aprobación</b>	Fecha en la que RE y RAN se aseguran que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.



## Instrucciones de llenado de la plantilla

Campo	Descripción
<b>Fecha del reporte</b>	Fecha en la que tanto el RDM y el CL y US empiezan a realizar la validación del documento. Es importante considerar que cada validación realizada puede generar un Reporte de Validación si es necesario.
<b>Lugar</b>	Lugar en donde el RDM y el CL y US realizan la validación del documento.
<b>Hora de inicio</b>	Hora exacta en la que comienza la validación del documento.
<b>Hora de fin</b>	Hora exacta en la que termina la validación del documento.
<b>Duración total</b>	Cálculo numérico (en horas) para expresar el tiempo total que RDM y CL y US tardaron en validar el documento. Es obvio que en la mejor de las situaciones este cálculo sería de la forma Duración total = Hora de fin – Hora de inicio, sin embargo es posible que tanto RDM como CL y US hayan tenido interrupciones durante la validación y esta relación no sea del todo cierta.
<b>Nombre completo</b>	Nombres del RDM, CL y US que participan en la validación.
<b>Puesto (rol)</b>	Puestos o roles de los participantes en la validación.
<b>Elemento a validar</b>	Se deberán listar todos los elementos que se deban validar usando los Criterios de Validación de la documentación de procesos. Por ejemplo: Consistencia o adherencia con la Especificación de Requerimientos, Recursos requeridos, etc. Se recomienda incluir todos los elementos que forman el documento del Análisis y Diseño y aplicar los criterios definidos para su validación.
<b>Descripción del defecto encontrado</b>	El RDM debe describir detalladamente cuál es el defecto encontrado. En caso de no tener defecto entonces deben describir la razón por la cual no existen defectos. Por ejemplo: “Sin defectos dado que el documento cubre claramente los requerimientos establecidos por el cliente en relación al sistema”. Un ejemplo de cómo describir un defecto puede ser: “No se ha incluido un diagrama de secuencia que represente la cancelación de una venta.”
<b>Fecha de aprobación</b>	Fecha en la que el RDM se asegura que el defecto encontrado fue corregido, en caso de que no se hayan encontrado defectos se puede poner la fecha inicial.

## Plantilla para Plan de Pruebas de Integración: DMS Actividad A.3.13

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
---------------	---------	------------------------	-------	---------------------	-------

<b>Nombre del subsistema</b>	[Inserte aquí el texto]				<b>Versión:</b>	[Rev]
<i>Componentes y subsistemas que lo integran</i>						
<b>Componente 1</b>	[Inserte aquí el texto]	<b>Versión</b>	[Rev]	<b>Implementador</b>	[Inserte aquí el texto]	
<b>Componente 2</b>	[Inserte aquí el texto]	<b>Versión</b>	[Rev]	<b>Implementador</b>	[Inserte aquí el texto]	
<b>Componente 3</b>	[Inserte aquí el texto]	<b>Versión</b>	[Rev]	<b>Implementador</b>	[Inserte aquí el texto]	
<b>Componente n</b>	[Inserte aquí el texto]	<b>Versión</b>	[Rev]	<b>Implementador</b>	[Inserte aquí el texto]	
<i>Requerimientos funcionales cubiertos por el subsistema</i>						
<b>Referencia</b>	<b>Descripción</b>					
[Rev]	[Inserte aquí el texto]					
[Rev]	[Inserte aquí el texto]					
[Rev]	[Inserte aquí el texto]					
[Rev]	[Inserte aquí el texto]					
[Rev]	[Inserte aquí el texto]					
[Rev]	[Inserte aquí el texto]					
[Rev]	[Inserte aquí el texto]					
[Rev]	[Inserte aquí el texto]					
[Rev]	[Inserte aquí el texto]					
<i>Casos de prueba</i>						
<b>Escenario relacionado</b>	[Rev]					
<b># de caso de prueba</b>	[Rev]	<b>Descripción:</b>	[Inserte aquí el texto]			
<b>Procedimiento de prueba</b>	[Inserte aquí el texto]					
<b>Entrada</b>	[Inserte aquí el texto]					
<b>Salida esperada</b>	[Inserte aquí el texto]					
<b>Errores encontrados</b>	[Inserte aquí el texto]					
<b>Sugerencias de corrección</b>	[Inserte aquí el texto]					
<b># de caso de prueba</b>	[Rev]	<b>Descripción:</b>	[Inserte aquí el texto]			
<b>Procedimiento de prueba</b>	[Inserte aquí el texto]					
<b>Entrada</b>	[Inserte aquí el texto]					
<b>Salida esperada:</b>	[Inserte aquí el texto]					
<b>Errores encontrados:</b>	[Inserte aquí el texto]					
<b>Sugerencias de corrección:</b>	[Inserte aquí el texto]					
<b># de caso de prueba:</b>	[Rev]	<b>Descripción:</b>	[Inserte aquí el texto]			
<b>Procedimiento de prueba</b>	[Inserte aquí el texto]					
<b>Entrada:</b>	[Inserte aquí el texto]					
<b>Salida esperada:</b>	[Inserte aquí el texto]					
<b>Errores encontrados:</b>	[Inserte aquí el texto]					
<b>Sugerencias de corrección:</b>	[Inserte aquí el texto]					
<b>Escenario relacionado:</b>	[Rev]					
<b># de caso de prueba:</b>	[Rev]	<b>Descripción:</b>	[Inserte aquí el texto]			
<b>Procedimiento de prueba</b>	[Inserte aquí el texto]					
<b>Entrada:</b>	[Inserte aquí el texto]					
<b>Salida esperada:</b>	[Inserte aquí el texto]					

<b>Errores encontrados:</b>	[Inserte aquí el texto]		
<b>Sugerencias de corrección:</b>	[Inserte aquí el texto]		
<b># de caso de prueba:</b>	[Rev]	<b>Descripción:</b>	[Inserte aquí el texto]
<b>Procedimiento de prueba</b>	[Inserte aquí el texto]		
<b>Entrada:</b>	[Inserte aquí el texto]		
<b>Salida esperada:</b>	[Inserte aquí el texto]		
<b>Errores encontrados:</b>	[Inserte aquí el texto]		
<b>Sugerencias de corrección:</b>	[Inserte aquí el texto]		
<b># de caso de prueba:</b>	[Rev]	<b>Descripción:</b>	[Inserte aquí el texto]
<b>Procedimiento de prueba</b>	[Inserte aquí el texto]		
<b>Entrada:</b>	[Inserte aquí el texto]		
<b>Salida esperada:</b>	[Inserte aquí el texto]		
<b>Errores encontrados:</b>	[Inserte aquí el texto]		
<b>Sugerencias de corrección:</b>	[Inserte aquí el texto]		

<i>Cobertura de requisitos funcionales</i>					
Escenario con datos	Entrada	Salida esperada	Salida obtenida	Error	Caso de prueba
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]

<b>Nombre del subsistema</b>	[Inserte aquí el texto]	<b>Versión:</b>	[Rev]
------------------------------	-------------------------	-----------------	-------

*Componentes y subsistemas que lo integran*

Componente	Descripción	Versión	Implementador
<b>Componente 1</b>	[Inserte aquí el texto]	[Rev]	[Inserte aquí el texto]
<b>Componente 2</b>	[Inserte aquí el texto]	[Rev]	[Inserte aquí el texto]
<b>Componente 3</b>	[Inserte aquí el texto]	[Rev]	[Inserte aquí el texto]
<b>Componente n</b>	[Inserte aquí el texto]	[Rev]	[Inserte aquí el texto]

*Requerimientos funcionales cubiertos por el subsistema*

Referencia	Descripción
[Rev]	[Inserte aquí el texto]

*Casos de prueba*

<b>Escenario relacionado</b>	[Rev]
------------------------------	-------



[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]
[Rev]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]	[Rev]

### Interacción de la Integración

<b>Subsistemas involucrados</b>	[Inserte aquí el texto]
<b>Entrada</b>	[Inserte aquí el texto]
<b>Resultado esperado</b>	[Inserte aquí el texto]
<b>Resultado obtenido</b>	[Inserte aquí el texto]
<b>Errores encontrados</b>	[Inserte aquí el texto]
<b>Sugerencias de corrección</b>	[Inserte aquí el texto]

<b>Subsistemas involucrados</b>	[Inserte aquí el texto]
<b>Entrada</b>	[Inserte aquí el texto]
<b>Resultado esperado</b>	[Inserte aquí el texto]
<b>Resultado obtenido</b>	[Inserte aquí el texto]
<b>Errores encontrados</b>	[Inserte aquí el texto]
<b>Sugerencias de corrección</b>	[Inserte aquí el texto]

<b>Subsistemas involucrados</b>	[Inserte aquí el texto]
<b>Entrada</b>	[Inserte aquí el texto]
<b>Resultado esperado</b>	[Inserte aquí el texto]
<b>Resultado obtenido</b>	[Inserte aquí el texto]
<b>Errores encontrados</b>	[Inserte aquí el texto]
<b>Sugerencias de corrección</b>	[Inserte aquí el texto]

### Evaluación

<b>Evaluación final</b>	[Inserte aquí el texto]
-------------------------	-------------------------

## Instrucciones de llenado de la plantilla

Campo	Descripción
<b>Descripción de pruebas de integración</b>	
<b>Fecha</b>	Fecha en la que se redacta/ejecuta el plan de pruebas.
<b>Hora de inicio</b>	Hora exacta en la que comienza la redacción/ejecución del plan de pruebas.
<b>Hora de fin</b>	Hora exacta en la que termina la redacción/ejecución del plan de pruebas.
<b>Nombre del subsistema</b>	Mediante su referencia deberán listarse todos los requerimientos que fueron aceptados en la <i>Especificación de Requerimientos</i> y que se relacionan con el subsistema descrito.
<b>Componentes y subsistemas que lo integran</b>	
<b>Componente</b>	Cada uno de los componentes que integran al subsistema de acuerdo a lo establecido en el <i>Análisis y Diseño</i> de la iteración (ciclo).
<b>Versión</b>	Indicador numérico para llevar el control de versiones cuando se empiece a modificar el componente, si fuera necesario.
<b>Implementador</b>	Nombre del programador que fue asignado para implementar el componente.
<b>Requerimientos funcionales cubiertos por el subsistema</b>	
<b>Referencia</b>	Se hace referencia a las funcionalidades identificadas en la <i>Especificación de Requerimientos</i> y que fueran relacionadas con los subsistemas en el <i>Análisis y Diseño</i> . Concretamente, se debe colocar la referencia que se le asignó al requerimiento desde la <i>Especificación de Requerimientos</i> .
<b>Descripción</b>	Descripción del requerimiento.
<b>Casos de prueba</b>	
<b>Escenario relacionado</b>	Se refiere a los escenarios identificados con los diagramas de secuencia que mostraron la relación de los subsistemas definidos en el <i>Análisis y Diseño</i> .
<b># de caso de prueba</b>	Número que se le asigna a cada caso de prueba (p.ej., 1, 2, 3, 4, ..., n).
<b>Procedimiento de prueba</b>	Descripción de los procedimientos de prueba para la realización de la funcionalidad de cada # de caso de prueba.
<b>Entrada</b>	Se describen los datos de prueba (condiciones de entrada) necesarios para realizar la prueba.
<b>Salida esperada</b>	Se describe la salida esperada para las condiciones de entrada especificadas en la entrada.
<b>Salida obtenida</b>	El RPR describe la salida real obtenida para cada valor de entrada especificado anteriormente.
<b>Errores encontrados</b>	El RPR describe los errores encontrados, si es posible la ubicación de cada uno y su gravedad e impacto en el sistema.
<b>Sugerencias de corrección</b>	El RPR proporciona sugerencias de corrección para los errores encontrados, si esto es posible.
<b>Cobertura de requisitos funcionales</b>	
<b>Escenario con datos</b>	Se hace referencia a cada # de caso de prueba planteado anteriormente, indicando la funcionalidad a probar y detallando los casos de prueba con datos asociados.
<b>Entrada</b>	Datos reales que debe recibir el sistema (p.ej., CONTRASEÑA: inválida, Nombre: Juan Pérez, Fecha de préstamo: 45/89/2130, CONTRASEÑA: válida, etc.).
<b>Salida esperada</b>	Respuesta esperada del sistema de acuerdo a la entrada recibida (p.ej., Entrada->CONTRASEÑA: inválida, Salida->El sistema envía mensaje de error y solicita de nuevo la autenticación).
<b>Salida obtenida</b>	El RPR describe la respuesta real de acuerdo a la entrada recibida (p.ej., Entrada->CONTRASEÑA: inválida, Salida->El sistema no hace nada).
<b>Error</b>	El RPR describe a detalle los errores encontrados, si fuera el caso.
<b>Caso de prueba</b>	Referencia al # de caso de prueba con el que se relaciona la prueba sobre los requisitos funcionales.
<b>Interacción de la integración</b>	
<b>Subsistemas involucrados</b>	Se debe representar en la matriz de pruebas la relación uno-a-uno entre los diferentes subsistemas que componen la arquitectura presentada en <i>Análisis y Diseño</i> .
<b>Entrada</b>	Se describen los valores utilizados para realizar la interacción entre los subsistemas.
<b>Resultado esperado</b>	Se describe el resultado esperado para cada valor de entrada especificado anteriormente.

<b>Resultado obtenido</b>	El RPR describe el resultado real obtenido para cada valor de entrada especificado anteriormente.
<b>Errores encontrados</b>	El RPR describe los errores encontrados, si es posible la ubicación de cada uno y su gravedad e impacto en el sistema.
<b>Sugerencias de corrección</b>	El RPR brinda sugerencias de corrección para los errores encontrados, si esto es posible
<i>Evaluación</i>	
<b>Evaluación final</b>	El RPR evalúa en forma global el comportamiento de los subsistemas y establece el estado del mismo utilizando: no aprobado (cuando las pruebas a los subsistemas no son satisfechas en su totalidad) o aprobado (cuando las pruebas a los subsistemas son satisfechas en su totalidad).

## Plantilla para la Minuta de Definición de Protocolo de Entrega: APE Actividad A.1.3

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
<b>Objetivo de la reunión</b>	El objetivo de la reunión es obtener la minuta que defina los términos de la entrega del producto final y evidencie que se ha definido, de manera conjunta con el Cliente, el <i>Protocolo de Entrega</i> de cada uno de los entregables especificados en la <i>Descripción del Proyecto</i> .				
<b>Participantes de la reunión</b>	RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos. RDM como responsable del proceso de Desarrollo y Mantenimiento de Software. Cliente como receptor directo de cada uno de los entregables.				
<b>Seguimiento de compromisos pendientes</b>	[Inserte aquí el texto]				
<b>Puntos tratados</b>	[Inserte aquí el texto]				
<b>Acuerdos obtenidos</b>	[Inserte aquí el texto]				
<b>Asuntos pendientes por resolver</b>	[Inserte aquí el texto]				
<b>Aprobación</b>	Los presentes aseguran haber entendido el proyecto y se comprometen a cumplir con los entregables en tiempo y forma de acuerdo a la Descripción del Proyecto y el Plan de Desarrollo.  [Inserte aquí el texto]				

## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha</b>	Fecha en la que se realiza la reunión.
<b>Hora de inicio</b>	Hora exacta en la que comienza la reunión.
<b>Hora de fin</b>	Hora exacta en la que termina la reunión.
<b>Objetivo de la reunión</b>	El objetivo de la reunión es alcanzar un acuerdo en relación a los entregables que se han comprometido y que cumplen con lo descrito en la <i>Descripción del Proyecto</i> .
<b>Participantes de la reunión</b>	Los roles de los participantes en la reunión que han sido definidos como RAPE, RDM, y Cliente. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
<b>Seguimiento de compromisos pendientes</b>	Es responsabilidad del RAPE documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
<b>Puntos tratados</b>	El RDM debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo: <ol style="list-style-type: none"> <li>1. Pase de lista.</li> <li>2. Revisión de la Descripción del Proyecto.</li> <li>3. Ruegos y preguntas.</li> <li>4. Obtención de compromiso de todos los participantes.</li> <li>5. Firma de minuta.</li> <li>6. Cierre de la reunión.</li> </ol>
<b>Acuerdos obtenidos</b>	El RAPE debe documentar con una lista los acuerdos finales o compromisos, estableciendo que los responsables y fechas que fueron establecidos por la Descripción del Proyecto son aceptados.
<b>Asuntos pendientes por resolver</b>	El RAPE debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
<b>Aprobación</b>	El RAPE debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.

## Plantilla para la Minuta en el Establecimiento de Tiempos Estimados: APE Actividad A.1.6

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
<b>Objetivo de la reunión</b>	El objetivo de la reunión es evidenciar que se estableció en consenso el tiempo estimado para desarrollar cada actividad identificada en el <i>Plan de Desarrollo</i> . Si se trata del Nivel 2 de madurez, es necesario evidenciar que se utilizó información histórica para establecer los tiempos de cada actividad. Por otro lado, si se trata del Nivel 4 de madurez es necesario establecer también las Metas Cuantitativas para el proyecto.				
<b>Participantes de la reunión</b>	RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos. RDM como responsable del proceso de Desarrollo y Mantenimiento de Software.				
<b>Seguimiento de compromisos pendientes</b>	[Inserte aquí el texto]				
<b>Puntos tratados</b>	[Inserte aquí el texto]				
<b>Acuerdos obtenidos</b>	[Inserte aquí el texto]				
<b>Asuntos pendientes por resolver</b>	[Inserte aquí el texto]				
<b>Aprobación</b>	Los presentes aseguran haber entendido el proyecto y se comprometen a cumplir con los tiempos estimados de acuerdo a lo establecido en el Plan de Desarrollo.  [Inserte aquí el texto]				

## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha</b>	Fecha en la que se realiza la reunión.
<b>Hora de inicio</b>	Hora exacta en la que comienza la reunión.
<b>Hora de fin</b>	Hora exacta en la que termina la reunión.
<b>Objetivo de la reunión</b>	El objetivo de la reunión es establecer a través del consenso, y en base a datos históricos de proyectos desarrollados anteriormente, tiempos estimados para las actividades identificadas en el Plan de Desarrollo.
<b>Participantes de la reunión</b>	Los roles de los participantes en la reunión que han sido definidos como RAPE, y RDM. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
<b>Seguimiento de compromisos pendientes</b>	Es responsabilidad del RAPE documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
<b>Puntos tratados</b>	El RDM debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo: <ol style="list-style-type: none"> <li>1. Pase de lista.</li> <li>2. Revisión de la Descripción del Proyecto.</li> <li>3. Ruegos y preguntas.</li> <li>4. Obtención de compromiso de todos los participantes.</li> <li>5. Firma de minuta.</li> <li>6. Cierre de la reunión.</li> </ol>
<b>Acuerdos obtenidos</b>	El RAPE debe documentar con una lista los acuerdos finales o compromisos, estableciendo que los responsables y fechas que fueron establecidos por el Plan de Desarrollo son aceptados.
<b>Asuntos pendientes por resolver</b>	El RAPE debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
<b>Aprobación</b>	El RAPE debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.

## Plantilla de la Minuta para Conformar al Equipo de Trabajo: APE Actividad A.1.8

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
<b>Objetivo de la reunión</b>	El objetivo de esta minuta es evidenciar que se conformó el equipo de trabajo, a través de la asignación de roles y responsabilidades basándose en la Descripción del Proyecto.				
<b>Participantes de la reunión</b>	RGPY como autoridad y responsable de la gestión del proyecto específico. RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos.				
<b>Seguimiento de compromisos pendientes</b>	[Inserte aquí el texto]				
<b>Puntos tratados</b>	[Inserte aquí el texto]				
<b>Acuerdos obtenidos</b>	[Inserte aquí el texto]				
<b>Asuntos pendientes por resolver</b>	[Inserte aquí el texto]				
<b>Aprobación</b>	<p style="text-align: center;">Los presentes aseguran haber entendido el proyecto y se comprometen a cumplir con los roles y responsabilidades asignados de acuerdo a lo establecido en el Plan de Desarrollo.</p> <p>[Inserte aquí el texto]</p>				

## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha</b>	Fecha en la que se realiza la reunión.
<b>Hora de inicio</b>	Hora exacta en la que comienza la reunión.
<b>Hora de fin</b>	Hora exacta en la que termina la reunión.
<b>Objetivo de la reunión</b>	El objetivo de la reunión es establecer a través del consenso, y en base a datos históricos de proyectos desarrollados anteriormente, el equipo de trabajo.
<b>Participantes de la reunión</b>	Los roles de los participantes en la reunión que han sido definidos como RGPY y RAPE. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
<b>Seguimiento de compromisos pendientes</b>	Es responsabilidad del RGPY documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
<b>Puntos tratados</b>	El RGPY debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo: <ol style="list-style-type: none"> <li>1. Pase de lista.</li> <li>2. Revisión de la Descripción del Proyecto.</li> <li>3. Ruegos y preguntas.</li> <li>4. Obtención de compromiso de todos los participantes.</li> <li>5. Firma de minuta.</li> <li>6. Cierre de la reunión.</li> </ol>
<b>Acuerdos obtenidos</b>	El RGPY debe documentar con una lista los acuerdos finales o compromisos, estableciendo los integrantes del equipo.
<b>Asuntos pendientes por resolver</b>	El RAPE debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
<b>Aprobación</b>	El RGPY debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.

## Plantilla de la Minuta Para Identificar Riesgos: APE Actividad A.1.11

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
<b>Objetivo de la reunión</b>	El objetivo de esta minuta es evidenciar que identificaron, describieron y evaluaron los riesgos en conjunto con el RGPY, RAPE y RDM.				
<b>Participantes de la reunión</b>	RGPY como autoridad y responsable de la gestión del proyecto específico. RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos. RDM como responsable del proceso de Desarrollo y Mantenimiento de Software.				
<b>Seguimiento de compromisos pendientes</b>	[Inserte aquí el texto]				
<b>Puntos tratados</b>	[Inserte aquí el texto]				
<b>Acuerdos obtenidos</b>	[Inserte aquí el texto]				
<b>Asuntos pendientes por resolver</b>	[Inserte aquí el texto]				
<b>Aprobación</b>	<p style="text-align: center;">Los presentes aseguran haber entendido el proyecto y se comprometen a actuar correctamente de acuerdo a los riesgos identificados y establecidos en el Plan de Desarrollo.</p> <p>[Inserte aquí el texto]</p>				

## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha</b>	Fecha en la que se realiza la reunión.
<b>Hora de inicio</b>	Hora exacta en la que comienza la reunión.
<b>Hora de fin</b>	Hora exacta en la que termina la reunión.
<b>Objetivo de la reunión</b>	El objetivo de la reunión es identificar, describir y evaluar a través del consenso, y en base a datos históricos de proyectos desarrollados anteriormente, los riesgos que pueden afectar al proyecto. Tales riesgos deben contemplar riesgos relacionados con el equipo de trabajo, incluyendo al cliente y a los usuarios, los riesgos con la tecnología o la metodología, los riesgos con la gestión del proyecto (costo, tiempo, alcance y recursos) o riesgos externos al proyecto.
<b>Participantes de la reunión</b>	Los roles de los participantes en la reunión que han sido definidos como RGPY, RAPE y RDM. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
<b>Seguimiento de compromisos pendientes</b>	Es responsabilidad del RGPY documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
<b>Puntos tratados</b>	El RGPY debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo: <ol style="list-style-type: none"> <li>1. Pase de lista.</li> <li>2. Revisión de la Descripción del Proyecto.</li> <li>3. Ruegos y preguntas.</li> <li>4. Obtención de compromiso de todos los participantes.</li> <li>5. Firma de minuta.</li> <li>6. Cierre de la reunión.</li> </ol>
<b>Acuerdos obtenidos</b>	El RGPY debe documentar con una lista los acuerdos finales o compromisos, estableciendo los integrantes del equipo.
<b>Asuntos pendientes por resolver</b>	El RAPE debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
<b>Aprobación</b>	El RGPY debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.

## Plantilla para el Plan del Proyecto: APE Actividad A.1.12

### 1. Introducción

[Inserte aquí el texto]

*Es necesario definir cuál es el propósito de este plan. Es decir, se debe proporcionar una justificación clara de por qué dicho documento será utilizado como guía para la ejecución y control del proyecto. Adicionalmente, es necesario versionar este documento en caso de que se requiera manejar diferentes versiones del mismo para cada proyecto.*

### 2. Ciclos y actividades

[Inserte aquí el texto]

*Es necesario que se especifique, de acuerdo al modelo de desarrollo de software seleccionado, el número de ciclos y las actividades específicas para producir los entregables y sus componentes de acuerdo a la Descripción del Proyecto. Se deberán incluir las actividades necesarias para llevar a cabo las revisiones de los productos.*

### 3. Tiempo estimado

[Inserte aquí el texto]

*Es necesario especificar el tiempo total estimado para el proyecto, considerando el grado de dificultad asignado a los casos de uso obtenidos en la Especificación de Requerimientos.*

### 4. Plan de adquisiciones y capacitación

[Inserte aquí el texto]

*Esta sección debe mostrar la relación de recursos humanos, capacitación, materiales, equipo y herramientas necesarios para la ejecución del proyecto. Será necesario incluir una tabla de adquisiciones y capacitación o hacer referencia al archivo que la contenga. Adicionalmente, es necesario versionar este documento en caso de que se requiera manejar diferentes versiones del mismo para cada proyecto.*

### 5. Equipo de trabajo

[Inserte aquí el texto]

*Es necesario especificar, de manera general, el tipo de recursos y la cantidad que será requerida para el proyecto. Los roles y responsabilidades deben tomarse de la Descripción del Proyecto y se deberá incluir a los subcontratistas en caso de que aplique. Es posible que aquí mismo se especifiquen de una vez los nombres de las personas que cubrirán los roles o bien especificarlos en el Plan de Desarrollo]*

Rol	Responsabilidad	Cantidad
[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]
[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]
[Inserte aquí el texto]	[Inserte aquí el texto]	[Rev]

## 6. Costo estimado

[Inserte aquí el texto]

*Es necesario especificar el costo laboral estimado para el proyecto considerando el número de horas y recursos asignados al proyecto, además de costos no laborales (por ejemplo, viáticos, compra de licencias, renta de equipo, etc.). Es importante recalcar que todos los costos deben estar debidamente justificados para demostrar su coherencia para con el proyecto.*

## 7. Calendario

[Inserte aquí el texto]

*Es necesario también asignar fechas de inicio y fin a cada una de las actividades, tomando en cuenta los recursos asignados, la secuencia y dependencia de las actividades. Esta disposición de actividades debe incluir fechas de entrega, o puntos de revisión y control propuestos para el proyecto. Es posible también incluir aquellas actividades administrativas que tengan que ver con el cliente. Vale la pena mencionar que las actividades del calendario son aquellas que permitirán generar los entregables propuestos en el Protocolo de Entrega. Adicionalmente, es necesario versionar este documento en caso de que se requiera manejar diferentes versiones del mismo para cada proyecto.*

## 8. Plan de manejo de riesgos

[Inserte aquí el texto]

*Una vez que se ha levantado la Minuta para Identificar Riesgos es necesario que aquí dichos riesgos sean definidos de manera precisa y que se establezca su nivel de impacto sobre el proyecto junto con las acciones correctivas que se tienen que desarrollar para su manejo. Adicionalmente, es necesario versionar este documento en caso de que se requiera manejar diferentes versiones del mismo para cada proyecto.*

## 9. Protocolo de entrega

[Inserte aquí el texto]

*[Se deberá definir en conjunto con el cliente la forma en que se entrega y reciben los documentos, componentes, etc. generados en el proyecto]*

Entregable	Fase	¿Quién revisa?	¿Quién aprueba?	Medio	Condiciones de entrega o Checklist
[Inserte aquí el texto]					
[Inserte aquí el texto]					
[Inserte aquí el texto]					
[Inserte aquí el texto]					

## Plantilla para generar el Plan de Desarrollo: APE Actividad A.1.13

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
<b>Objetivo de la reunión</b>	El objetivo de esta minuta es evidenciar que se generó el Plan de Desarrollo en función del Plan de Proyecto, o que se actualizó antes de iniciar un nuevo ciclo en conjunto con el RDM. Esta minuta no será necesaria en caso que la organización cuente con alguna otra forma de evidenciar que esta actividad la hicieron en conjunto el RAPE y el RDM.				
<b>Participantes de la reunión</b>	RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos. RDM como responsable del proceso de Desarrollo y Mantenimiento de Software.				
<b>Seguimiento de compromisos pendientes</b>	[Inserte aquí el texto]				
<b>Puntos tratados</b>	[Inserte aquí el texto]				
<b>Acuerdos obtenidos</b>	[Inserte aquí el texto]				
<b>Asuntos pendientes por resolver</b>	[Inserte aquí el texto]				
<b>Aprobación</b>	Los presentes aseguran haber entendido el proyecto y se comprometen a cumplir con lo establecido en el Plan de Desarrollo.  [Inserte aquí el texto]				

## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha</b>	Fecha en la que se realiza la reunión.
<b>Hora de inicio</b>	Hora exacta en la que comienza la reunión.
<b>Hora de fin</b>	Hora exacta en la que termina la reunión.
<b>Objetivo de la reunión</b>	El objetivo de la reunión es generar a través del consenso el Plan de Desarrollo en función del Plan del Proyecto o actualizarlo una versión anterior que se tenga antes de comenzar un nuevo ciclo.
<b>Participantes de la reunión</b>	Los roles de los participantes en la reunión que han sido definidos como RAPE y RDM. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
<b>Seguimiento de compromisos pendientes</b>	Es responsabilidad del RAPE documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
<b>Puntos tratados</b>	El RAPE debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo: <ol style="list-style-type: none"> <li>1. Pase de lista.</li> <li>2. Revisión de la Descripción del Proyecto.</li> <li>3. Ruegos y preguntas.</li> <li>4. Obtención de compromiso de todos los participantes.</li> <li>5. Firma de minuta.</li> <li>6. Cierre de la reunión.</li> </ol>
<b>Acuerdos obtenidos</b>	El RAPE debe documentar con una lista los acuerdos finales o compromisos, estableciendo los integrantes del equipo.
<b>Asuntos pendientes por resolver</b>	El RAPE debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
<b>Aprobación</b>	El RAPE debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.

## Plantilla para el Plan de Desarrollo: APE Actividad A.1.13

### 1. Introducción

*Este documento es utilizado como guía para la ejecución del desarrollo o mantenimiento de software.*

### 2. Descripción del producto y entregables

[Inserte aquí el texto]

*Es necesario extraer la información descrita en la Descripción del Proyecto, específicamente en los Entregables.*

### 3. Equipo de trabajo

[Inserte aquí el texto]

*Es necesario especificar el nombre, rol y responsabilidad de todas las personas asignadas al proyecto.*

Nombre	Rol	Responsabilidad
[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]
[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]
[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]
[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]

### 4. Calendario

[Inserte aquí el texto]

*Una vez identificadas y documentadas las relaciones y dependencias entre las actividades, se debe definir el calendario considerando los tiempos establecidos en el Plan del Proyecto. De manera similar al Plan del Proyecto, este calendario debe contener las actividades a realizar con fechas de inicio y de fin, además de responsables. Adicionalmente, es necesario versionar este documento en caso de que se requiera manejar diferentes versiones del mismo para cada proyecto.*



## Plantilla de la Minuta para el Inicio Formal de un Nuevo Ciclo: APE Actividad A.1.18

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
<b>Objetivo de la reunión</b>	El objetivo de esta minuta es evidenciar que se dio inicio formal a un nuevo ciclo en conjunto con el RDM. Esta minuta no será necesaria en caso que la organización cuente con alguna otra forma de evidenciar que esta actividad la hicieron en conjunto el RAPE y el RDM.				
<b>Participantes de la reunión</b>	RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos. RDM como responsable del proceso de Desarrollo y Mantenimiento de Software.				
<b>Seguimiento de compromisos pendientes</b>	[Inserte aquí el texto]				
<b>Puntos tratados</b>	[Inserte aquí el texto]				
<b>Acuerdos obtenidos</b>	[Inserte aquí el texto]				
<b>Asuntos pendientes por resolver</b>	[Inserte aquí el texto]				
<b>Aprobación</b>	Los presentes aseguran haber entendido el proyecto y se comprometen a iniciar el nuevo ciclo de acuerdo a lo establecido en el Plan de Desarrollo.  [Inserte aquí el texto]				

## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha</b>	Fecha en la que se realiza la reunión.
<b>Hora de inicio</b>	Hora exacta en la que comienza la reunión.
<b>Hora de fin</b>	Hora exacta en la que termina la reunión.
<b>Objetivo de la reunión</b>	El objetivo de la reunión es aceptar, a través del consenso, el inicio de un nuevo ciclo del proyecto una vez que se haya asegurado el cumplimiento de las condiciones iniciales del mismo.
<b>Participantes de la reunión</b>	Los roles de los participantes en la reunión que han sido definidos como RAPE y RDM. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
<b>Seguimiento de compromisos pendientes</b>	Es responsabilidad del RAPE documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
<b>Puntos tratados</b>	El RAPE debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo: <ol style="list-style-type: none"> <li>1. Pase de lista.</li> <li>2. Revisión de la Descripción del Proyecto.</li> <li>3. Ruegos y preguntas.</li> <li>4. Obtención de compromiso de todos los participantes.</li> <li>5. Firma de minuta.</li> <li>6. Cierre de la reunión.</li> </ol>
<b>Acuerdos obtenidos</b>	El RAPE debe documentar con una lista los acuerdos finales o compromisos, estableciendo los integrantes del equipo.
<b>Asuntos pendientes por resolver</b>	El RAPE debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
<b>Aprobación</b>	El RAPE debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.

## Plantilla de la Minuta para el Seguimiento al Plan de Adquisiciones y Capacitación: APE Actividad A.2.4

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
<b>Objetivo de la reunión</b>	El objetivo de esta minuta es evidenciar que se le dio seguimiento al Plan de Adquisiciones y Capacitación, que la asignación de recursos humanos o subcontratistas fue aceptada o rechazada, y que se distribuyeron los recursos a los miembros del equipo de trabajo para que pudieran llevar a cabo sus actividades. Esta actividad se debe realizar en conjunto con el RAPE, RDM y RSC.				
<b>Participantes de la reunión</b>	RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos. RDM como responsable del proceso de Desarrollo y Mantenimiento de Software. RSC como Responsable del Subcontrato.				
<b>Seguimiento de compromisos pendientes</b>	[Inserte aquí el texto]				
<b>Puntos tratados</b>	[Inserte aquí el texto]				
<b>Acuerdos obtenidos</b>	[Inserte aquí el texto]				
<b>Asuntos pendientes por resolver</b>	[Inserte aquí el texto]				
<b>Aprobación</b>	Los presentes aseguran haber entendido el proyecto y se comprometen a dar seguimiento a lo establecido en el Plan de Adquisiciones y Capacitación.  [Inserte aquí el texto]				

## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha</b>	Fecha en la que se realiza la reunión.
<b>Hora de inicio</b>	Hora exacta en la que comienza la reunión.
<b>Hora de fin</b>	Hora exacta en la que termina la reunión.
<b>Objetivo de la reunión</b>	El objetivo de la reunión es establecer a través del consenso, los mecanismos para dar seguimiento al Plan de Adquisiciones y Capacitación.
<b>Participantes de la reunión</b>	Los roles de los participantes en la reunión que han sido definidos como RAPE, RDM y RSC. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
<b>Seguimiento de compromisos pendientes</b>	Es responsabilidad del RAPE documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
<b>Puntos tratados</b>	El RAPE debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo: <ol style="list-style-type: none"> <li>1. Pase de lista.</li> <li>2. Revisión de la Descripción del Proyecto.</li> <li>3. Ruegos y preguntas.</li> <li>4. Obtención de compromiso de todos los participantes.</li> <li>5. Firma de minuta.</li> <li>6. Cierre de la reunión.</li> </ol>
<b>Acuerdos obtenidos</b>	El RAPE debe documentar con una lista los acuerdos finales o compromisos, estableciendo los integrantes del equipo.
<b>Asuntos pendientes por resolver</b>	El RAPE debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
<b>Aprobación</b>	El RAPE debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.

## Plantilla de la Minuta para la Relación de Subcontratistas: APE Actividad A.2.5

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
<b>Objetivo de la reunión</b>	El objetivo de esta minuta es evidenciar que se gestionará la relación con los subcontratistas, lo que implica planificar, revisar, y auditar las actividades, asegurando en todo momento la calidad de los productos o servicios contratados y el cumplimiento con los estándares y especificaciones acordadas. Esta actividad se debe realizar en conjunto con el RAPE y RSC.				
<b>Participantes de la reunión</b>	RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos. RSC como Responsable del subcontrato.				
<b>Seguimiento de compromisos pendientes</b>	[Inserte aquí el texto]				
<b>Puntos tratados</b>	[Inserte aquí el texto]				
<b>Acuerdos obtenidos</b>	[Inserte aquí el texto]				
<b>Asuntos pendientes por resolver</b>	[Inserte aquí el texto]				
<b>Aprobación</b>	Los presentes aseguran haber entendido el proyecto y se comprometen a cumplir con las actividades establecidas para gestionar la relación de los subcontratistas de acuerdo a lo establecido en el Plan de Adquisiciones y Capacitación.  [Inserte aquí el texto]				

## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha</b>	Fecha en la que se realiza la reunión.
<b>Hora de inicio</b>	Hora exacta en la que comienza la reunión.
<b>Hora de fin</b>	Hora exacta en la que termina la reunión.
<b>Objetivo de la reunión</b>	El objetivo de la reunión es establecer, a través del consenso, las actividades necesarias para manejar la relación con los proveedores externos (subcontratistas), lo que implica planificar, revisar y auditar las actividades, asegurando la calidad de los productos o servicios contratados y el cumplimiento con los estándares y especificaciones acordadas.
<b>Participantes de la reunión</b>	Los roles de los participantes en la reunión que han sido definidos como RAPE y RSC. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
<b>Seguimiento de compromisos pendientes</b>	Es responsabilidad del RAPE documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
<b>Puntos tratados</b>	El RAPE debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo: <ol style="list-style-type: none"> <li>1. Pase de lista.</li> <li>2. Revisión de la Descripción del Proyecto.</li> <li>3. Ruegos y preguntas.</li> <li>4. Obtención de compromiso de todos los participantes.</li> <li>5. Firma de minuta.</li> <li>6. Cierre de la reunión.</li> </ol>
<b>Acuerdos obtenidos</b>	El RAPE debe documentar con una lista los acuerdos finales o compromisos, estableciendo los integrantes del equipo.
<b>Asuntos pendientes por resolver</b>	El RAPE debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
<b>Aprobación</b>	El RAPE debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.

## Plantilla de la Minuta para establecer el Seguimiento a los Costos y Recursos y al Registro de Rastreo: APE Actividades A.2.7 y A.2.8

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
<b>Objetivo de la reunión</b>	El objetivo de esta minuta es evidenciar la realización de las actividades 2.7 y 2.8 del proceso de APE que se refieren a: <ol style="list-style-type: none"> <li>1. Registrar los Costos y Recursos reales del ciclo.</li> <li>2. Revisar el Registro de Rastreo desde los requisitos del usuario a través del ciclo.</li> </ol>				
<b>Participantes de la reunión</b>	RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos. RDM como responsable del proceso de Desarrollo y Mantenimiento de Software. ET como participantes activos del ciclo.				
<b>Seguimiento de compromisos pendientes</b>	[Inserte aquí el texto]				
<b>Puntos tratados</b>	[Inserte aquí el texto]				
<b>Acuerdos obtenidos</b>	[Inserte aquí el texto]				
<b>Asuntos pendientes por resolver</b>	[Inserte aquí el texto]				
<b>Aprobación</b>	Los presentes aseguran haber entendido el proyecto y se comprometen a registrar los costos y recursos del ciclo, y revisar el Registro de Rastreo acuerdo a lo establecido en el Plan de Desarrollo y Registro de Rastreo.  [Inserte aquí el texto]				

## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha</b>	Fecha en la que se realiza la reunión.
<b>Hora de inicio</b>	Hora exacta en la que comienza la reunión.
<b>Hora de fin</b>	Hora exacta en la que termina la reunión.
<b>Objetivo de la reunión</b>	El objetivo de la reunión es establecer, a través del consenso, mecanismos adecuados para el registro de costos y recursos de cada ciclo del proyecto. Además, los participantes deberán aceptar que el Registro de Rastreo sea continuamente revisado para actualizar su contenido.
<b>Participantes de la reunión</b>	Los roles de los participantes en la reunión que han sido definidos como RAPE, RDM, y ET. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
<b>Seguimiento de compromisos pendientes</b>	Es responsabilidad del RAPE documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
<b>Puntos tratados</b>	El RAPE debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo: <ol style="list-style-type: none"> <li>1. Pase de lista.</li> <li>2. Revisión de la Descripción del Proyecto.</li> <li>3. Ruegos y preguntas.</li> <li>4. Obtención de compromiso de todos los participantes.</li> <li>5. Firma de minuta.</li> <li>6. Cierre de la reunión.</li> </ol>
<b>Acuerdos obtenidos</b>	El RAPE debe documentar con una lista los acuerdos finales o compromisos, estableciendo los integrantes del equipo.
<b>Asuntos pendientes por resolver</b>	El RAPE debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
<b>Aprobación</b>	El RAPE debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.

## Plantilla de la Minuta para el Seguimiento a la Configuración de Software y al Plan de Proyecto: APE Actividades A.2.9 y A.2.10

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
<b>Objetivo de la reunión</b>	<p>El objetivo de esta minuta es evidenciar que se ha realizado el seguimiento de las actividades 2.9 y 2.10 del proceso de APE que se realizan en conjunto con del RDMS, concretamente:</p> <ol style="list-style-type: none"> <li>1. Revisar los productos generados durante el ciclo, que forman parte de la Configuración de Software.</li> <li>2. Recibir y analizar las Solicitudes de Cambio e incorporar los cambios aprobados en el Plan del Proyecto y en el Plan de Desarrollo. En caso de cambios a requisitos se incorporan al inicio de un nuevo ciclo.</li> </ol>				
<b>Participantes de la reunión</b>	<p>RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos. RDM como responsable del proceso de Desarrollo y Mantenimiento de Software.</p>				
<b>Seguimiento de compromisos pendientes</b>	[Inserte aquí el texto]				
<b>Puntos tratados</b>	[Inserte aquí el texto]				
<b>Acuerdos obtenidos</b>	[Inserte aquí el texto]				
<b>Asuntos pendientes por resolver</b>	[Inserte aquí el texto]				
<b>Aprobación</b>	<p>Los presentes aseguran haber entendido el proyecto y se comprometen a cumplir con las actividades de seguimiento de acuerdo a lo establecido en el Plan de Desarrollo y el Plan del Proyecto.</p> <p>[Inserte aquí el texto]</p>				

## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha</b>	Fecha en la que se realiza la reunión.
<b>Hora de inicio</b>	Hora exacta en la que comienza la reunión.
<b>Hora de fin</b>	Hora exacta en la que termina la reunión.
<b>Objetivo de la reunión</b>	El objetivo de la reunión es establecer, a través del consenso, los mecanismos necesarios para actualizar la información contenida en la Configuración de Software y en el Plan del Proyecto y en el Plan de Desarrollo de acuerdo a las Solicitudes de Cambio recibidas en el ciclo.
<b>Participantes de la reunión</b>	Los roles de los participantes en la reunión que han sido definidos como RAPE y RDM. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
<b>Seguimiento de compromisos pendientes</b>	Es responsabilidad del RAPE documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
<b>Puntos tratados</b>	El RAPE debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo: <ol style="list-style-type: none"> <li>7. Pase de lista.</li> <li>8. Revisión de la Descripción del Proyecto.</li> <li>9. Ruegos y preguntas.</li> <li>10. Obtención de compromiso de todos los participantes.</li> <li>11. Firma de minuta.</li> <li>12. Cierre de la reunión.</li> </ol>
<b>Acuerdos obtenidos</b>	El RAPE debe documentar con una lista los acuerdos finales o compromisos, estableciendo los integrantes del equipo.
<b>Asuntos pendientes por resolver</b>	El RAPE debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
<b>Aprobación</b>	El RAPE debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.

## Plantilla para la Minuta de Revisión entre Equipo de Trabajo y Cliente: APE Actividad A.2.11

<b>Fecha:</b>	[Fecha]	<b>Hora de inicio:</b>	[Rev]	<b>Hora de fin:</b>	[Rev]
<b>Objetivo de la reunión</b>	El objetivo de esta minuta es evidenciar la revisión del proyecto conduciendo reuniones con el Equipo de Trabajo y con el Cliente, generando minutas con puntos tratados y acuerdos tomados.				
<b>Participantes de la reunión</b>	RAPE como autoridad y responsable del proceso de Administración de Proyectos Específicos. ET como participantes activos del ciclo. CL como usuarios directos del producto final.				
<b>Seguimiento de compromisos pendientes</b>	[Inserte aquí el texto]				
<b>Puntos tratados</b>	[Inserte aquí el texto]				
<b>Acuerdos obtenidos</b>	[Inserte aquí el texto]				
<b>Asuntos pendientes por resolver</b>	[Inserte aquí el texto]				
<b>Aprobación</b>	Los presentes aseguran haber entendido el proyecto y se comprometen a aceptar lo establecido en el Plan de Desarrollo, en términos de entregables.  [Inserte aquí el texto]				

## Instrucciones de llenado de la plantilla

<b>Campo</b>	<b>Descripción</b>
<b>Fecha</b>	Fecha en la que se realiza la reunión.
<b>Hora de inicio</b>	Hora exacta en la que comienza la reunión.
<b>Hora de fin</b>	Hora exacta en la que termina la reunión.
<b>Objetivo de la reunión</b>	El objetivo de la reunión es conducir revisiones con el Equipo de Trabajo y el Cliente, generando como resultado final minutas con puntos que fueron tratados y los acuerdos a los que se llegó.
<b>Participantes de la reunión</b>	Los roles de los participantes en la reunión que han sido definidos como RAPE, ET y CL. Es importante aclarar que este campo únicamente requiere roles, los nombres de los participantes deberán ser agregados más adelante.
<b>Seguimiento de compromisos pendientes</b>	Es responsabilidad del RAPE documentar en esta parte los compromisos que son rescatados de reuniones pasadas. Dado que la minuta de lanzamiento se genera en una reunión inicial, es posible que estos compromisos no existan.
<b>Puntos tratados</b>	El RAPE debe documentar con una lista el proceso a través del cual se llegó a las decisiones tomadas en la reunión. Por ejemplo: <ol style="list-style-type: none"> <li>1. Pase de lista.</li> <li>2. Revisión de la Descripción del Proyecto.</li> <li>3. Ruegos y preguntas.</li> <li>4. Obtención de compromiso de todos los participantes.</li> <li>5. Firma de minuta.</li> <li>6. Cierre de la reunión.</li> </ol>
<b>Acuerdos obtenidos</b>	El RAPE debe documentar con una lista los acuerdos finales o compromisos, estableciendo los integrantes del equipo.
<b>Asuntos pendientes por resolver</b>	El RAPE debe listar todos los asuntos que necesitan mayor discusión o investigación, escalarlos a un siguiente nivel (al RGPY) o que por otra razón no pudieron ser resueltos, y establecer los siguientes pasos para resolver estos pendientes.
<b>Aprobación</b>	El RAPE debe asegurarse de recoger las firmas de todos los presentes editando en este campo de la minuta sus nombres y roles en relación al proyecto y que fueron definidos en el campo Participantes de la reunión.





## Plantilla para el Reporte de Seguimiento del Proyecto: APE Actividad A.3.3

Reporte de Seguimiento No. [Inserte aquí el texto]			
Datos generales			
<b>Nombre del Proyecto</b>	[Inserte aquí el texto]		
<b>Clave del Proyecto</b>	[Inserte aquí el texto]		
<b>Nombre del Responsable del Proyecto</b>	[Inserte aquí el texto]		
<b>Fecha del reporte</b>	[Fecha]	<b>Periodo comprendido</b>	[Fecha]

Registro de actividades							
Actividad del Plan de Proyecto	Fecha Inicio		Fecha Fin		Esfuerzo		Responsables
	Planeado	Real	Planeado	Real	Planeado	Real	
<i>Este registro de actividades puede llevarse en Microsoft Project. Si este fuera el caso, deberá hacerse referencia al archivo que contenga el registro de actividades del proyecto. Es necesario incluir las actividades del Plan de Manejo de Riesgos.</i>	[Fecha]	[Fecha]	[Fecha]	[Fecha]	[Fecha]	[Fecha]	[Inserte aquí el texto]

Mediciones				
	Planeado original	Re-planeado	Real	Observaciones
<b>Fecha Inicio</b>	[Inserte aquí el texto]			
<b>Fecha Final</b>	[Inserte aquí el texto]			
<b>Costo</b>	[Inserte aquí el texto]			
<b>Esfuerzo</b>	[Inserte aquí el texto]			

Control de Cambios			
Cambio	Tipo de cambio	Fecha de autorización	Esfuerzo
[Inserte aquí el texto]			
<b>Total</b>			[Inserte aquí el texto]

Defectos			
Encontrados en	Cantidad	Esfuerzo Ver/Val/Pru	Trabajo duplicado
Verificaciones	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]
Validaciones	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]
Pruebas	[Inserte aquí el texto]	[Inserte aquí el texto]	[Inserte aquí el texto]
<b>Total</b>	[Inserte aquí el texto]		

Tamaño de Productos	
Productos	Tamaño Real
[Inserte aquí el texto]	[Inserte aquí el texto]
[Inserte aquí el texto]	[Inserte aquí el texto]
[Inserte aquí el texto]	[Inserte aquí el texto]
[Inserte aquí el texto]	[Inserte aquí el texto]
[Inserte aquí el texto]	[Inserte aquí el texto]

## **9. Anexo D.- Acta de publicaciones**



## Project Management in Small-sized Software Enterprises: A Metamodeling-based approach

Garcia, I.<sup>1</sup>, Pacheco, C.<sup>1</sup>, Arcilla, M.<sup>2</sup>, and Sanchez, N.<sup>1</sup>

<sup>1</sup> Universidad Tecnológica de la Mixteca, Division de Estudios de Posgrado,  
Carretera a Acatlilma, 69000 Oaxaca, Mexico  
{ivan, leninca, nsanchez}@mixteco.utn.mx

<sup>2</sup> Universidad Nacional de Educación a Distancia, Escuela Técnica Superior de Ingenieros  
Informáticos, Ciudad Universitaria, 28040 Madrid, España  
marcilla@issi.uned.es

**Abstract.** Software development involves a unique effort that comprises managing many activities, resources, skills, and people to build a quality product. Thus, this effort is frequently seen from two different perspectives: the software development perspective and the project management perspective. Nowadays, any software enterprise that aims to develop high quality products should perform an adequate combination of both perspectives. However, such integration is generally not well addressed by the small-sized software enterprises due to the lack of knowledge, resources and time. In this sense, this paper introduces a metamodel to define a “lite” version of the project management process and to manage the generated knowledge during the software development.

**Keywords:** Software process improvement, project management, metamodel, small-sized software enterprises, small teams.

### 1 Introduction

The small-sized software enterprises —companies with less than 50 employees that have been independently financed and organized— make a significant contribution to the economy of any country in terms of employment, innovation and growth. According to [1], project management can play a significant role in facilitating this contribution, but this kind of enterprises require less bureaucratic ways of projects management than those used by larger organizations. More importantly, in the US, Brazil, Canada, China, Mexico, India, Finland, Ireland, Hungary, and many other countries, small companies represent up to 85 percent of all software organizations [2]. However, to persevere and grow, the small-sized software enterprises need efficient and effective software engineering solutions. Unlike large companies, the small ones don't have enough staff to develop functional specialties that would enable them to perform complex and secondary tasks to improve the quality of their software products. In this sense, it is true that some project management procedures/methods for medium-sized enterprises have been developed and these also have paid attention in the human aspects of Software Engineering (e.g., teamworking). Even so, they are



## 10. Referencias bibliográficas

Albrecht, A. J. & Gaffney, J. E. (1983). "Software function, source lines of code, and development effort prediction: a software science validation" *IEEE Transactions on Software Engineering*, 9(6): 639-648.

Amengual, E. & Mas, A. (2007). "Software process improvement in small companies: an experience" *Proc. of the 14<sup>th</sup> European Systems and Software Process Improvement and Innovation Conference (EuroSPI 2007)*, Industry Proceedings, pp. 11-17.

Ampatzoglou, A., Frantzeskou, G., & Stamelos, I. (2012). "A methodology to assess the impact of design patterns on software quality" *Information and Software Technology*, 54(4): 331-346.

Anda, B., Angelvik, E., & Ribu, K. (2002). "Improving estimation practices by applying use case models" In: M. Oivo & S. Komi-Sirviö (Eds.), *Product Focused Software Process Improvement*, Springer Berlin Heidelberg, pp. 383-397.

Anda, B., Benestad, H. C., & Hove, S. E. (2005). "A multiple-case study of effort estimation based on use case point" *Proc. of the Fourth International Symposium on Empirical Software Engineering (ISESE' 2005)*, IEEE Computer Society, pp. 407-416.

Atkinson, C. (1997). "Metamodelling for distributed object environments" *Proc. of the First International Enterprise Distributed Object Computing Workshop (EDOC'97)*, IEEE Computer Society, pp. 90-101.

Atkinson, R. (1999). "Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria" *International Journal of Project Management*, 17(6): 337-342.

Bailey, J. W. & Basili, V. R. (1981). "A meta-model for software development resource expenditures" *Proc. of the 5<sup>th</sup> International Conference on Software Engineering*, IEEE Computer Society, pp. 107-116.

Biffel, S., Winkler, D., Reinhard, H., & Wetzel, H. (2006). "Software process improvement in Europe: potential of the new V-modell XT and research issues" *Software Process: Improvement and Practice*, 11(3): 229-238.

Blomquist, T., Hällgren, M., Nilsson, A., & Söderholm, A. (2010). "Project-as-practice: In search of project management research that matters" *Project Management Journal*, 41(1): 5-16.

Boehm, B. W. (1981). *Software engineering economics*. Englewood Cliffs, NJ: Prentice Hall.

Bogado, V. S., Dapozo, G. N., & García-Martínez, R. (2011). "Estimación en gestión de proyectos de software basada en explotación de información" *Proc. of the XIII Workshop de Investigadores en Ciencias de la Computación (WICC 2011)*, RedUNCI, pp. 616-620.

Boucher, Q., Perrouin, G., Deprez, J. C., & Heymans, P. (2012). "Towards configurable ISO/IEC 29110-compliant software development processes for very small entities" In: D. Winkler,

R. V. O'Connor & R. Messnarz (Eds.), *Systems, Software and Services Process Improvement*, Springer Berlin Heidelberg, pp. 169-180.

Braun, C. & Winter, R. (2005). "A comprehensive enterprise architecture metamodel and its implementation using a metamodeling platform" In: J. Desel & U. Frank (Eds.), *Enterprise Modeling and Information Systems Architectures*, GI-Edition Lecture Notes (LNI), pp. 75-86.

Brooks, F. P. (1975). *The mythical man-month*. Reading, MA: Addison-Wesley.

Brooks, C. (2012). "The role of project management (PM) in academic information technology (IT)" UC Berkeley College of Engineering IT Staff. PMP, Center for Hybrid and Embedded Software, Systems (CHESS).

Calabrese, G. & Di Nauta, P. (2014). "Professionalization and dissemination processes for project management practices: a situationist perspective" In: D. Baglieri, C. Metallo, C. Rossignoli & M. Pezzillo Iacono (Eds.), *Information Systems, Management, Organization and Control*, Springer International Publishing, pp. 289-300.

Callegari, D. A., & Bastos, R. M. (2007). "Project Management and Software Development Processes: Integrating RUP and PMBOK" *Proc. of the International Conference on Systems Engineering and Modeling (ICSEM '07)*, IEEE Computer Society, pp. 1-8.

Calvo-Manzano, J., García, I., & Arcilla, M. (2008). "Hacia la gestión cuantitativa en la gestión de proyectos en el ámbito de las PYMES" *REICIS Revista Española de Innovación, Calidad e Ingeniería del Software*, 4(2): 7-19.

Canós, J. H., Letelier, P., & Penadés, M. C. (2003). "Metodologías ágiles en el desarrollo de software" *Proc. of the VIII Jornadas de Ingeniería de Software y Bases de Datos (JISBD)*, DSIC - Universidad Politécnica de Valencia, pp. 57-72.

Chrissis, M. B., Konrad, M., & Shrum, S. (2011). *CMMI® for Development, Guidelines for Process Integration and Product Improvement*. Third Edition, Boston, MA: Pearson Education, Inc.

Capilla, R., Zimmermann, O., Zdun, U., Avgeriou, P., & Küster, J. M. (2011). "An enhanced architectural knowledge metamodelo linking architectural design decisions to other artifacts in the software engineering lifecycle" In: I. Crnkovic, V. Gruhn & M. Book (Eds.), *Software Architecture*, Springer-Verlag: Heidelberg, Germany, pp. 303-318.

Chaudron, M. R. V., Heijstek, W., & Nugroho, A. (2012). "How effective is UML modeling?" *Software & Systems Modeling*, 11(4): 571-580.

Clarke, P., & O'Connor, R. V. (2012). "The situational factors that affect the software development process: Towards a comprehensive reference framework" *Information and Software Technology*, 54(5), 433-447.

CMMI Product Team. (2006). *Capability Maturity Model Integration for Development (CMMI-DEV) version 1.2*, CMU/SEI-2006-TR-008, Software Engineering Institute, Carnegie Mellon University.

CMMI Product Team. (2010). *Capability Maturity Model Integration for Development (CMMI-DEV) version 1.3*, CMU/SEI-2010-TR-033, Software Engineering Institute, Carnegie Mellon University.

Corbett, C. J., Montes-Sancho, M. J., & Kirsch, D. A. (2005). "The financial impact of ISO 9000 certification in the United States: an empirical analysis" *Management Science*, 51(7): 1046-1059.

Da Cruz, A. M. R. & Faria, J. P. (2010). "A metamodel-based approach for automatic user interface generation" In: D. C. Petriu, N. Rouquette & Ø. Haugen (Eds.), *Model Driven Engineering Languages and Systems*, Springer-Verlag: Heidelberg, Germany, pp. 256-270.

Do, Q., Cook, S., Campbell, P., Scott, W., Robinson, K., Power, W. & Tramoundanis, D. (2012). "Requirements for a metamodel to facilitate knowledge sharing between project stakeholders" *Procedia Computer Science*, 8(1): 285-292.

- Doi, T., Nishio, Y., & Osaki, S. (1999). "Optimal software release scheduling based on artificial neural networks" *Annals of Software Engineering*, 8(1-4):167-185.
- Dvir, D. (2005). "Transferring projects to their final users: the effect of planning and preparations for commissioning on project success" *International Journal of Project Management*, 23(4): 257-265.
- Ferrer, F., Giráldez, R., & Ruiz, R. (2007). "Estimación y toma de decisiones mediante Minería de Datos" In: J. Tuya, I. Ramos & J. Dolado (Eds.), *Técnicas Cuantitativas para la Gestión en Ingeniería del Software*, Editorial Netbiblo S. L., pp. 245-266.
- Fu, J., Luo, A., & Luo, X. "An UML-based meta-modeling method of building architecture product" In: K. Liu, S. R. Gulliver, W. Li & C. Yu (Eds.), *Service Science and Knowledge Innovation*, Springer-Verlag: Heidelberg, Germany, pp. 210-220.
- Fuggetta, A. (2000). "Software process: a roadmap" *Proc. of the Conference on the Future of Software Engineering (ICSE'00)*, ACM Publisher, pp. 25-34.
- Gantt, H. L. (1919). *Organizing for work*. London: Allen & Unwin.
- García, F., Piattini, M., Ruiz, F., Canfora, G., & Visaggio, C. A. (2006). "FMESP: Framework for the modeling and evaluation of software processes" *Journal of Systems Architecture*, 52(11): 627-639.
- García, I., Calvo-Manzano, J. A., Cuevas, G., & San Feliu, T. (2007). "Determining practice achievement in project management using a two-phase questionnaire on small and medium enterprises" In: P. Abrahamsson, N. Baddoo, T. Margaria, and R. Messnarz (Eds.), *Software Process Improvement*, Springer-Verlag: Heidelberg, Germany, pp. 46-58.
- García, I., Pacheco, C., & Calvo-Manzano, J. (2010). "Using a web-based tool to define and implement software process improvement initiatives in a small industrial setting" *IET software*, 4(4): 237-251.
- García, I., Pacheco, C., & Calvo-Manzano, J. A. (2015). "Changing the software engineering education: a report from current situation in Mexico" In: R. Lee (Editor), *Software Engineering Research, Management and Applications*, Springer-Verlag: Heidelberg, Germany, pp. 43-58.
- García, I., Pacheco, C., & Calvo-Manzano, J. A. (2014). "Quantitative project management in small and medium-sized software enterprises" *IEEE Latin America Transactions*, 12(3): 508-513.
- García, J., de Amescua, A., Sánchez, M. I., & Bermón, L. (2011). Design guidelines for software processes knowledge repository development. *Information and Software Technology*, 53(8): 834-850.
- García, J., Martín, D., Urbano, J., & de Amescua, A. (2013). "Practical experiences in modelling software engineering practices: The project patterns approach" *Software Quality Journal*, 21(2): 325-354.
- García, J. A. (2014). "Herramienta para la integración de la metodología Scrum con MoProSoft para incrementar la productividad del personal y la calidad del producto en las MiPyMES de desarrollo software" Tesis para obtener el grado de Maestro en Electrónica y Computación. Universidad Tecnológica de la Mixteca.
- García, S., Graettinger, C., & Kost, K. (2006). First International Research Workshop for Process Improvement in Small Settings. *Special Report CMU/SEI-2006-SR-001*. Software Engineering Institute, Carnegie Mellon University.
- Gonzalez-Perez, C., McBride, T., & Henderson-Sellers, B. (2005). "A metamodel for assessable software development methodologies" *Software Quality Journal*, 13(2): 195-214.
- Gonzalez-Perez, & Henderson-Sellers, B. (2006). "A powertype-based metamodeling framework" *Software & Systems Modeling*, 5(1): 72-90.
- Gonzalez-Perez, C. & Henderson-Sellers, B. (2007). "Modelling software development methodologies: A conceptual foundation" *Journal of Systems and Software*, 80(11): 1778-1796.

Gonzalez-Perez, C. & Henderson-Sellers, B. (2008). *Metamodelling for software engineering*. Chichester, UK: John Wiley & Sons, Inc.

Guizzardi, G. (2005). *Ontological foundations for structural conceptual models*. PhD Thesis, University of Twente, Centre for Telematics and Information Technology. Enschede, The Netherlands.

Habra, N., Alexandre, S., Desharnais, J. M., Laporte, C. Y., & Renault, A. (2008). "Initiating software process improvement in very small enterprises: Experience with a light assessment tool" *Information and Software Technology*, 50(7-8): 763-771.

Hair, J. F., Black, W. C., Babin, B. J., & Anderson, R. E. (2010). *Multivariate data analysis*. New Jersey: Pearson Prentice Hall.

Hannay, J. E., Sjöberg, D. I. K., & Dyba, T. (2007). "A systematic review of theory use in software engineering experiments" *IEEE Transactions on Software Engineering*, 33(2): 87-107.

Hastings, T. E. & Sajeev, A. S. M. (2001). "A vector-based approach to software size measurement and effort estimation" *IEEE Transactions on Software Engineering*, 27(4): 337-350.

Henderson-Sellers, B. (2011). "Bridging metamodels and ontologies in software engineering" *The Journal of Systems and Software*, 84(2), 301-313.

Hogan, J. M., Smith, G., & Thomas, R. (2002). "The real world software process" *Proc. of the Ninth Asia-Pacific Software Engineering Conference*, IEEE Computer Society, pp.366- 375.

Humphrey, W. S. (1996). *Introduction to the Personal Software Process*. 1<sup>st</sup> Edition. Addison-Wesley Professional.

Humphrey, W. S. (2000). *Introduction to the Team Software Process*, 1st Edition. Addison-Wesley Professional.

International Organization for Standardization/International Electrotechnical Commission. (2008). ISO/IEC 12207:2008(e): Information technology – software life cycle processes. Geneva, Switzerland: ISO/IEC.

International Organization for Standardization/International Electrotechnical Commission. (2011). ISO/IEC TR 29110:2011- Software engineering – Lifecycle profiles for Very Small Entities (VSEs) - Part 5-1-2: management and engineering guide. Geneva, Switzerland: ISO/IEC.

Johnson, M., Rosebrugh, R., & Wood, R. J. (2002). "Entity-relationship-attribute designs and sketches" *Theory and Applications of Categories*, 10(3): 94-112.

Juristo, N. & Gómez, O. (2012). "Replication of software engineering" B. Meyer and M. Nordio (Editors), *Empirical Software Engineering and Verification*, Berlin, Heidelberg, Germany: Springer-Verlag, pp. 60-88, 2012.

Juristo, N. & Moreno, A. (2013). *Basics of software engineering experimentation*. Dordrecht, The Netherlands: Kluwer Academic Press.

Kerzner, H. R. (2013). *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. 11<sup>th</sup> Edition. Hoboken, NY: John Wiley & Sons, Inc.

Krasner, H. (1997). "Accumulating the body of evidence for the payoff of software process improvement" In: R. B. Hunter & R. H. Thayer (Eds.), *IEEE Software Process Improvement*, IEEE Computer Society, pp. 519-540.

Kruchten, P. & Stafford, J. (2006). "The past, present, and future for software architecture" *IEEE Software*, 23(2): 22-30.

Kühne, T. (2006). "Matters of (meta-) modeling" *Software & Systems Modeling*, 5(4): 369-385.

Kwak, Y. (2005). "A brief history of project management" In: E. G. Carayannis, Y. Kwak & F. T. Anbari (Eds.), *The Story of Managing Projects: An Interdisciplinary Approach*, Praeger Publishers, pp. 1-9.

- López, A. M., Cabrera, C., & Valencia, L. E. (2008). "Introducción a la calidad de software" *Scientia et Technica*, 2(39): 326-331.
- Mahaney, R. C. & Lederer, A. L. (2006). "The effect of intrinsic and extrinsic rewards for developers on information systems project success" *Project Management Journal*, 37(4): 42-54.
- Marante, M., Letelier, P., & Suárez, F. (2009). "TUNE-UP: seguimiento de proyectos software dirigido por la gestión de tiempos" *Proc. of the XIV Jornadas de Ingeniería de Software y Bases de Datos (JISBD 2009)*, SISTEDES, pp. 57-68.
- Mas, A., Fluxà, B., & Amengual, E. (2012). "Lessons learned from an ISO/IEC 15504 SPI programme in a company" *Journal of Software Maintenance and Evolution-Research and Practice*, 24(5):493-500.
- Mas, A., Mesquida, A. L., & Fluxà, B. (2012). "The long way to maturity: a road map to success" *Proc. of the 19<sup>th</sup> European Systems and Software Process Improvement and Innovation Conference (EuroSPI 2012)*, Industry Proceedings, pp. 76-87.
- Mas, A. & Mesquida, A. (2013). "Software project management in small and very small entities" *Proc. of the 8<sup>th</sup> Iberian Conference in Information Systems and Technologies (CISTI)*, IEEE Computer Society, pp.1-6.
- Menzies, T., Chen, Z., Hihn, J., & Lum, K. (2006). "Selecting best practices for effort estimation" *IEEE Transactions on Software Engineering*, 32(11): 883-895.
- Mesquida, A. L., & Mas, A. (2014). "A project management improvement program according to ISO/IEC 29110 and PMBOK®" *Journal of Software: Evolution and Process*, 26(9): 846-854.
- Ministerio de Industria, Energía y Turismo. (2014). La nueva definición de PYME. Guía de usuario y ejemplo de declaración. Publicaciones de Empresa e Industria. Comisión Europea. Disponible en: <http://www.ipyme.org/es-ES/CPyme/Documents/NuevaDefinicionPyme.pdf>, Último acceso: Julio 2014.
- Mitre-Hernández, H., Ortega-Martínez, E., & Lemus-Olalde, C. (2014). "Estimación y control de costos en métodos ágiles para desarrollo de software: un caso de estudio" *Ingeniería Investigación y Tecnología*, 15(3): 403-418.
- Mochi, P. & Hualde, A. (2009). *Desafíos y Oportunidades de la Industria del Software en América Latina*. México, D.F.: Ediciones Mayol S. A.
- Mohapatra, S. & Sreejesh, S. (2014). "Model for improving productivity without impacting quality of deliverables in IT projects" *International Journal of Information Technology Project Management*, 5(2): 25-41.
- Mora, B., Ruiz, F., García, F., & Piattini, M. (2006). *Definición de lenguajes de modelos MDA vs DSL*. Reporte Técnico. Departamento de Tecnologías y Sistemas de Información, Universidad de Castilla-La Mancha, España.
- Niazi, M., Babar, M. A., & Verner, J. M. (2010). "Software process improvement barriers: a cross-cultural comparison" *Information and Software Technology*, 52(11): 1204-1216.
- Niazi, M. & Zahran, S. (2013). "Software process lines: a step towards software industrialization" In: H. Probst (Editor), *Industrial Engineering: Concepts, Methodologies, Tools, and Applications*. Hershey: PA, IGI Global. pp. 1-17.
- Nizam, M. H. & Sahibuddin, S. (2011). "Critical success factors for software projects: comparative study" *Scientific Research and Essays*, 6(10): 2174-2186.
- Normalización y Certificación. (2005). NMX-I-059/02-NYCE-2005. Tecnología de la Información -Software- Modelos de procesos y evaluación para desarrollo y mantenimiento de software. Parte 02: Requisitos de procesos (MoProSoft). México, DF: NYCE.
- Object Management Group. (2002). "Software process engineering metamodel specification (SPEM): adopted specification", version 1.0.

Object Management Group. (2003). Meta Object Facility (MOF) 2.0. OMG Adopted Specification. Disponible en: <http://www.omg.org>, Último acceso: Mayo 2015.

O'Connor, R. V. & Laporte, C. Y. (2012). "Software project management in very small entities with ISO/IEC 29110" In: D. Winkler, R. V. O'Connor & R. Messnarz (Eds.), *Systems, Software and Services Process Improvement*, Springer-Verlag: Heidelberg, Germany, pp. 330-341.

Office of Government Commerce. (2005). Tailoring PRINCE2. The Stationery Office, London.

Ohsugi, N., Monden, A., Kikuchi, N., Barker, M. D., Tsunoda, M., Kakimoto, T., & Matsumoto, K. (2007). "Is this cost estimate reliable? – The relationship between homogeneity of analogues and estimation reliability" *Proc. of the First International Symposium on Empirical Software Engineering and Measurement (ESEM'07)*, IEEE Computer Society, pp. 384-392.

Oktaba, H. (2006). "MoProSoft: A software process model for small enterprises" *Proc. of the First International Research Workshop for Process Improvement in Small Settings (Special Report CMU/SEI-2006-SR-001)*. Software Engineering Institute, Carnegie Mellon University, pp. 93-101.

Oktaba, H., Morales, M. E., & Dávila, M. (2012). *KUALI-BEH: Software Project Common Concepts. Version 1.1*. México, D. F: Universidad Autónoma de México.

Oktaba, H., Esquivel, C., Su Ramos, A., Martínez, A., Quintanilla, G., Ruvalcaba, M., & Fernández, Y. (2005). *Modelo de Procesos para la Industria de Software MoProSoft Versión 1.3*. México, D. F: Secretaría de Economía.

Olson, T. G. (2006). "Defining short and usable processes" *CrossTalk – The Journal of Defense Software Engineering*, 19(6): 24-28.

O'Sheedy, D., Xu, J., & Sankaran, S. (2010). "Preliminary results of a study of agile project management techniques for an SME environment" *International Journal of Arts and Sciences*, 3(7): 278-291.

O'Sheedy, D. & Sankaran, S. (2013). "Agile project management for IT projects in SMEs: a framework and success factors" *The International Technology Management Review*, 3(3): 187-195.

Patanakul, P., Iewwongcharoen, B., & Milosevic, D. (2010). "An empirical study on the use of project management tools and techniques across project life-cycle and their impact on project success" *Journal of General Management*, 35(3): 41-65.

Pérez-Castillo, R., García-Rodríguez, I. & Piattini, M. (2011). "Knowledge discovery metamodel-ISO/IEC 19506: A standard to modernize legacy systems" *Computer Standards & Interfaces*, 33(6): 519-532.

Pfleeger, S. L. (1995). "Experimental design and analysis in software engineering" *Annals of Software Engineering*, 1(1): 219-253.

Pfleeger, S. L. (2005). "Soup or art? The role of evidential force in empirical software engineering" *IEEE Software*, 22(1): 66-73.

Pino, F. J., García, F., & Piattini, M. (2007). "Contribución de los estándares internacionales a la gestión de procesos software" *Revista de Procesos y Métricas*, 4(2): 33-43.

Pino, F. J., García, F., & Piattini, M. (2008). "Software process improvement in small and medium software enterprises: a systematic review" *Software Quality Journal*, 16(2): 237-261.

Pino, F. J., Pardo, C., García, F., & Piattini, M. (2010). "Assessment methodology for software process improvement in small organizations" *Information and Software Technology*, 52(10): 1044-1061.

Pino, F. J., Ruiz, F., García, F., & Piattini, M. (2012). "A software maintenance methodology for small organizations: Agile\_MANTEMA" *Journal of Software: Evolution and Process*, 24(8): 851-876.

Putnam, L. H. (1978). "A general empirical solution to the macro software sizing and estimating problem" *IEEE Transactions on Software Engineering*, 4(4): 345-361.

- Pressman, R. & Maxim, B. (2014). *Software Engineering: A Practitioner's Approach*, England, UK: McGraw Hill Science.
- Project Management Institute. (2013). *Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBoK)*, 5ª Edición. Newtown Square, PA: Project Management Institute.
- Radatz, J., Geraci, A., & Katki, F. (1990). IEEE standard glossary of software engineering terminology. *IEEE Std. 610121990* (121990).
- Richardson, I. & Gresse von Wangenheim, C. (2007). "Why are small software organisations different?" *IEEE Software*, 24(1): 18-22.
- Rivera, M. A. & Quiroz, V. (2013). "Redes de conocimiento y aprendizaje tecnológico en las empresas de la industria del software en los territorios de Guadalajara y Distrito Federal, México" *Proc. of the XV Congreso Latino-Iberoamericano de Gestión de Tecnología (ALTEC 2013)*, UTEN, pp. 125-136.
- Robillard, M. P., Walker, R. J., & Zimmermann, T. (2010). "Recommendation systems for software engineering" *IEEE Software*, 27(4): 80-86.
- Rodríguez, G. H., Berdun, L. S., Soria, Á., Amandi, A., & Campo, M. (2014). "Un enfoque inteligente para asistir en la planificación de proyectos ágiles" *Proc. of the XLIII Jornadas Argentinas de Informática e Investigación Operativa (JAIIO)*, SADIO, pp. 75-82.
- Roszkowski, J. & Roszkowska, A. (2013). "The quality management metamodel in the enterprise architecture" In: T. Skersys, R. Butleris & R. Butkiene (Eds.), *Information and Software Technologies - Communications in Computer and Information Science*, Springer-Verlag: Heidelberg, Germany, pp. 11-21.
- Ruíz, C., Piore, M. & Schrank, A. (2005). "Los retos para la industria del software" *Comercio Exterior*, 55(9): 744-753.
- Ruíz, J. S., Ledeneva, Y., & Cuesta, H. (2014). "Gestión del conocimiento en la micro y pequeña empresa mexicana de la industria del software" *Research in Computing Science*, 73(1): 103-116.
- Runeson, P. & Höst, M. (2009). "Guidelines for conducting and reporting case study research in software engineering" *Empirical Software Engineering*, 14(2): 131-164.
- Sabini, L. (2014). "The institutionalization of project management: an analysis of the actors, strategies and paths that lead to the professionalization of a new discipline" Doctoral dissertation. Department of Business and Management, LUISS Guido Carli University.
- Salo, O. & Abrahamsson, P. (2008). "Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum" *IET Software*, 2(1): 58-64.
- Salomo, S., Weise, J., & Gemunden, H. G. (2007). "NPD planning activities and innovation performance: the mediating role of process management and the moderating effect of product innovativeness" *Journal of Product Innovation Management*, 24(4): 285-302.
- Sampedro, J. L. & Vera-Cruz, J. A. (2008). "Absorptive capacity of information and knowledge through interfaces in the customized software industry: the case of micro and small-sized Mexican firms" *Proc. of the GLOBELICS 6th International Conference*, Georgia Institute of Technology, pp. 1-17.
- Saunders, M., Lewis, P., & Thornhill, A. (2003). *Research methods for business students*. 3<sup>rd</sup> Edition. Harlow, EN: Financial Times Prentice Hall.
- Sidman, M. (1960). *Tactics of scientific research: Evaluating experimental data in psychology*. New York, NY: Basic Books.
- Söderlund, J. & Müller, R. (2014). "Project management and organization theory: IRNOP meets PMJ" *Project Management Journal*, 45(4): 2-6.

Söderström, E., Andersson, B., Johannesson, P., Perjons, E., & Wangler, B. (2002). "Towards a framework for comparing process modelling languages" In: A. Banks, M. Tamer, J. Mylopoulos, & C. C. Woo (Eds.), *Advanced Information Systems Engineering*, Springer-Verlag: Heidelberg, Germany, pp. 600-611.

Somerville, I. (2015). *Software Engineering*, 10<sup>th</sup> Edition. Boston, MA: Pearson Education Inc.

Sosa, R., Blanc, R., Pralong, L., Álvarez, C., & Galáz, S. (2013). "El desafío de producir software de calidad aplicando prácticas de CMMI para las pymes de Concepción del Uruguay, Entre Ríos" *Proc. of the 1<sup>er</sup> Congreso Nacional de Ingeniería Informática/Sistemas de Información (CoNaIISI 2013)*, CONFEDI, pp. 125-135.

Spendolini, M. (1992). *The benchmarking book*. New York: Amacom.

Taylor, F. W. (1914). *The principles of scientific management*. New York: Harper & Brothers.

Tasevska, F., Damij, T. & Damij, N. (2014). "Project planning practices based on enterprise resource planning systems in small and medium enterprises — A case study from the Republic of Macedonia" *International Journal of Project Management*, 32(3): 529-539.

The Standish Group International. (2013). *Chaos Manifesto 2013. Think Big, Act Small*. The Standish Group International, Incorporated.

Turner, R., Ledwith, A., & Kelly, J. (2010). "Project management in small to medium-sized enterprises: Matching processes to the nature of the firm" *International Journal of Project Management*, 28(8): 744-755.

van den Bergh, A., Van Haaren, J., Van Baelen, S., Berbers, Y., & Joosen, W. (2013). "Towards an automated pattern selection procedure in software models" *Proc. of the International Conference on Inductive Logic Programming (ILP 2012)*, CEUR-WS, pp. 68-73.

Varkoi, T. (2010). "Process assessment in very small entities — An ISO/IEC 29110 base method" *Proc. of the Seventh International Conference on the Quality of Information and Communications Technology (QUATIC)*, IEEE Computer Society, pp. 436-440.

Wachsmuth, G. (2007). "Metamodel adaptation and model co-adaptation" *Proc. of the 21<sup>st</sup> European Conference on Object-Oriented Programming (ECOOP 2007)*, Springer Berlin Heidelberg, pp. 600-624.

Walker, E. (2003). "Implementing best practices in the Joint Battlespace Infosphere (JBI) program at AFRL" *Conference on the Acquisition of Software-Intensive System*, Carnegie Mellon University, January 28-30, 2003.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B. & Wesslén, A. (2012). *Experimentation in software engineering*. Boston, MA: Kluwer Academic Publishers.

Yin, R. K. (2013). *Case study research: Design and methods*. 4<sup>th</sup> Edition, Thousand Oaks, CA: Sage Publications.

Young, H., Fang, T., & Hu, C. (2006). "A successful practice of applying software tools to CMMI process improvement" *Journal of Software Engineering Studies*, 1(2): 78-95.

## 10.1. Sitios de internet

[URL-1] <http://www.plantrac.co.uk/overview.htm>

Página Oficial de Plantrac (Último acceso: Noviembre, 2014).

[URL-2] <http://www.pricystems.com/leadership/pricemilestones/tabid/262/nessus/>

Página oficial de PRINCE (Último acceso: Noviembre, 2014).

[URL-3] <http://www.hermes.admin.ch/>

Página oficial de Hermes 5 (Último acceso: Marzo 2015).